Evidential Reasoning with Gister-CL
A Manual

John D. Lowrance

1997

Preface

This document is designed to serve as a self-contained introduction to evidential reasoning and Gister-CL. Evidential reasoning is a collection of techniques for automated reasoning from evidence; Gister-CL is an application independent implementation of these techniques. As such, Gister-CL might serve either as the foundation for application specific implementations of evidential reasoning or as a basis for research in uncertain reasoning. Both evidential reasoning and Gister-CL are undergoing further development. Therefore, this document will be periodically updated.

Gister-CL's data structures are represented as Grasper-CL graphs. In addition, Gister-CL's user interface is closely modeled after Grasper-CL's. For these reasons, knowledge of Grasper-CL provides a good foundation for learning how to use Gister-CL, however it is not a prerequisite. The *Grasper-CL User's Guide* (in [KLS93]) provides a thorough overview of the foundations and use of Grasper-CL. The *Grasper-CL Programmer's Manual* (also in [KLS93]) provides descriptions of the COMMON LISP procedures that comprise Grasper-CL.

We suggest that a novice begin by reading the first two chapters of this manual to acquire a basic understanding of evidential reason, and then read the fourth chapter while simultaneously working with an implementation of Gister-CL. The last section of this chapter contains some suggested exercises.

The third chapter introduces the more advanced concepts in evidential reasoning and should only be read after one is comfortable with the basic concepts. Finally, the last chapter should be read if the reader wants to use Gister-CL under program control.

John D. Lowrance

Contents

List of Figures

Chapter 1

Introduction

The Artificial Intelligence Center at SRI has a long history in developing technology to address the problem of automated information management within real-world contexts [GLF81, Low82, LG83, LSG86, Wes88, Gar86, Rus86, Rus87, SL89, Str90, LGS90, LSW$^+$91]. The information required to understand real-world situations comes from several sources: real-time sensor data, previously stored general knowledge, and current contextual information. Sensors typically provide *evidence* in support of certain conclusions. Evidence is characteristically uncertain: it allows for multiple possible explanations; it is incomplete: the source rarely has a full view of the situation; and it may be completely or partially incorrect. The quality and the ease with which situational information may be extracted from a synthesis of current sensor data and prestored knowledge is a function both of how strongly the characteristics of the sensed data focus on appropriate intermediate conclusions and on the strength and effectiveness of the relations between those conclusions and situation events.

SRI's work in automated uncertain reasoning emphasizes the practical application of theoretically sound techniques for reasoning from evidence—that is, information that is potentially incomplete, inexact, inaccurate, and from diverse sources. SRI pioneered *evidential reasoning* for drawing conclusions from multiple sources of evidential information about dynamic real-world situations. We have developed formal foundations for reasoning under uncertainty covering both probabilistic models (i.e., Bayesian and Dempster-Shafer) and possibilistic models (i.e., Fuzzy Logic) and have incorporated all of these into a single uncertain reasoning tool, Gister-CL.

Gister-CL supports the rapid development of evidential reasoning systems through an interactive menu-driven, graphical interface. The user interacts with the system in much the same way as with electronic spreadsheets, by simply selecting from menus to add evidential operations to an analysis (including fusion, source discounting, time projection, summarization, evidential interpretation, and sensitivity analysis), to modify data or operation parameters, or to change any portion of the uncertain knowledge base. In response, Gister-CL updates its analyses to reflect the new information.

Gister-CL is designed as a tool for the domain expert. With this tool, an expert can quickly and flexibly develop an argument (i.e., a line of reasoning) specific to a given domain situation. Gister-CL helps the expert keep track of the complex interrelationships among the components of his arguments, insure that the relevant information has been properly incorporated, and reveal the more tentative aspects of the arguments. Once an analysis has been established by an expert, it can be instantiated over different situations by nonexperts. To improve run-time efficiency, Gister-CL supports the compilation of such analyses into stand-alone procedures. Gister-CL has been applied to a wide range of problems, including multisensor interpretation, mission planning, intelligence analysis, vehicle tracking, and threat identification, among others.

We have developed both a formal basis and a framework for implementing automated reasoning systems based upon evidential reasoning techniques. Both the formal and practical approach can be divided into four parts: (1) specifying a set of distinct propositional spaces (i.e., *frames of discernment*), each of which delimits a set of possible world situations; (2) specifying the interrelationships among these propositional spaces (i.e., *compatibility relations* in a *gallery*); (3) representing bodies of evidence as belief distributions over these propositional spaces (i.e., *mass distributions*); and (4) establishing paths (i.e., *analyses*) for evidence to flow through these propositional spaces by means of evidential operations, eventually converging on spaces where the target questions can be answered. These steps specify a means for arguing from multiple bodies of evidence toward a particular (probabilistic) conclusion. Argument construction is the process by which such evidential analyses are constructed and is the analogue of constructing proof trees in a logical context.

This technology features the ability to reason from uncertain, incomplete, and occasionally inaccurate information, these being characteristics of the information available in real-world domains. It provides options for the representation of information: independent opinions are expressed by multiple (independent) bodies of evidence; dependent opinions can be expressed either by a single body of evidence or by a network (i.e., analysis) that describes the interrelationships among several bodies of evidence. These networks of bodies of evidence capture the genealogy of each body and are used as data-flow models to automatically update interrelated beliefs whenever any given belief is revised. The technology includes the following evidential operations, which are based in theory but have intuitive appeal as well.

- FUSION: This operation pools multiple bodies of evidence pertaining to a common question into a single body of evidence that emphasizes points of agreement and deemphasizes points of disagreement.
- DISCOUNTING: This operation adjusts a body of evidence to reflect the credibility of its source. If a source is completely reliable, discounting has no effect; if it is completely unreliable, discounting strips away all apparent information content; otherwise, discounting reduces the apparent information content in proportion to the source's unreliability.
- TRANSLATION: This operation moves a body of evidence away from its original context to a related one, to assess its impact on dependent hypotheses.
- INTERPRETATION: This operation calculates the truthfulness of a given statement

based upon a given body of evidence. It produces an estimate of both the positive and negative effects of the evidence on the truthfulness of the statement.

- PROJECTION: This operation moves a body of evidence away from its original temporal context, to a related one, to assess its impact on a different time.
- SUMMARIZATION: This operation eliminates extraneous details from a body of information. The resulting body of evidence is slightly less informative, but remains consistent with the original.
- GISTING: This operation produces a single statement that captures the general sense of a body of evidence, without reporting degrees of certainty.

Our goal is to provide effective automated aids to domain experts for argument construction.

1.1 Background and Motivation

Expert system technology emerged from the artificial intelligence (AI) research laboratories and entered the market place during the 1980s. This technology emerged in the form of expert system shells, application-independent software systems that support the construction and automated use of knowledge bases. As a result, expert systems have been successfully developed for a wide variety of applications.

Many of the commercially available expert-system shells used production rules as their formalism for representing knowledge. In its simplest form, the *production rule* (also termed an "if-then" rule) consists of an antecedent, the "if" part, and a consequent, the "then" part. The interpretation of the rule is: given information establishing the truth of the antecedent, the truth of the consequent can be inferred. In practice, production rules are applied, in accordance with a system's control strategy, by matching rule antecedents against a database of facts; when a successful match is made, the consequent is added to the database.

However, expert knowledge is frequently suggestive rather than conclusive; the rules may therefore include a strength that is related to the conditional probabilities of the consequent given the antecedent, and of the antecedent given the consequent. Thus, based upon the current confidence in the antecedent in the database of facts and on the rule's strength, a confidence is derived for the consequent. This consequent may match the antecedent of other rules and thereby trigger their activation, thus resulting in the propagation of the influence of the original match throughout the database. Of course, multiple rules may share a common consequent, in which case it is necessary to have a means of resolving different confidence estimates that are derived through the use of different rules.

One way to visualize such a knowledge base is as a directed graph or *inference network*, where each antecedent and consequent is represented by a node and each rule is represented by a directed arc connecting its antecedent to its consequent. In essence, this inference network [DHN81] represents an argument, a line of reasoning that explains how certain premises support certain conclusions. Given probabilistic estimates of the truthfulness of certain facts, probabilistic conclusions can be automatically drawn. Thus, if the knowledge base (i.e., argument) was developed by an expert in the domain of application, this expertise becomes accessible to nonexperts.

In complex domains, it is often extremely difficult to select and coordinate all of the potentially relevant rules. The strength, and simultaneous weakness, of the rule-based approach is that the knowledge is represented in small chunks. In principle, each rule captures a very limited piece of knowledge (i.e., how one concept is directly related to another), and each is easily elicited from an expert and is easily understood. The original concept was that one could establish the validity of each rule in isolation and that the validity of the entire rule base would follow. Unfortunately, this is not the case.

The rules are often numerous and unstructured, redundantly expressing some relationships while leaving others unspecified. Thus the confidences and strengths frequently do not represent sufficient probabilistic information to solve uniquely for the probabilities of the consequents. In addition, the confidences and strengths often represent subjective estimates, which are invariably inconsistent. Thus, no formally justifiable methods emerged for uncertain reasoning based upon unstructured collections of rules. Therefore, either heuristic methods for confidence propagation were adopted, despite the loss of formal semantics and the consequent difficulties in understanding and controlling them, or uncertainties were eliminated entirely, limiting the applicability of the resulting systems to those situations where certain information is available.

More recent theoretical work has resulted in a new approach to this problem. Instead of constructing knowledge bases from unstructured collections of weighted rules, this new approach requires that the rules each represent a probabilistic constraint between two variables and that the collection of rules form a tree (or can be systematically converted into a tree). Between any two variables in this knowledge base, there is a unique path of intervening variables; the variables at the ends of the path must be conditionally independent of one another, given any of the intervening variables. In Bayesian nets [Pea88], these constraints are expressed in terms of conditional probabilities; in Markov trees [SS86, SS88], which are based upon the theory of belief functions, joint distributions are used to express these constraints.

Using this approach, each variable is represented by a node and each constraint by an arc. Given a probabilistic estimate of the value of a variable, this estimate can be propagated through the network via the constraints, to predict the values of related variables. When independent estimates are provided for several variables, their joint impact on all variables can be computed by propagating effects throughout the network. The advantage of this approach is that it is theoretically sound and computationally practical; the disadvantage is that it requires the knowledge engineer to think much more carefully about how the knowledge base must be structured prior to construction.

This theoretical work can be used to guide the construction of evidential analyses [LGS90], retaining the soundness of the theory, while providing a practical framework for constructing expert systems. Although our approach was founded based upon the mathematics of belief functions developed by Dempster and Shafer [Dem68, Sha76a, Sha86], it is now capable of supporting both Markov trees and Bayesian nets. When adequate probabilistic information is available, the computations reduce to those of Bayesian nets; when more limited information must suffice, the computations relax to those found in Markov trees. In addition, our evidential analyses employ several practical constructs that ease their development without violating the underlying theory.

1.2 Requirements for Automated Argument Construction

Conventional expert-system techniques are adequate for applications where the potential relevance of pertinent information can be prespecified. However, an important aspect of some application domains (e.g., intelligence analysis) is the ability to link what might first seem to be both irrelevant and unrelated pieces of information in unique ways that permit new conclusions to be drawn. This linking requires far more flexible interaction of the expert with the knowledge.

A system that can support automated argument construction must satisfy a number of requirements. First, its knowledge representations and operations, with which the user needs to interact, should be intuitive and easily understood. The user, who is interested in an answer to his problem and not in uncertain-reasoning techniques, should not be burdened with technical matters that are a function of the underlying technology. Although it is by no means a requirement, we have found that graphically oriented representations are often a particularly good choice.

Second, the knowledge should be represented in such a way as to be independent of how it is eventually used during argument construction. That is, the user should be able to state the facts directly as he sees them, without having to specify how they might be used in an argument. If two events tend to co-occur, then the user should be able to so state, without having to select one event as an indicator for the other.

Third, the knowledge must be easily modifiable. This requirement includes both the ability to make a modification quickly and the ability to understand its impact. Toward this end, if multiple actions are required on the part of the user, the system should guide the user through any modification step by step. Once a modification is complete, the argument should automatically react to the change, updating any conclusions that depend upon it. Thus, the system serves as an experimental environment in which a space of alternative formulations can be easily explored.

Fourth, the system's information requirements should match the availability and precision of the information in the problem domain. If prior probabilities cannot be reasonably assessed, they should not be required, or they should be representable in a form that retains their tentative nature.

Finally, the system needs to be theoretically well grounded. If it is not, both its stability and understandability suffer. If the user is to be free to explore his problem, the system must be flexible enough to support the user's exploration, without fear of its collapse because of heuristic frailties. In addition, there should be opportunity for true analysis, not just experimental testing. The user must be able to analyze his argument to understand its structure and its sensitivities.

Although satisfying these requirements is obviously very difficult, Gister-CL represents our best efforts to automate argument construction.

Chapter 2

The Basics of Argument Construction

Gister-CL divides the problem of argument construction into two major steps: framing the problem and analyzing the evidence. In framing the problem, the user establishes a gallery of frames and compatibility relations that delimits a space of possibilities. In analyzing the evidence, each body of evidence is represented relative to a frame in the gallery and a sequence of evidential operations is established. This sequence determines how the evidence is transformed into pertinent conclusions. Collectively the gallery of frames and compatibility relations, together with the analyses, are the rough equivalent of an expert system's knowledge base.

2.1 Framing the Problem

2.1.1 Frames of Discernment

Suppose that the answer to some question $\mathcal{A}$ is contained in a finite set Θ_A. That is, each element a_i of Θ_A corresponds to a distinct possible answer to the question $\mathcal{A}$, no two of which can be simultaneously true. For example, $\mathcal{A}$ might be a question concerning the location of some ship. In this case, Θ_A would consist of all the possible locations for that vessel. Θ_A is called a *frame of discernment*:

$$\Theta_A = \{a_1, a_2, \ldots, a_n\} \quad .$$

Once a frame of discernment has been established for a given question, it formalizes a *variable* where each possible value for the variable is an element of the frame. A statement pertaining to the value of this variable is discerned by the frame, just in case the impact of the statement is to focus on some subset of the possible values in the frame as containing the true value. In other words, a propositional statement A_i about the answer to question $\mathcal{A}$ corresponds to a subset of Θ_A. For example, if the statement is "the ship is docked," then it corresponds to the set of locations in Θ_A that are adjacent to docks.

$$A_i \subseteq \Theta_A \quad .$$

Other propositions related to this question can be similarly represented as subsets of Θ_A (i.e., as elements of the power set of Θ_A, denoted 2^{Θ_A}); the subset A_j might correspond to those locations in Θ_A that are at sea. Once this has been accomplished, logical questions involving multiple statements can be posed and resolved in terms of the frame. Given two propositions, A_i and A_j, and their corresponding sets, A_i and A_j, the following logical operations and relation can be resolved through the associated set operations and relation:

$$\begin{aligned}
\neg\mathsf{A}_i &\iff \Theta_A - A_i \\
\mathsf{A}_i \wedge \mathsf{A}_j &\iff A_i \cap A_j \\
\mathsf{A}_i \vee \mathsf{A}_j &\iff A_i \cup A_j \\
\mathsf{A}_i \Rightarrow \mathsf{A}_j &\iff A_i \subseteq A_j \quad .
\end{aligned}$$

Thus, when two statements pertaining to the same question are available, and they are each represented as subsets of the same frame, their joint impact is calculated by intersecting those two subsets. All other statements that correspond to supersets of this result in Θ_A, are implicitly true; all of those statements whose corresponding sets are disjoint from this result are implicitly false; and the truthfulness of all others statements is undetermined. As additional information becomes available, it can be combined with the current result in the same way. Since intersection is commutative and associative, the order in which information enters is of no consequence.

In implementing this formal approach, we have found that frames, like the other formal elements in this theory, can be straightforwardly represented as graphs consisting of nodes connected by directed edges (i.e., arcs). Because they are graphs, these formal elements are easily understood, and they provide an intuitive basis for human-computer interaction. A frame is represented by a named graph that includes a node for each element of the frame and may include additional nodes representing *aliases*, i.e., named disjunctions of elements. Each of these additional nodes has edges pointing to elements of the frame (or other aliases) that make up the disjunction. Here, the possible locations for a ship might be represented by a graph named LOCATIONS (Figure 2.1) that includes six elements (ZONE1, ZONE2, ZONE3, CHANNEL, LOADING-DOCK, REFUELING-DOCK) and three aliases (IN-PORT, DOCKED, AT-SEA).

If other aspects of ships are of interest besides their location, then additional frames of discernment might be defined. For example, the activities of these ships might be of interest. If so, an additional frame Θ_B might be defined to include elements corresponding to refueling, loading cargo, unloading cargo, being enroute, and being under tug escort. Propositional statements pertaining to a ship's activity can then be defined relative to this frame; e.g.,

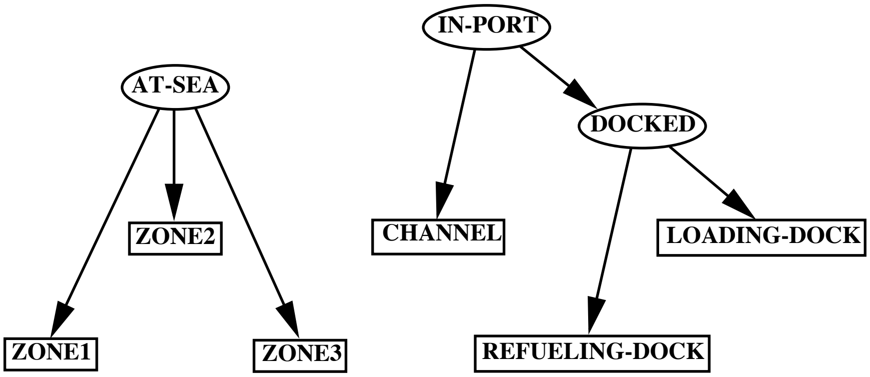


Figure 2.1: LOCATIONS frame

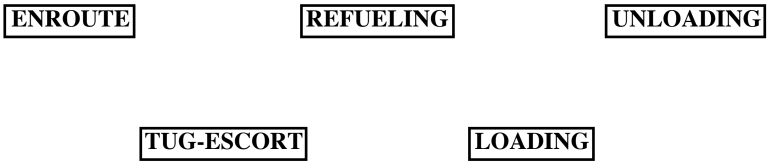


Figure 2.2: ACTIVITIES frame

$$\begin{aligned} \Theta_B &= \{b_1, b_2, \ldots, b_m\} \\ B_j &\subseteq \Theta_B \quad . \end{aligned}$$

The frame is represented by a graph named ACTIVITIES (Figure 2.2) that includes five elements: ENROUTE, TUG-ESCORT, UNLOADING, LOADING, REFUELING.

2.1.2 Compatibility Relations

So far, propositional statements pertaining to a ship's location or to its activity can be addressed separately, but they cannot be jointly considered. To do this, one must first define a *compatibility relation* between the two frames. A compatibility relation simply describes which elements from the two frames can be true simultaneously. For example, a ship located at a loading dock might be loading or unloading cargo, but is not refueling, or enroute. In other words, being located at a loading dock is compatible only with one of two activities, loading or unloading. Thus, the compatibility relation between frames Θ_A and Θ_B is a subset of the cross product of the two frames. A pair (a_i, b_j) is included if and only if they can be true simultaneously. Typically, there is at least one pair (a_i, b_j) included for each a_i in Θ_A with the analogue true for each b_j:

$$\Pi_{(A,B)} \subseteq \Theta_A \times \Theta_B \quad .$$

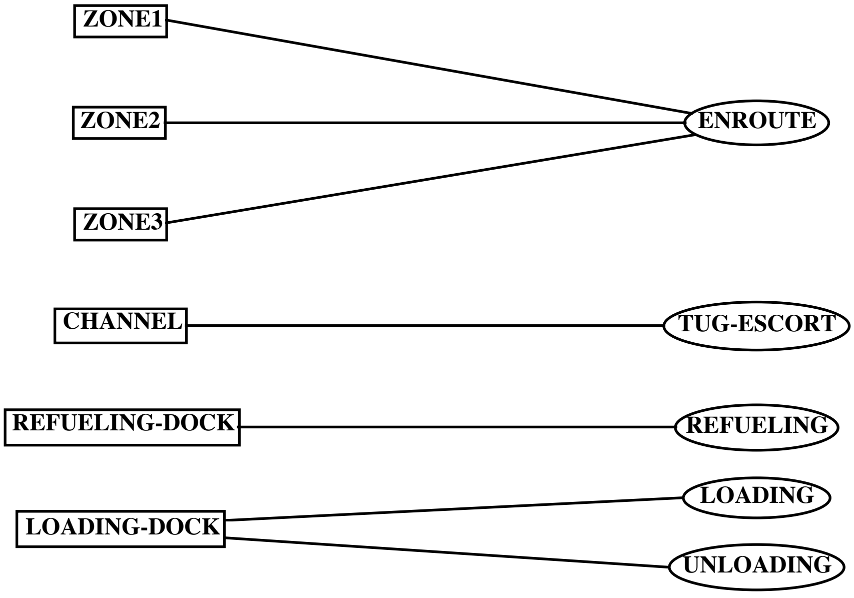


Figure 2.3: LOCATIONS-ACTIVITIES compatibility relation

Using the compatibility relation $\Pi_{(A,B)}$ we can define a *compatibility mapping* $\Gamma_{A\mapsto B}$ for translating propositional statements expressed relative to Θ_A to statements relative to Θ_B. If a statement A_k is true, then the statement $\Gamma_{A\mapsto B}(A_k)$ is also true:

$$\begin{aligned} \Gamma_{A\mapsto B} : 2^{\Theta_A} &\mapsto 2^{\Theta_B} \\ \Gamma_{A\mapsto B}(A_k) &= \{\, b_j \mid (a_i, b_j) \in \Pi_{(A,B)},\ a_i \in A_k \,\} \quad . \end{aligned}$$

A compatibility relation is represented as a graph that includes the nodes from the frames that it relates with edges connecting compatible elements. For example, in the LOCATIONS-ACTIVITIES compatibility relation (Figure 2.3) relating the LOCATIONS and ACTIVITIES frames, ZONE1, ZONE2, and ZONE3 are all connected to ENROUTE (because these zones represent areas at sea), CHANNEL is connected to TUG-ESCORT (because a ship entering or leaving the port at the end of this channel would be under tug-boat control), LOADING-DOCK is connected to both LOADING and UNLOADING (because either activity is consistent with being at that dock), and REFUELING-DOCK is connected to REFUELING. The edges define the compatibility mapping from LOCATIONS to ACTIVITIES, moving left-to-right along the edges, and the mapping from ACTIVITIES to LOCATIONS, moving right-to-left along the edges.

Instead of translating propositional statements between these two frames via $\Gamma_{A\mapsto B}$ and $\Gamma_{B\mapsto A}$, we might choose to translate these statements to a common frame that captures all of the information. This common frame, $\Theta_{(A,B)}$, is identical to the compatibility relation $\Pi_{(A,B)}$. Frame Θ_A, and analogously Θ_B, is trivially related to frame $\Theta_{(A,B)}$ via the following compatibility relation and compatibility mappings:

$$\begin{aligned}
\Pi_{(A,(A,B))} &= \{\, (a_i, (a_i, b_j)) \mid (a_i, b_j) \in \Pi_{(A,B)} \,\} \\
\Gamma_{A\mapsto(A,B)}(A_k) &= \{\, (a_i, b_j) \mid (a_i, (a_i, b_j)) \in \Pi_{(A,(A,B))},\ a_i \in A_k \,\} \\
&= \{\, (a_i, b_j) \mid (a_i, b_j) \in \Pi_{(A,B)},\ a_i \in A_k \,\} \\
\Gamma_{(A,B)\mapsto A}(X_k) &= \{\, a_i \mid (a_i, b_j) \in \Pi_{(A,B)},\ (a_i, b_j) \in X_k \,\} \quad .
\end{aligned}$$

Faced with a complex problem, we might either define a single complex frame that encompasses all aspects of interest or, alternatively, define a complex network of frames that includes a distinct frame for each aspect of interest. However, these may not be equivalent. For example, consider the following frame:

$$\Theta_{(A,B,C)} = \{(a_1, b_1, c_1), (a_2, b_1, c_2), (a_2, b_2, c_2)\} \quad .$$

If this frame properly captures the relationship among frames Θ_A, Θ_B, and Θ_C, then c_1 is the only element from Θ_C compatible with a_1 from Θ_A. However, if we maintain these as three separate frames connected by compatibility mappings, $\Gamma_{A\mapsto B}, \Gamma_{B\mapsto A}, \Gamma_{B\mapsto C}$, and $\Gamma_{C\mapsto B}$, both c_1 and c_2 are compatible with a_1 because a_1 is compatible with b_1, and b_1 is compatible with both c_1 and c_2, i.e., $\Gamma_{B\mapsto C}(\Gamma_{A\mapsto B}(\{a_1\})) = \{c_1, c_2\}$. However, if a_1 is true, then it follows that either c_1 or c_2 is true. Similarly, a_2 is compatible with both c_1 and c_2. Thus, this sequence of three frames connected by compatibility mappings corresponds to the following composite frame:

$$\hat{\Theta}_{(A,B,C)} = \{(a_1, b_1, c_1), (a_2, b_1, c_2), (a_2, b_2, c_2), (a_1, b_1, c_2), (a_2, b_1, c_1)\} \quad .$$

We will return to this discussion later.

In dynamic environments, compatibility relations can be used to reason over time. If Θ_{A1} represents the possible states of the world at time one and Θ_{A2} represents the possible states at time two, then a compatibility relation, $\Pi_{A1,A2}$, can capture the possible state transitions. For example, if Θ_{A1} and Θ_{A2} both represent the possible locations of a ship (i.e., they are identical to Θ_A as previously defined), then $\Pi_{A1,A2}$ could represent the constraints on that ship's movement. A pair of locations (a_i, a_j) would be included in $\Pi_{A1,A2}$ if a ship located at a_i at Hour 1 (i.e., time) could reach a_j by Hour 2. If we assume that the possible movements of a ship are constrained in the same way over any two-hour period, then the compatibility mapping associated with this compatibility relation can be reapplied as many times as necessary to constrain the possible locations of a ship across an arbitrary number of hours.

DELTA-LOCATIONS and DELTA-ACTIVITIES (Figures 2.4 and 2.5) are two compatibility relations that relate frames to themselves. They represent possible state transitions in their respective frames over any two hour period. Edges connect compatible elements from one hour to the next. DELTA-LOCATIONS indicates that the zones are linearly ordered and that a ship must pass through the channel to get to either the loading or refueling docks. It also indicates that a ship will remain at the refueling dock or in the channel

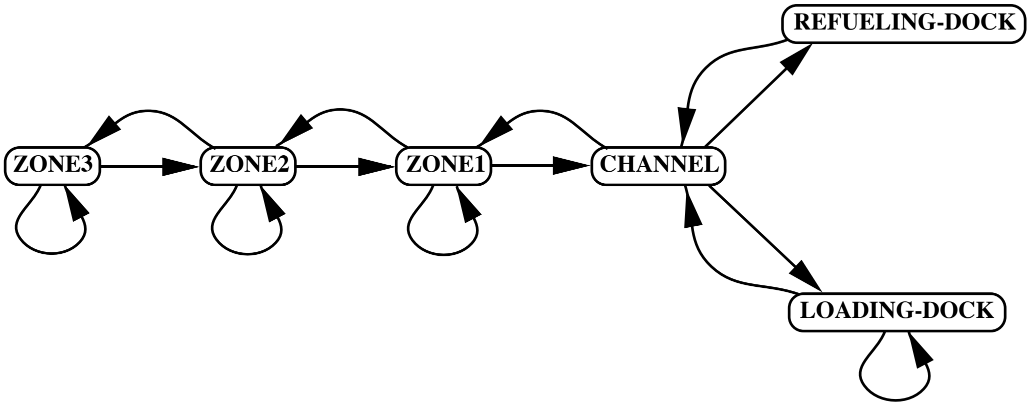


Figure 2.4: DELTA-LOCATIONS compatibility relation

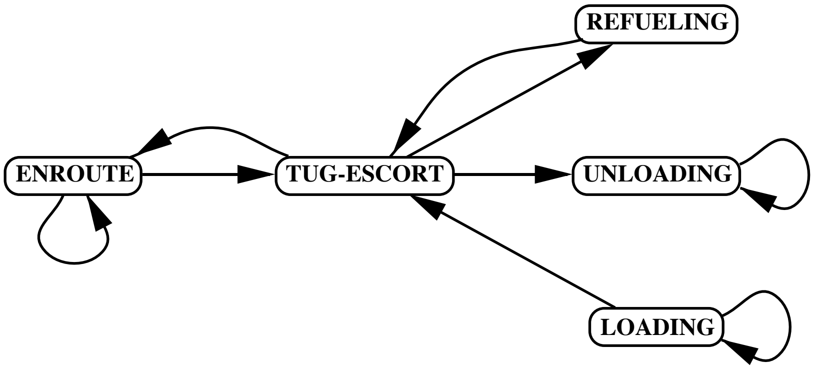


Figure 2.5: DELTA-ACTIVITIES compatibility relation

only for one hour at a time but may remain anywhere else for any number of hours. In DELTA-ACTIVITIES it can be seen that a ship must progress through TUG-ESCORT from ENROUTE before proceeding to REFUELING or UNLOADING and that REFUELING and TUG-ESCORT are one-hour activities. Further, a ship must go through LOADING after UNLOADING before returning to TUG-ESCORT.

2.1.3 Galleries

The overall topology, formed by frames and compatibility relations, is represented as a graph, called a *gallery*. In a gallery the frames are represented as nodes and the compatibility relations are represented as edges connecting the frames they relate. In Figure 2.6, the LOCATIONS and ACTIVITIES frames are represented as nodes and the LOCATIONS-ACTIVITIES compatibility is represented as an edge connecting the two frames. The other two compatibility relations, DELTA-LOCATIONS and DELTA-ACTIVITIES, relate frames to themselves and are represented by edges that begin and end at the same node.

Figure 2.6: SHIP-GALLERY gallery

2.2 Analyzing the Evidence

Once a gallery has been established, Gister-CL can analyze the available evidence. The goal of this analysis is to establish a line of reasoning, based upon both the possibilistic information in the gallery and the probabilistic information from the evidence, that determines the most likely answers to some questions. The gallery delimits the space of possible situations, and the evidential information establishes the likelihoods of these possibilities. Within an analysis, bodies of evidence are expressed relative to frames in the gallery, and paths are established for the bodies of evidence to move through the frames via the compatibility mappings. An analysis also specifies if other evidential operations are to be performed, including whether multiple bodies of evidence are to be combined when they arrive at common frames. Finally, an analysis specifies which frame and ultimate bodies of evidence are to be used to answer each target question. Thus, an analysis specifies a means for arguing from multiple bodies of evidence toward a particular (probabilistic) conclusion. An analysis, in an evidential context, is the analogue of a proof tree in a logical context.

2.2.1 Mass Distributions

To begin, each body of evidence is expressed relative to a frame in the gallery. Each is represented as a *mass distribution* (e.g., m_A) over propositional statements discerned by a frame (e.g., Θ_A):

$$\begin{aligned} m_A : 2^{\Theta_A} &\mapsto [0,1] \\ \sum_{A_i \subseteq \Theta_A} m_A(A_i) &= 1 \\ m_A(\emptyset) &= 0 \ . \end{aligned}$$

Intuitively, mass is attributed to the most precise propositions a body of evidence supports. If a portion of mass is attributed to a proposition A_i, it represents a minimal commitment to that proposition and all the propositions it implies. Additional mass attributed to a proposition A_j that is compatible with A_i, but does not imply it (i.e., $\emptyset \neq A_i \cap A_j \neq A_j$), represents a potential commitment: mass that neither supports nor denies that proposition at present but that might later move either way based upon additional information.

2.2.2 Interpretation

To *interpret* this body of evidence relative to the question A_j, we calculate its *support*[1] and *plausibility* to derive its *evidential interval* as follows:

$$\begin{aligned} Spt(A_j) &= \sum_{A_i \subseteq A_j} m_A(A_i) \\ Pls(A_j) &= 1 - Spt(\Theta_A - A_j) \\ [Spt(A_j), Pls(A_j)] &\subseteq [0,1] \ . \end{aligned}$$

The lower bound of an evidential interval indicates the degree to which the evidence supports the proposition, while the upper bound indicates the degree to which the evidence fails to refute the proposition, i.e., the degree to which it remains plausible. This evidential interval, for the most part, corresponds to bounds on the probability of A_j. Thus, complete ignorance is represented by an evidential interval of $[0.0, 1.0]$ and a precise probability assignment is represented by the "interval" collapsed about that point (e.g., $[0.7, 0.7]$). Other degrees of ignorance are captured by evidential intervals with widths other than 0 or 1 (e.g., $[0.6, 0.8], ; , [0.0, 0.5], ; , [0.9, 1.0]$).

Propositional statements that are attributed nonzero mass are called the *focal elements* of the distribution. When a mass distribution's focal elements are all single element sets, the distribution corresponds to a classical *additive* probability distribution and the evidential interval, for any proposition discerned by the frame, collapses to a point: support is equivalent to plausibility. For any other choice of focal elements, some propositional statement discerned by the frame will have an evidential interval with support strictly less than plausibility. This reflects the fact that mass attributed to a set consisting of more than one element represents an incomplete assessment; if additional information were available, the mass attributed to this set of elements would be distributed over its single element subsets. Thus, an evidential interval with support strictly less than plausibility is indicative of incomplete information relative to the frame. A mass distribution that has the frame as its sole focal element, is said to be *vacuous*; an evidential interval of$[0.0, 1.0]$ is attributed to all propositions in the frame other than the proposition corresponding to the frame itself.

2.2.3 Discounting

Discounting is an evidential operation that adjusts a mass distribution to reflect its source's credibility (expressed as a discount rate $r \in [0, 1]$). If a source is completely reliable ($r = 0$), discounting has no effect; if it is completely unreliable ($r = 1$), discounting strips away all apparent information content; otherwise, discounting lowers the apparent information

---

[1] What we call *support* and represent by $Spt(A_j)$ is often called *belief* and represented by $Bel(A_j)$ by others. In [Sha76a], two distinct vocabularies were defined: the subjective vocabulary consisted of belief, upper probability, and doubt; the evidential vocabulary consisted of support, plausibility, and dubiety. Very early on, we choose to use the evidential vocabulary. Although today the two most frequently used terms in the literature are belief and plausibility, we have chosen to stick with the original evidential vocabulary.

content in proportion to the source's unreliability. It has the effect of widening the evidential intervals, reflecting increased ignorance. Discounting is defined as follows:

$$m_A^{\%}(A_i) = \begin{cases} (1-r)\, m_A(A_i), & A_i \neq \Theta_A \\ r + (1-r)\, m_A(\Theta_A), & \text{otherwise} \end{cases} .$$

2.2.4 Translation and Projection

If a body of evidence is to be interpreted relative to a question expressed over a frame other than the one over which the evidence is expressed, a path of compatibility relations connecting the two frames is required. The mass distribution expressing the body of evidence is then repeatedly *translated* from frame to frame, via compatibility mappings, until it reaches the ultimate frame of the question. In translating m_A from frame Θ_A to frame Θ_B via compatibility mapping $\Gamma_{A\mapsto B}$, the following computation is applied to derive the translated mass distribution m_B:

$$\begin{aligned} m_B(B_j) &= \frac{1}{1-\kappa} \sum_{\Gamma_{A\mapsto B}(A_i)=B_j} m_A(A_i) \\ \kappa &= \sum_{\Gamma_{A\mapsto B}(A_i)=\emptyset} m_A(A_i) \\ &< 1 \ . \end{aligned}$$

Intuitively, if we (partially) believe A_i, and A_i implies B_j, then we should (partially) believe B_j; if some focal element A_i is incompatible with every element in Θ_B, then there is *conflict* (i.e., κ) between the evidence and the logic of the frames and compatibility relation. As we will see later, this is equivalent to the conflict in Dempster's rule. This method is also applied to move mass distributions among frames that represent states of the world at different times; however, when this is the case, the operation is called *projection*.

2.2.5 Fusion

Once two mass distributions m_A^1 and m_A^2 representing independent opinions are expressed relative to the same frame of discernment, they can be *fused* (i.e., combined) using *Dempster's Rule of Combination*. Dempster's rule pools mass distributions to produce a new mass distribution m_A^3 that represents the consensus of the original disparate opinions. That is, Dempster's rule produces a new mass distribution that leans toward points of agreement between the original opinions and away from points of disagreement. Dempster's rule is defined as follows:

$$\begin{aligned} m_A^3(A_k) &= m_A^1 \oplus m_A^2(A_k) \\ &= \frac{1}{1-\kappa} \sum_{A_i \cap A_j = A_k} m_A^1(A_i) m_A^2(A_j) \end{aligned}$$

$$\begin{aligned}\kappa &= \sum_{A_i \cap A_j = \emptyset} m_A^1(A_i) m_A^2(A_j) \\ &< 1 \ .\end{aligned}$$

Because Dempster's rule is both commutative and associative, multiple (independent) bodies of evidence can be combined in any order without affecting the result. If the initial bodies of evidence are independent, then the derivative bodies of evidence are independent as long as they share no common ancestors. Thus, in the course of constructing an analysis, we must take care that evidence is propagated and combined in such a way as to guarantee the independence of the evidence at each combination. Gister-CL protects the user by tracking the evidence and preventing such dependent combinations.

The *conflict* (i.e., κ) generated during the application of Dempster's rule quantifies the degree to which the mass distributions being combined are incompatible, that is, the degree to which the two distributions are directly contradictory. When $\kappa = 1$, the distributions are in direct and complete contradiction to one another, no consensus exists, and Dempster's rule is undefined; when $\kappa = 0$, there is no contradiction and the evidential intervals based upon the consensus distribution will be contained within the bounds of the evidential intervals based upon the component distributions, i.e., the combination is *monotonic*; otherwise, the component distribution are partially contradictory. In this case, Dempster's rule focuses the consensus on the compatible portions of the component distributions by eliminating the contradictory portions and normalizing what remains; some evidential intervals based upon the consensus distributions will not fall within the bounds of intervals based upon the component distributions, i.e., the combination is *nonmonotonic.*

In general, there are three possible sources of conflict. (1) One or both of the component distributions might simply be in error e.g., due to a miscalculation or malfunction during information collection. (2) The component distributions might not be referring to the same aspect of the environment; for example, one distribution might be referring to the location of one ship while the other is referring to the location of a different ship. (3) The frame might not properly capture the range of possibilities: e.g., one distribution might indicate that a ship is docked, the other might indicate that the ship is in the channel, and, mistakenly, the locations at which a ship might be docked exclude the channel.

2.2.6 Summarization

Another useful evidential operation is *summarization*, which attempts to eliminate extraneous details from a mass distribution, thereby reducing its size and potentially reducing the computational costs in subsequent operations. Summarization collects all of the extremely small amounts of mass (determined by a threshold $t \in [0, 1]$) attributed to propositions, then attributes the sum to the disjunction of those propositions. The resulting mass distribution is slightly less informative than the original in that some evidential intervals based upon this resulting mass distribution will be wider than those based upon the original, but it remains consistent with the original in that the intervals based on the resulting distribution contain those based on the original. Thus:

$$\begin{aligned} m_A^+(\mathsf{A}_i) &= \begin{cases} m_A(\mathsf{A}_i), & \mathsf{A}_i \neq \mathsf{S} \\ s + m_A(\mathsf{S}), & \text{otherwise} \end{cases} \\ \mathsf{S} &= \bigcup_{0 < m_A(\mathsf{A}_i) < t} \mathsf{A}_i \\ s &= \sum_{0 < m_A(\mathsf{A}_i) < t} m_A(\mathsf{A}_i) \quad . \end{aligned}$$

Another form of summarization can be defined in terms of translation. If mass is distributed over a fairly fine-grained frame, i.e., a frame with a large number of elements because it preserves subtle distinctions, but the question at hand could be resolved in the context of a coarser frame, i.e., one with fewer elements that makes fewer distinctions, and the coarser is related to the finer by means of a compatibility relation, then the mass distribution can be translated to the coarser frame to reduce its complexity. As with the thresholded summarization operation defined above, the resulting mass distribution is generally less informative but consistent, so long as the gallery is well formed. Thus, the first summarization operation defined above discards information that will likely have a miniscule impact on future questions, while the second summarization technique discards details that are irrelevant to a particular set of questions, i.e., the set of propositions discerned by the frame.

2.2.7 Gisting

For some applications it is important to communicate with the user without making use of probabilities: the user may require only a summary of the components of an evidential argument and a rough description of what has been concluded and why. At times the masking of probabilities is required because the user lacks an adequate background in probabilities; other times it is required because the selected means of displaying the results cannot accommodate probabilistic answers; still other times it is simply a useful means for even a sophisticated user to quickly assess the lay of the probabilistic landscape.

To support this nonprobabilistic assessment of evidential arguments, we have developed *Gisting*, which produces a statement that attempts to capture the essence of a mass distribution. In other words, it attempts to summarize the contents of a body of evidence in terms of a single statement from the frame, void of any uncertainty. Such a summary is particularly useful when explaining lines of reasoning. As defined below, the gist of a mass distribution is the most pointed statement from the frame whose support meets or exceeds a selected level. This definition uses cardinality as a measure of specificity (i.e., pointedness) and thereby assumes that all elements of Θ_A are equally specific. It also ignores problems of instability stemming from small variations in the support, specificity, and the gist level. The gist G of a mass distribution, m_{A}, is defined relative to a gist level, $g \in [0, 1]$:

$$\begin{aligned} \mathsf{G} &= \bigcup_{\mathsf{A}_i \in G} \mathsf{A}_i \\ G &\subseteq 2^{\Theta_A} \quad , \end{aligned}$$

for all $\mathsf{A}_i, \mathsf{A}_j \in G, \mathsf{A}_k \notin G$

$$\begin{array}{rcl} Spt(\mathsf{A}_i) = Spt(\mathsf{A}_j) & \geq & g \\ |\mathsf{A_i}| & = & |\mathsf{A_j}| \\ Spt(\mathsf{A}_i) > Spt(\mathsf{A}_k) & or & |\mathsf{A_k}| > |\mathsf{A_i}| \quad . \end{array}$$

When gisting is used to follow an evolving argument, as new evidence is considered, the gists frequently change in ways that are reminiscent of Boolean valued nonmonotonic reasoning systems. The relationship between evidential gisting and nonmonotonic logics has yet to be formally explored.

2.2.8 Decisions

Evidential analyses are constructed to support decision making. Once all of the available evidence has been properly incorporated into an analysis, it can be used to select the most credible statement from among several alternative statements about the domain of application. When choosing between two statements, A_i and A_j, the decision maker considers their evidential intervals with respect to a single body of evidence. This body of evidence typically is the ultimate result of the analysis, but may be some other intermediate result. If the evidential intervals are nonoverlapping, then the best choice is the proposition with the superior interval (i.e., choose A_i if $Pls(A_j) < Spt(A_i)$; choose A_j if $Pls(A_i) < Spt(A_j)$). If the intervals are overlapping, then the best answer is that there is no clear choice and additional information should be factored into the analysis to separate the intervals.

If two evidential intervals remain overlapping after all evidence has been considered, or other factors prevent the consideration of additional evidence (e.g., the decision cannot be further delayed due to time constraints), then one must turn from consideration of evidence to consideration of the decision maker's preferences. In an evidential decision theory [Str90], this decision maker's *bias* is represented by a single coefficient b. Based on this coefficient, a representative likelihood is selected from each interval and the proposition with the greatest representative likelihood is chosen. When this coefficient is 0.5, the midpoints of the intervals serve as their representative likelihoods; as the coefficient decreases to zero, the representative likelihoods approach the support values; as the coefficient increases to one, the likelihoods approach the plausibilities. The representative likelihood l for a proposition A_i is defined relative to the bias b as follows:

$$l = Spt(A_i) + b(\,Pls(A_i) - Spt(A_i)\,) \quad .$$

Still other evidential operations allow an established analysis to be examined and revised. These operations are not included in analyses, but are used as tools to assist in their development.

2.2.9 Specificity and Consonance

There have been numerous proposals for characterizing bodies of evidence that can be used as the basis for forming informative explanations. However, we have found some due to Yager to be particularly useful [Yag83].

The theory of evidence supports varying degrees of precision as well as uncertainty in the representation of a body of evidence. The relative precision of a mass distribution m_A can be characterized by the following expression for its *specificity*:

$$Spec(m_A) = \sum_{A_i \subseteq \Theta_A} \frac{m_A(A_i)}{|A_i|} \quad ,$$

where $|A_i|$ is the cardinality of the subset A_i. It is easy to show that

$$0 < \frac{1}{|\Theta_A|} \leq Spec(m_A) \leq 1 \quad ,$$

for any mass distribution m_A. Roughly speaking, $Spec(m_A)$ measures the degree of commitment of a mass distribution to precise propositions, assuming that each element of Θ_A is equally precise. The vacuous mass distribution has the smallest possible specificity, while an additive mass distribution has specificity 1.

The relative uncertainty of a body of evidence can be characterized by an entropy-like measure. Yeager defines

$$Ent(m_A) = \sum_{A_i \subseteq \Theta_A} m_A(A_i)\, lnPls(A_i) \quad ,$$

and shows that $Ent(m_A)$ is just Shannon's measure of entropy in the special case when m_A is an additive distribution. To use this measure in generating explanations, it is more convenient to work with a measure of *consonance*:

$$Cons(m_A) = \frac{1}{1 + Ent(m_A)} \quad ,$$

so that

$$0 < Cons(m_A) \leq 1 \quad .$$

Minimal consonance is thus maximal entropy (i.e., *dissonance*), and exists when the focal elements of a mass distribution are mutually exclusive. Consonance equal to 1 occurs when all the focal elements are nested; such mass distributions are said to be *consonant* [Sha76a]. All the evidence embodied in a consonant mass distribution points in the same direction, and it is internally heterogeneous only to the extent that it varies in its precision.

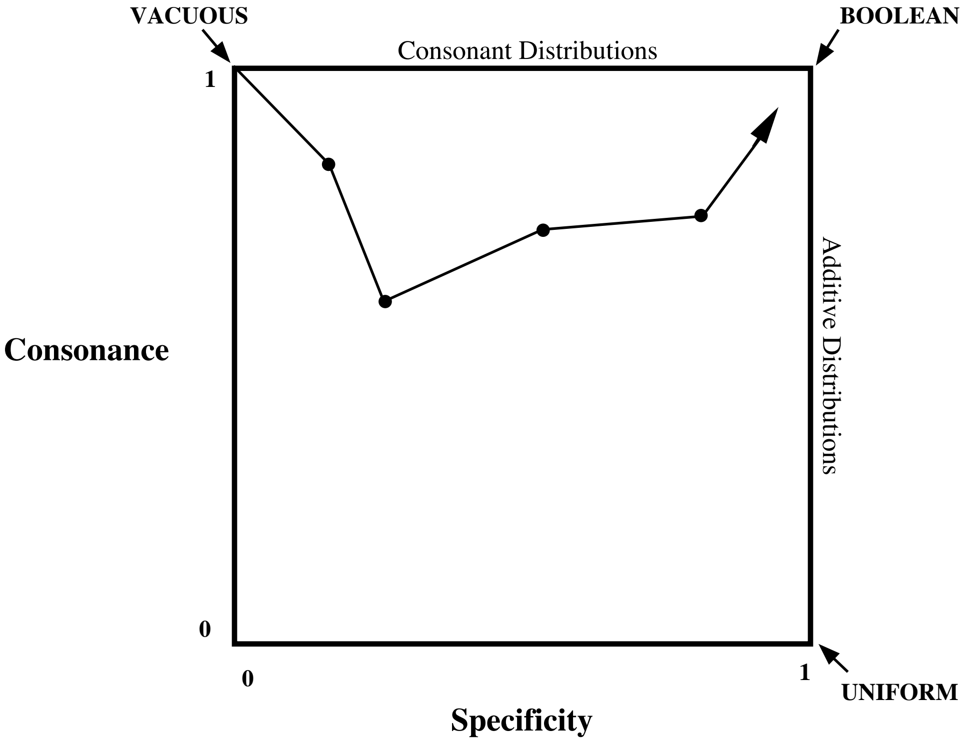


Figure 2.7: Specificity-consonance plot

Utilizing these two measures, specificity and consonance, one can come to better understand the role of each body of evidence in combination, by observing how these measures for the original bodies of evidence differ from the measures for the fused result. If specificity rises then the combination is focusing on more specific results; if it falls the combination is becoming more diffuse. If consonance increases then the bodies of evidence tend to agree; if it decreases, then they tend to disagree. If the addition of a body of evidence leaves these measures relatively unchanged, then the new evidence is largely irrelevant.

To gain some intuition, it is useful to note that any mass distribution is characterized by a point in the unit square shown in Figure 2.7. The special cases of consonant distributions and additive distributions lie on the boundaries of the square. A Boolean valued statement has $Cons(m_A) = Spec(m_A) = 1$. The vacuous distribution has $Cons(m_A) = 1$ and $Spec(m_A) = 0$ and is represented by the upper-left corner of the square. A uniform distribution over the elements of the frame is represented by the lower-right corner. Starting with no information and gradually fusing new pieces of evidence as they become available, we trace a path in the square that starts at the upper-left corner and wanders through the space. The ideal analysis would reach a Boolean conclusion (upper-right corner), but typically the path stops somewhere short. The intuition, then, is that pieces of evidence that move the path closer to the upper-right corner are the most important ones for making decisions.

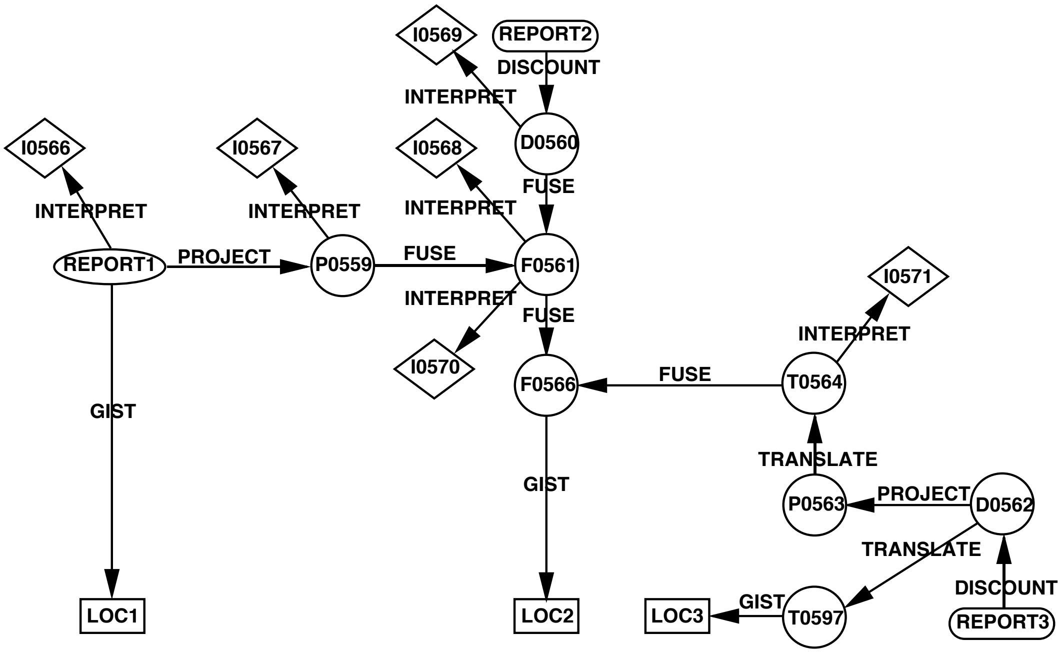
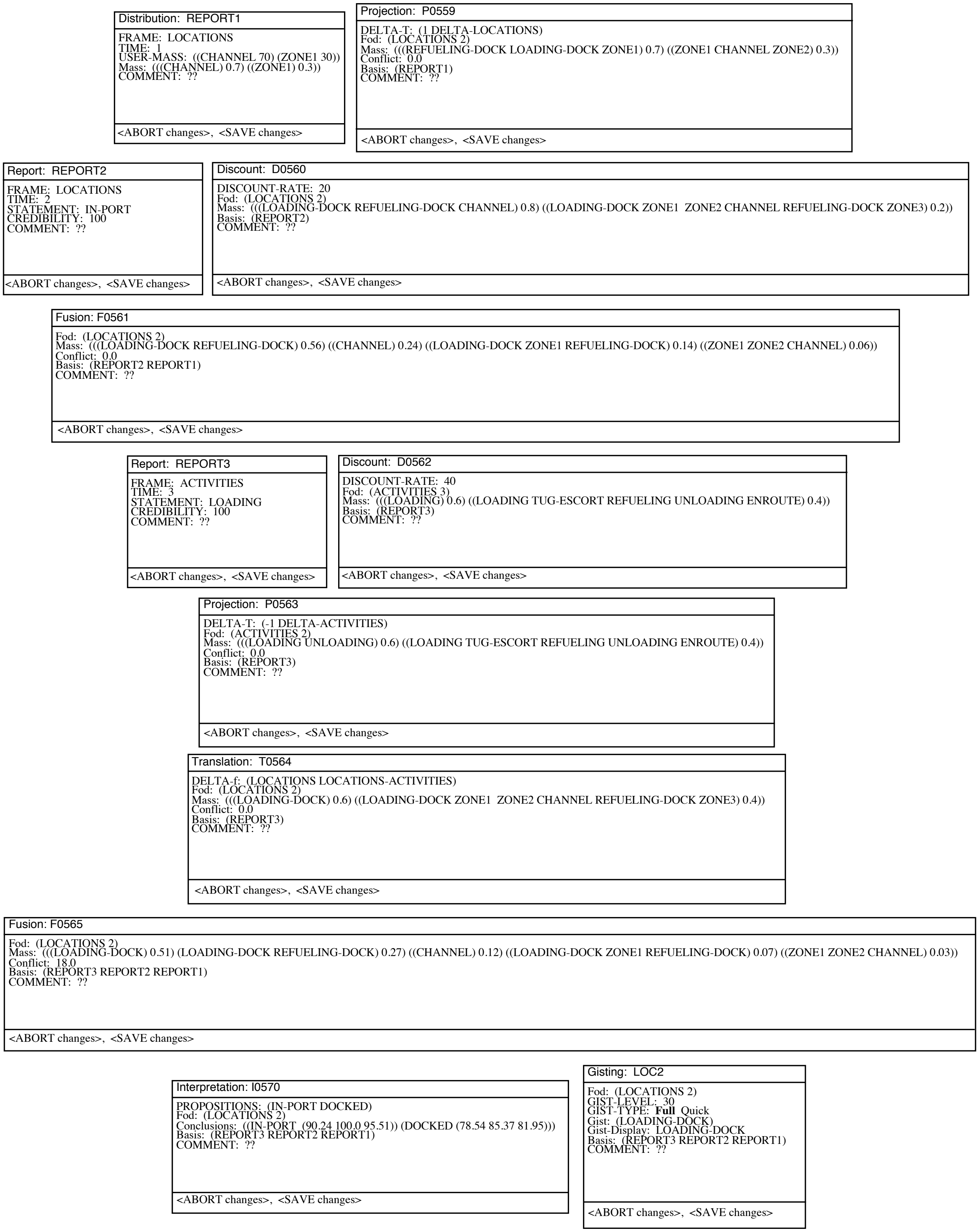


Figure 2.8: ANALYSIS1 analysis

2.2.10 Analyses

After the gallery and its supporting frames and compatibility relations have been established, evidential analyses can be constructed. These *analyses* are represented as data-flow graphs where the data and the operations are evidential. Figure 2.8 is one such analysis. Here primitive bodies of evidence are represented by elliptical nodes, and derivative bodies of evidence are represented by circular nodes. Diamond-shaped nodes represent interpretations of bodies of evidence and rectangular nodes represent gists of bodies of evidence. The values of these nodes are used as repositories for the information (i.e., data) that they represent (Figure 2.9). For bodies of evidence, this information includes a frame of discernment (including the time to which the evidence pertains), a mass distribution, and other supporting information. Edges pointing to a derivative node are labeled with the evidential operation that is applied to the bodies of evidence, at the other ends of the edges, to derive the body of evidence represented by this node.

In the analysis of a ship in Figures 2.8 and 2.9, there are three primitive bodies of evidence. REPORT1 locates the ship at time 1, saying that there is a 70 percent chance that it can be found in the CHANNEL and a 30 percent chance that it is in ZONE1; REPORT2 says that the ship was IN-PORT at time 2; and REPORT3 indicates that the ship was LOADING cargo at time 3. REPORT1 is taken at face value, but REPORT2 and REPORT3 have been discounted by 20 percent and 40 percent, respectively, to derive D0560 and D0562, reflecting doubt in the credibility of these reports. REPORT1 has been projected forward by one hour to derive P0559 and then fused with discounted REPORT1 (i.e., D0560) to derive F0561. D0562 (i.e., discounted REPORT2) has been projected backward in time by one hour to derive P0563 and then has been translated from the



```
Distribution:  REPORT1
FRAME:  LOCATIONS
TIME:  1
USER-MASS:  ((CHANNEL 70) (ZONE1 30))
Mass:  (((CHANNEL) 0.7) ((ZONE1) 0.3))
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Projection:  P0559
DELTA-T:  (1 DELTA-LOCATIONS)
Fod:  (LOCATIONS 2)
Mass:  (((REFUELING-DOCK LOADING-DOCK ZONE1) 0.7) ((ZONE1 CHANNEL ZONE2) 0.3))
Conflict:  0.0
Basis:  (REPORT1)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Report:  REPORT2
FRAME:  LOCATIONS
TIME:  2
STATEMENT:  IN-PORT
CREDIBILITY:  100
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Discount:  D0560
DISCOUNT-RATE:  20
Fod:  (LOCATIONS 2)
Mass:  (((LOADING-DOCK REFUELING-DOCK CHANNEL) 0.8) ((LOADING-DOCK ZONE1  ZONE2 CHANNEL REFUELING-DOCK ZONE3) 0.2))
Basis:  (REPORT2)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Fusion: F0561
Fod:  (LOCATIONS 2)
Mass:  (((LOADING-DOCK REFUELING-DOCK) 0.56) ((CHANNEL) 0.24) ((LOADING-DOCK ZONE1 REFUELING-DOCK) 0.14) ((ZONE1 ZONE2 CHANNEL) 0.06))
Conflict:  0.0
Basis:  (REPORT2 REPORT1)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Report:  REPORT3
FRAME:  ACTIVITIES
TIME:  3
STATEMENT:  LOADING
CREDIBILITY:  100
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Discount:  D0562
DISCOUNT-RATE:  40
Fod:  (ACTIVITIES 3)
Mass:  (((LOADING) 0.6) ((LOADING TUG-ESCORT REFUELING UNLOADING ENROUTE) 0.4))
Basis:  (REPORT3)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Projection:  P0563
DELTA-T:  (-1 DELTA-ACTIVITIES)
Fod:  (ACTIVITIES 2)
Mass:  (((LOADING UNLOADING) 0.6) ((LOADING TUG-ESCORT REFUELING UNLOADING ENROUTE) 0.4))
Conflict:  0.0
Basis:  (REPORT3)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Translation:  T0564
DELTA-f:  (LOCATIONS LOCATIONS-ACTIVITIES)
Fod:  (LOCATIONS 2)
Mass:  (((LOADING-DOCK) 0.6) ((LOADING-DOCK ZONE1  ZONE2 CHANNEL REFUELING-DOCK ZONE3) 0.4))
Conflict:  0.0
Basis:  (REPORT3)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Fusion: F0565
Fod:  (LOCATIONS 2)
Mass:  (((LOADING-DOCK) 0.51) (LOADING-DOCK REFUELING-DOCK) 0.27) ((CHANNEL) 0.12) ((LOADING-DOCK ZONE1 REFUELING-DOCK) 0.07) ((ZONE1 ZONE2 CHANNEL) 0.03))
Conflict:  18.0
Basis:  (REPORT3 REPORT2 REPORT1)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Interpretation: I0570
PROPOSITIONS:  (IN-PORT DOCKED)
Fod:  (LOCATIONS 2)
Conclusions:  ((IN-PORT  (90.24 100.0 95.51)) (DOCKED (78.54 85.37 81.95)))
Basis:  (REPORT3 REPORT2 REPORT1)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>

Gisting:  LOC2
Fod:  (LOCATIONS 2)
GIST-LEVEL:  30
GIST-TYPE:  Full  Quick
Gist:  (LOADING-DOCK)
Gist-Display:  LOADING-DOCK
Basis:  (REPORT3 REPORT2 REPORT1)
COMMENT:  ??
<ABORT changes>,  <SAVE changes>
```

Figure 2.9: Data from ANALYSIS1

ACTIVITIES frame to the LOCATIONS frame. Finally, this result, T0564, has been fused with F0561 to derive a consensus (i.e., F0566), based on all three reports, about the ship's location at time 2.

To track the "gist" of this argument, three gisting nodes have been created. LOC1 is simply the gist of REPORT1; LOC2 is the gist of the consensus about the ship's location at time 2; LOC3 is the gist of REPORT3 after it has been discounted and translated to the LOCATIONS frame. These indicate that the ship is in the CHANNEL at time 1 and at the LOADING-DOCK at times 2 and 3.

The interpretation nodes in this analysis track the evidential intervals for some key propositions. I0566 is based solely on REPORT1 and indicates that there is precisely a 70 percent chance of the ship being IN-PORT $[0.7, 0.7]$ and no chance of it being DOCKED $[0.0, 0.0]$ at time 1. I0567 indicates that, based solely upon REPORT1, after one hour has elapsed, nothing is known about whether the ship is IN-PORT $[0.0, 1.0]$, but that it may now be DOCKED $[0.0, 0.7]$. If REPORT2 is included after being discounted, I0568 indicates that there is strong reason to believe that the ship is IN-PORT $[0.8, 1.0]$, but there is conflicting information concerning whether or not it is DOCKED $[0.56, 0.7]$. I0571 indicates that, based solely upon REPORT3, after having been discounted, projected backward an hour, and translated to the LOCATION frame, that there is 0.6 support and 1.0 plausibility for both IN-PORT and DOCKED. Finally, when all three reports are considered, I0570 indicates strong belief that the ship is IN-PORT $[0.9, 1.0]$ at time 2 and a reasonably strong belief, though mixed, that it is also DOCKED $[0.78, 0.85]$.

2.3 Exploring Alternative Arguments

Of course, the example above is not the only argument that can be constructed from these data. For example, the credibility given to the initial reports might be assessed differently. To explore such alternatives using Gister-CL , the user has only to modify the discount rates stored on the appropriate discount nodes in ANALYSIS1. In response, Gister-CL recalculates the dependent conclusions. For example, if the discount rate applied to REPORT3 at node D0562 is reduced from 40 percent to 10 percent, then Gister-CL recalculates all of the dependent nodes, including D0562, T0597, LOC3, P0563, T0564, I0571, F0565, I0570, and LOC2. As a result, the interpretation (i.e., I0571) of REPORT3 after having been discounted, projected backwards, and translated, gives greater support (from 0.6 to 0.9) to the ship being DOCKED and IN-PORT at time 3. Similarly, the interpretation (i.e., I0570) of the consensus for the ship's location at time 2 more heavily favors both DOCKED (from $[0.79, 0.85]$ to $[0.94, 0.95]$) and IN-PORT (from $[0.9, 0.95]$ to $[0.97, 1.0]$).

Alternatively, the user might decide to develop a parallel line of reasoning within ANALYSIS1, constructing a new sequence of evidential operations with different parameters to argue for the same or a different conclusion. Or a completely new analysis under a different name might be constructed. Consider the analysis in Figure 2.10. The result for time 2 (i.e, F0565) is derived as in ANALYSIS1 except that REPORT3 has been translated then projected, instead of having been projected then translated. As a result, the information that the ship was LOADING at time 3 supports the conclusion that it was

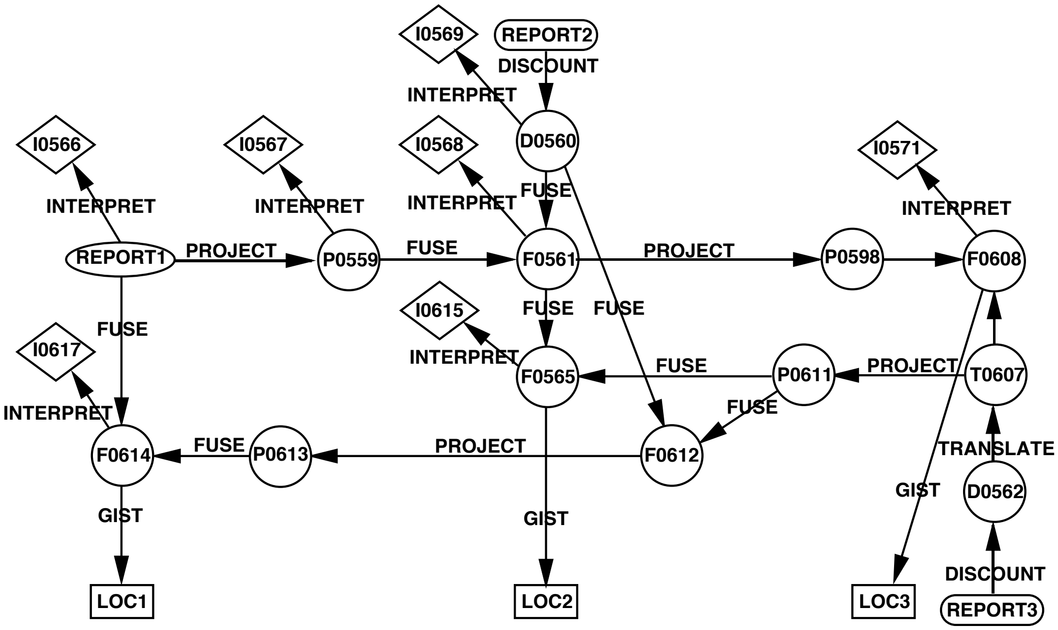


Figure 2.10: An alternative to ANALYSIS1

located at the LOADING-DOCK. In turn, this supports the conclusion that the ship was in the CHANNEL or at the LOADING-DOCK on the previous hour. This result is weaker than the result in ANALYSIS1 where LOADING is projected to UNLOADING and then translated to LOADING-DOCK. When this weaker conclusion is taken in combination with the results from the other reports at node F0565, CHANNEL is believed slightly more and LOADING-DOCK slightly less, although the gist for the hour remains unchanged.

In addition, variations in the gallery can be explored. Here, elements might be deleted or new elements added to frames and compatibility relations to determine their effect on established analyses. For example, REPAIR-DOCK might be added to LOCATIONS and DELTA-LOCATIONS (Figures 2.11 and 2.12), MAINTENANCE to ACTIVITIES and

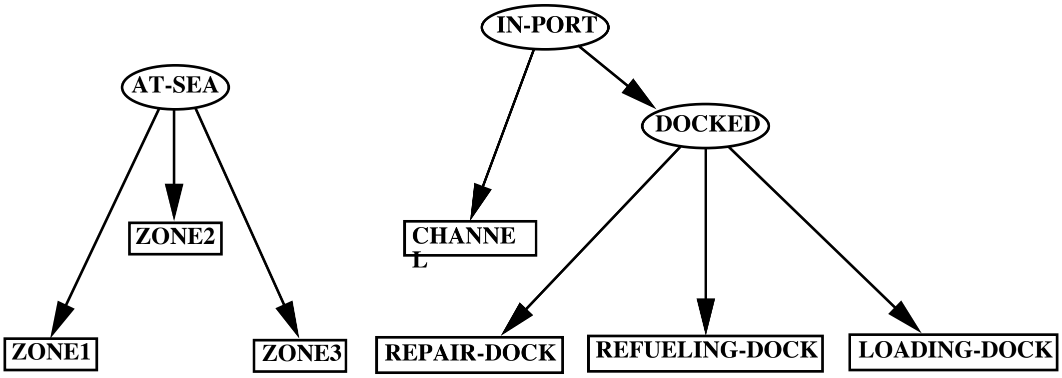


Figure 2.11: Modified LOCATIONS frame

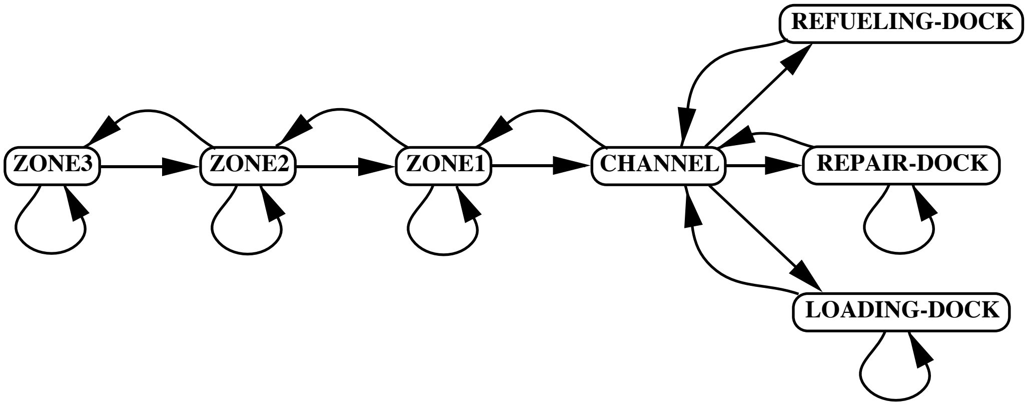


Figure 2.12: Modified DELTA-LOCATIONS compatibility relation

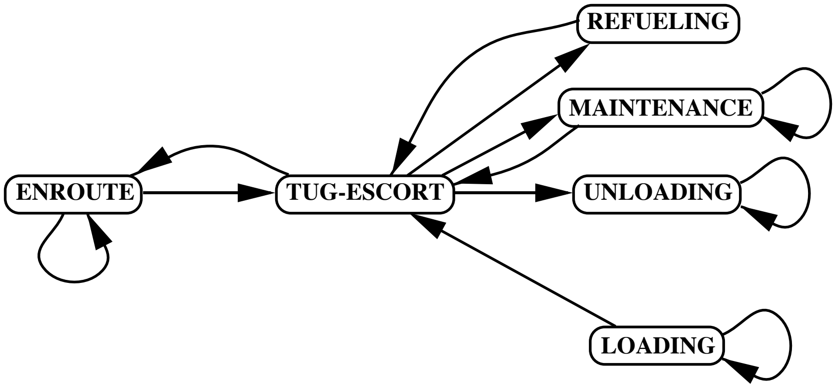


Figure 2.13: Modified DELTA-ACTIVITIES compatibility relation

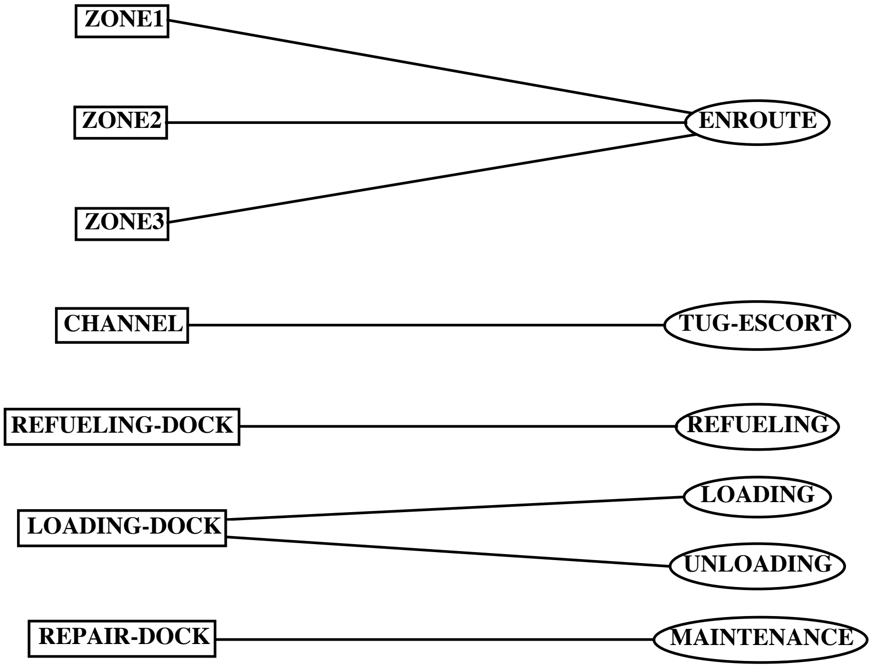


Figure 2.14: Modified LOCATIONS-ACTIVITIES compatibility relation

DELTA-ACTIVITIES (Figure 2.13), and both to LOCATIONS-ACTIVITIES (Figure 2.14). The impact on ANALYSIS1 is negligible since (1) the information that the ship was LOADING at time 3 still projects and translates to UNLOADING at the LOADING-DOCK for time 2, and (2) although REPAIR-DOCK at time 2 is now compatible with CHANNEL at time 1 (in addition to REFUELING-DOCK and LOADING-DOCK), it is eliminated as a possibility when taken in combination with the LOADING-DOCK conclusion derived from REPORT3. Alternatively, different frames and compatibility relations might be constructed, after which new analyses might be created or old analyses modified to examine the data relative to a set of different possibilistic assumptions.

Gister-CL separates the specific data about the current situation, from the general knowledge of what is possible, from the way in which this data and knowledge are utilized to draw conclusions. Further, it allows these to be independently varied, giving the user the freedom to create and examine alternative formulations. Through this interaction, the user comes to better understand the basis and sensitivities of his arguments and conclusions.

In essence, our approach reflects the view that there is not so much a correct argument, but rather a set of alternative competing arguments that must be explored.

Chapter 3

Advanced Argument Construction

3.1 Advantages of Multiframe Reasoning

Recall that two frames related by a compatibility relation can be converted into a single, more complex, frame. Conversely, a complex frame can be decomposed into multiple frames related by compatibility relations. However, this multiframe gallery may or may not fully capture the information in the original complex frame. If, at best, multiframe evidential reasoning is the equivalent of single-frame evidential reasoning, why pursue the multiframe approach? The answer is two fold: there are both conceptual and computational advantages to the multiframe approach.

Given a complex problem, principles of modular design suggest that it is better to solve and assemble solutions to several subproblems, than to try and address the complex problem directly. Each subproblem has a corresponding smaller frame requiring the designer to simultaneously address fewer concepts.

An often criticized aspect of evidential reasoning, and more generally the theory of belief functions, is its computational complexity [Bar81, GS85]. Whereas an additive probability distribution over a frame contains no greater number of nonzero probability assignments than there are elements in the frame (n), a mass distribution conceivably might assign mass to every nonempty subset of the frame ($2^n - 1$). Using Dempster's rule to combine two such mass distributions results in $(2^n - 1)^2$ computations. Thus, by dividing a frame into several smaller frames, we can gain an exponential reduction in the number of computations. At times this is the key to overall computational feasibility, while at other times, characteristics of the application domain keep the required computation far below this theoretical maximum.

3.2 Omnidirectional Evidential Reasoning

The central question to be addressed is: How can we benefit from the psychological and computational advantages of multiframe evidential reasoning while retaining the validity of single-frame evidential reasoning? The answer is to divide complex problems into conditionally independent subproblems.

This solution, for belief functions, has been previously reported [SS86, Kon86, SSM87, SS88]. The essence of the approach is to partition complex problems into several smaller problems, thereby, substituting applications of Dempster's rule over smaller frames for applications over the original, larger frame. The validity of the results is guaranteed if the smaller frames and their connecting compatibility relations form a *Markov tree* and if the reported propagation algorithm is used to move belief functions through the tree. A parallel implementation of this algorithm, assuming a processor for each frame in the tree, was also proposed.

Our approach is to adapt this solution to the evidential reasoning framework. The concept of a Markov tree is used to structure the gallery, and the propagation algorithm is used to structure analyses.

3.2.1 Markov Galleries

Markov trees are based upon the idea of *qualitative conditional independence* , a nonprobabilistic (i.e., qualitative) concept of conditional independence. Consider three frames of discernment, Θ_A, Θ_B, and Θ_C, connected by two compatibility relations, $\Pi_{(A,B)}$, and $\Pi_{(B,C)}$.

$$\begin{array}{rcl}
\Theta_A & = & \{a_1, a_2, \ldots, a_n\} \\
\Theta_B & = & \{b_1, b_2, \ldots, b_m\} \\
\Theta_C & = & \{c_1, c_2, \ldots, c_l\} \\
\Pi_{(A,B)} & \subseteq & \Theta_A \times \Theta_B \\
\Pi_{(B,C)} & \subseteq & \Theta_B \times \Theta_C \quad .
\end{array}$$

Four compatibility mappings are defined by these two compatibility relations, $\Gamma_{A\mapsto B}$, $\Gamma_{B\mapsto A}$, $\Gamma_{B\mapsto C}$, and $\Gamma_{C\mapsto B}$.

$$\begin{array}{rcl}
\Gamma_{A\mapsto B} : 2^{\Theta_A} & \mapsto & 2^{\Theta_B} \\
\Gamma_{A\mapsto B}(A_k) & = & \{\, b_j \mid (a_i, b_j) \in \Pi_{(A,B)},\ a_i \in A_k \,\}
\end{array}$$

$$\begin{array}{rcl}
\Gamma_{B\mapsto A} : 2^{\Theta_B} & \mapsto & 2^{\Theta_A} \\
\Gamma_{B\mapsto A}(B_k) & = & \{\, a_i \mid (a_i, b_j) \in \Pi_{(A,B)},\ b_j \in B_k \,\}
\end{array}$$

$$\begin{array}{rcl}
\Gamma_{B\mapsto C} : 2^{\Theta_B} & \mapsto & 2^{\Theta_C} \\
\Gamma_{B\mapsto C}(B_k) & = & \{\, c_j \mid (b_i, c_j) \in \Pi_{(B,C)},\ b_i \in B_k \,\}
\end{array}$$

$$\begin{aligned}\Gamma_{C\mapsto B} : 2^{\Theta_C} &\mapsto 2^{\Theta_B} \\ \Gamma_{C\mapsto B}(C_k) &= \{\, b_i \mid (b_i, c_j) \in \Pi_{(B,C)},\, c_j \in C_k \,\} \quad .\end{aligned}$$

If the true relationship among these three frames is defined by

$$\Theta_{(A,B,C)} \subseteq \Theta_A \times \Theta_B \times \Theta_C \quad ,$$

then frames Θ_A and Θ_C are *qualitatively conditionally independent* given frame Θ_B, if and only if,

$$\begin{aligned}\Theta_{(A,B,C)} &= \{\, (a_i, b_j, c_k) \mid a_i \in \Theta_A,\, b_j \in \Gamma_{A\mapsto B}(\{a_i\}),\, c_k \in \Gamma_{B\mapsto C}(\{b_j\}) \,\} \\ &= \{\, (a_i, b_j, c_k) \mid c_k \in \Theta_C,\, b_j \in \Gamma_{C\mapsto B}(\{c_k\}),\, a_i \in \Gamma_{B\mapsto A}(\{b_j\}) \,\} \quad .\end{aligned}$$

Intuitively, if it is known which elements of Θ_B bound the truth, then knowing which elements of Θ_A bound the truth provides no additional information about the truth in Θ_C, and knowing which elements of Θ_C bound the truth provides no additional information about the truth in Θ_A.

Given a tree of frames and compatibility relations, it is a *Markov tree* if the following condition holds:

- For all frames, Θ_A, Θ_B, and Θ_C, connected by compatibility relations, $\Pi_{(A,B)}$, and $\Pi_{(B,C)}$, frames Θ_A and Θ_C are qualitatively conditionally independent given Θ_B.

If this same condition holds over a gallery of frames and compatibility relations, it is a *Markov gallery*. In essence, this condition guarantees that each frame in a gallery fully captures the interaction of its neighboring frames and that these local constraints are sufficient to discern the global interactions.

Figures 3.1, 3.2, and 3.3, represent a complete Markov gallery. This gallery discerns the relationships among a person's job, education, wealth, car, and dwelling. The jobs considered are limited to accountant, lawyer, carpenter, and store clerk; education is characterized by the type of school from which they last graduated, either primary, secondary, undergraduate, or graduate school; wealth is categorized as rich, comfortable, or poor; cars as foreign or domestic; dwelling as an apartment, condominium, townhouse, house, or mansion. The compatibility relations describe the range of possible combinations of these attributes; while these relationships may exclude some possible combinations, this gallery suffices to illustrate the techniques.

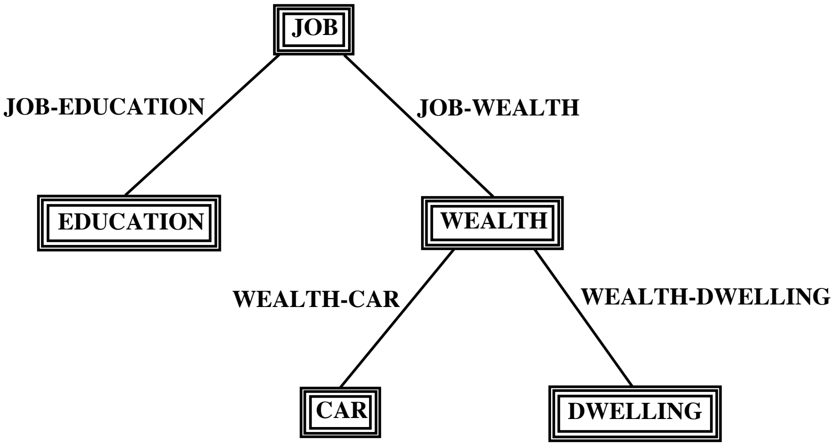


Figure 3.1: Markov gallery

3.2.2 Omnidirectional Analyses

Given a Markov gallery and independent evidence over each frame, we wish to calculate a consensus for each frame, taking all of the evidence into account. The first step is to select a Markov tree from the gallery. Since qualitative conditional independence holds throughout the gallery, any tree selected from that gallery is a Markov tree.

Select a frame Θ_i from the Markov tree (e.g., the JOB frame from Figure 3.1). Hanging from this frame are one or more subtrees, one rooted at each neighboring frame (i.e., the EDUCATION and WEALTH frames). Assuming that each of its neighboring frames already has an associated mass distribution representing the consensus of the evidence on the subtree rooted at that frame, the consensus for Θ_i can be calculated by translating each of these mass distributions from the subtrees to Θ_i and combining the results with the distribution at Θ_i. Because of the independence of the initial evidence and the qualitative conditional independence of the neighboring frames, we know that the interaction of the evidence from each subtree is limited to its interaction at Θ_i. If we remove Θ_i from the tree, we can apply this technique to each neighboring frame, to derive the consensus for each subtree. By recursively applying this technique, the evidence, starting at the leaves of the tree, is repeatedly translated and combined, until Θ_i is reached. This algorithm can be straightforwardly represented by an analysis, as illustrated in Figure 3.4 for calculating the result for the JOB frame; unlabeled arcs in this and the following figures are fusion arcs leading to fusion nodes.

A similar analysis can be constructed for each frame in a Markov tree. Collapsing these into a single analysis results in an omnidirectional analysis. The computational structure pertaining to a single frame in this analysis (shown in Figure 3.5), is a *compound node*[1], which consists of an input and output node for its associated frame (W.IN and W.OUT), and an input and output node for each neighboring frame in the tree (W.IN.1 and W.OUT.1,

---

[1] The concept of a compound node is used to help structure omnidirectional analyses; it is not a distinct evidential reasoning construct; compound nodes are simply assemblies of other evidential reasoning constructs.

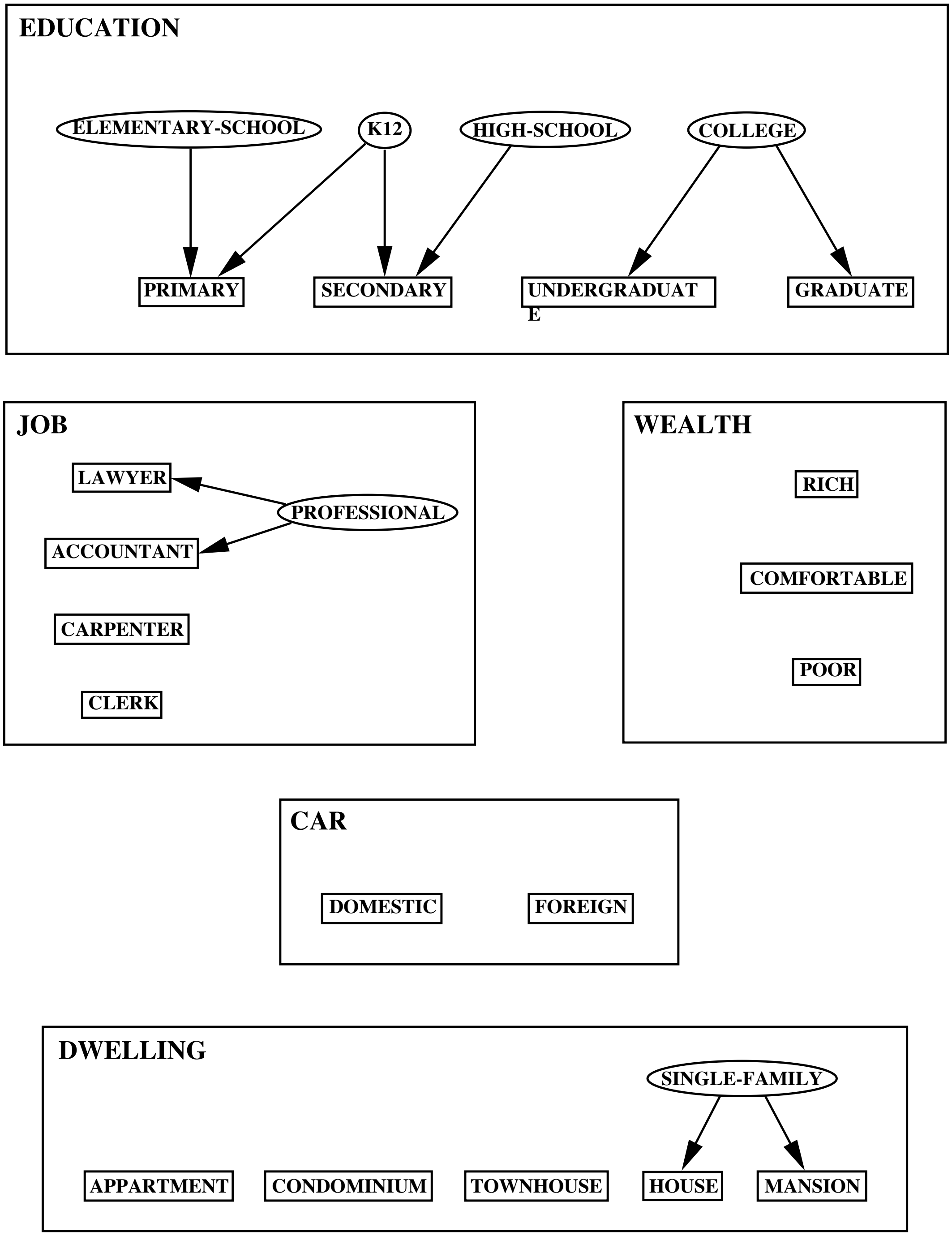


Figure 3.2: Frames from the Markov gallery

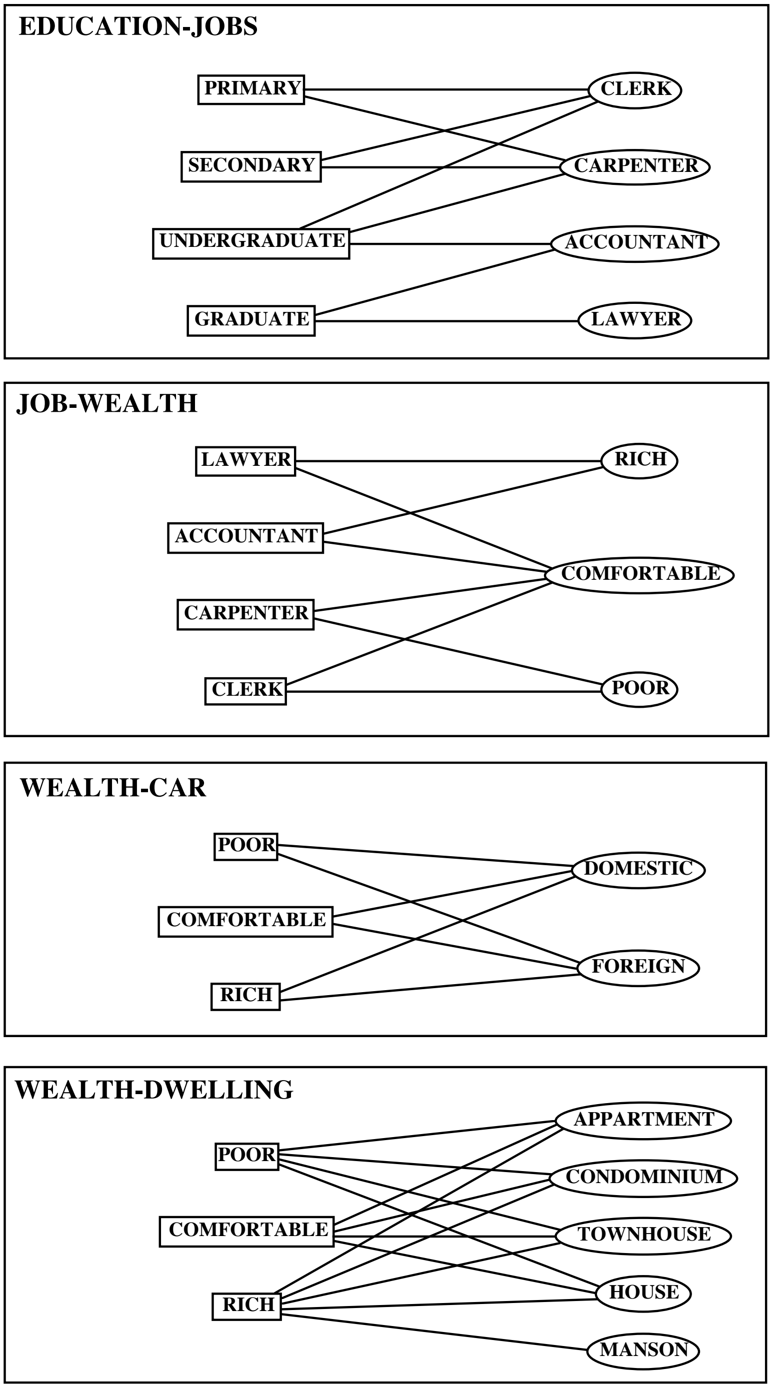


Figure 3.3: Compatibility relations from the Markov gallery

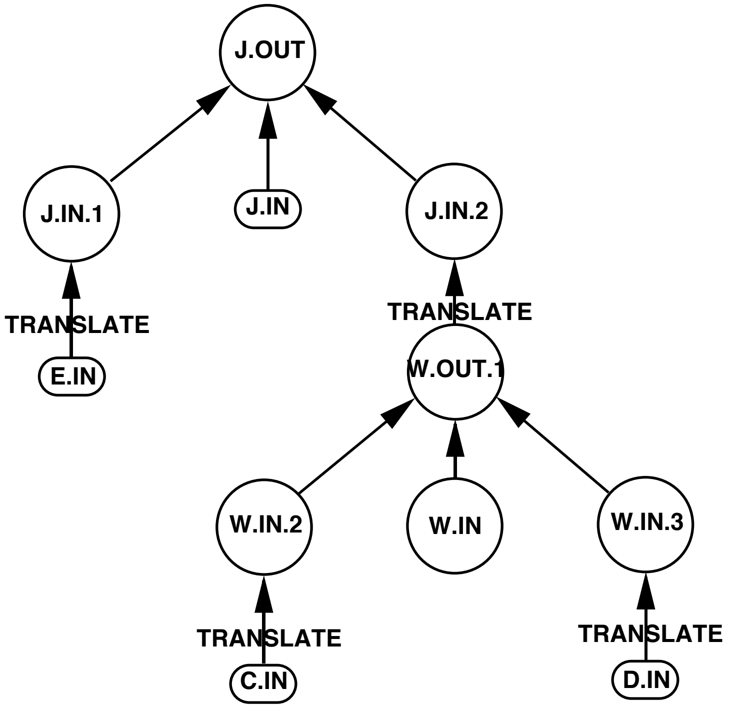


Figure 3.4: Analysis for consensus at a single frame in a Markov tree

W.IN.2 and W.OUT.2, W.IN.3 and W.OUT.3). The output node for the associated frame (W.OUT) fuses all of the inputs from neighboring frames; the output node for each neighboring frame (e.g., W.OUT.1) fuses all the input nodes except the one from that neighboring frame; the input nodes from neighboring frames (e.g., W.IN.1) are the translations of their respective outputs. The compound node in Figure 3.5 has three *ports*, or places to connect three neighboring frames. The omnidirectional analysis in Figure 3.6 has compound nodes with three or fewer ports.

3.2.3 Constructing Markov Galleries and Omnidirectional Analyses

There are two basic approaches to constructing a Markov gallery, either top-down or bottom-up. In the top-down approach, one begins by specifying a single frame of discernment that encompasses all aspect of interest. Then one looks for qualitative conditional independent partitions of this all-encompassing frame, to base the construction of a Markov tree on those partitions. This approach has been called the *partitive* approach [SSM87]. The bottom-up approach begins by identifying the variables of interest, then establishing the relationships among those variables. Markov trees are built from subsets of these variables and eventually merged to form an overall Markov tree. This approach has been termed the *multivariate* approach [SSM87]. In general, the multivariate approach is of more practical value when constructing Markov trees, while the strength of the partitive approach is in mathematical proof and analysis, because of its abstract simplicity.

The practical aspects of the multivariate approach derive from the incremental manner

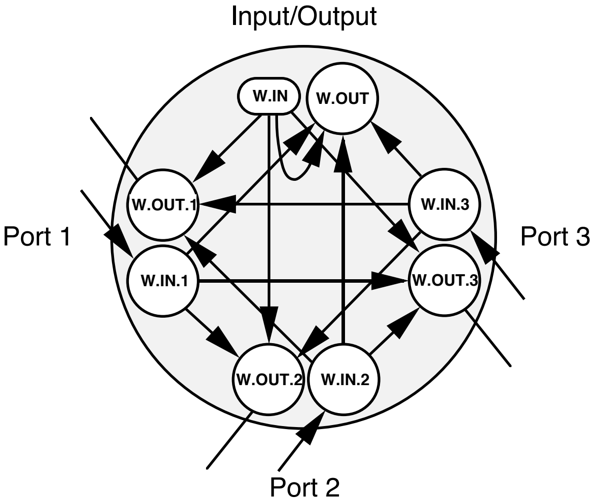


Figure 3.5: Omnidirectional compound node with three ports

in which it can be applied. Clusters of variables that are conditionally independent of the others, with respect to a selected one, are isolated. Markov trees are independently constructed for each such cluster, using the same technique, then assembled to form larger Markov trees. When a cluster of variables can no longer be subdivided along conditionally independent lines, the variables are merged into a single, more complex, variable.

This approach is consistent with the principles of modular design that help to motivate multiple frame reasoning. The developer identifies conditionally independent subproblems, solves them, and assembles the solutions. This strategy can be based on any of a number of methods for forming the conditionally independent clusters. Pearl has suggested that these clusters be discovered by identifying the causal relationships among the variables [Pea88]; a general solution for collapsing a prespecified hypergraph of interrelated variables into a Markov tree is described in [Kon86].

Once a Markov gallery has been constructed and a Markov tree identified, an omnidirectional analysis can be automatically generated. However, for any given application, a fully omnidirectional analysis may not be required. Some frames may serve as evidence for a conclusion in another frame, but be of no further interest; in this case, there is no reason to propagate beliefs back to that starting frame. The unnecessary structures are simply excluded from the analysis, thereby reducing the required computation.

3.3 Evidential Compatibilities in Omnidirectional Analyses

One can draw an analogy between the compound nodes in an omnidirectional analysis and the nodes of an inference network. Each node in an inference network can be viewed as a variable with a range of possible values. Once the value of a variable is specified, the values of neighboring variables can be determined; in turn, the values of all related dependent variables can be determined. The compound nodes in omnidirectional analyses

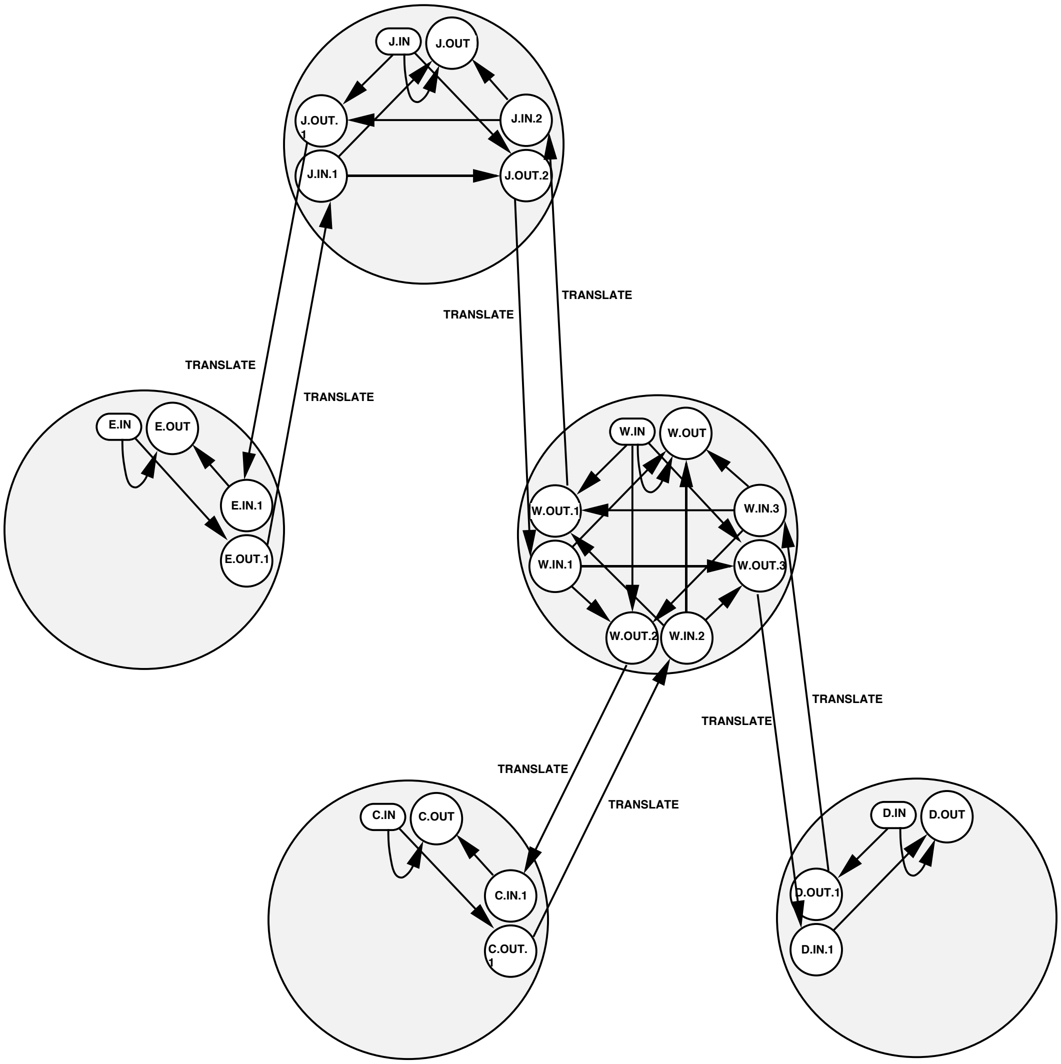


Figure 3.6: Omnidirectional analysis

similarly calculate the values of variables. Like those of many inference networks, the values of these variables can be logically or probabilistically specified. Unlike those of many inference networks, the relationships among these compound nodes are logically, rather than probabilistically, specified.

To obtain the effect of probabilistic connections among compound nodes in an omnidirectional analysis, we can transform the omnidirectional analysis and its underlying Markov gallery. After this transformation, we introduce distributions that encodes these probabilistic dependencies among the variables.

3.3.1 Prior Probabilities in Evidential Reasoning

"Prior information" generally refers to a state of knowledge before new information is integrated. As such, it can be considered as simply another source of information. Unlike Bayesian conditioning, evidential reasoning places no special status on prior information; in fact, prior information is not required. When prior information is available, it is represented as a mass distribution, then combined with the other available evidence using Dempster's rule. As we have discussed, Dempster's rule is commutative and associative, allowing evidence to be combined in any order. Thus, the prior information need not precede, or be distinguished from, the other information. In addition, the prior distribution itself may be the product of applying Dempster's rule to several component distributions.

$$m_A^{prior}(A_j) = m_A^{prior_1} \oplus m_A^{prior_2} \oplus \ldots \oplus m_A^{prior_m}(A_j) \quad .$$

If the prior distribution m_A^{prior} is an additive probability distribution, i.e., all focal elements are single element sets, then the corresponding support and plausibility for any proposition A_j are equal to the marginal probability $P(A_j)$:

$$m_A^{prior}(A_j) = \begin{cases} P(a_i), & A_j = \{a_i\} \\ 0, & \text{otherwise} \end{cases}$$

$$P(A_j) = Spt(A_j) = Pls(A_j) = \sum_{a_i \in A_j} m_A^{prior}(\{a_i\}) \quad .$$

If additional evidence consists of a mass distribution $m_A^{A_1}$ that attributes all of its mass to a single proposition A_1,

$$m_A^{A_1}(A_j) = \begin{cases} 1, & A_j = A_1 \\ 0, & \text{otherwise} \end{cases} \quad ,$$

then the application of Dempster's rule results in mass assignments equal to the conditional probabilities given A_1.

$$m_A^{prior} \oplus m_A^{A_1}(A_j) = \begin{cases} P(a_i|A_1), & A_j = \{a_i\} \\ 0, & \text{otherwise} \end{cases} .$$

If additional distributions are combined such that each attribute all of their mass to one proposition, $A_2, A_3, \ldots, A_n$, the resulting assignments are the conditionals, given the conjunction of propositions $A_1, A_2, \ldots A_n$.

$$m^{prior} \oplus m_A^{A_1} \oplus m_A^{A_2} \oplus \ldots \oplus m_A^{A_n}(\{A_j\}) = \begin{cases} P(a_i|A_1, A_2, \ldots, A_n), & A_j = \{a_i\} \\ 0, & \text{otherwise} \end{cases} .$$

The corresponding evidential interval for any proposition A_j has support and plausibility equal to $P(A_j \mid A_1, A_2, \ldots, A_n)$.

$$P(A_j|A_1, A_2, \ldots, A_n) = Spt(A_j) = Pls(A_j) = \sum_{A_i \subseteq A_j} m_A^{prior} \oplus m_A^{A_1} \oplus m_A^{A_2} \oplus \ldots \oplus m_A^{A_n}(A_i) \quad .$$

3.3.2 Transforming Omnidirectional Analyses

Given two frames, Θ_A and Θ_B, and a compatibility relation, $\Pi_{(A,B)}$, propositional statements can be translated between these two frames. As we have previously discussed, rather than translating propositional statements between these two frames via compatibility relation $\Pi_{(A,B)}$, we might choose to translate these statements to a common frame, $\Theta_{(A,B)}$, that captures all the information. This common frame is identical to the compatibility relation $\Pi_{(A,B)}$. Frames Θ_A and Θ_B are related to frame $\Theta_{(A,B)}$ via compatibility relations $\Pi_{(A,(A,B))}$ and $\Pi_{(B,(A,B))}$:

$$\begin{aligned} \Theta_{(A,B)} &= \Pi_{(A,B)} \subseteq \Theta_A \times \Theta_B \\ \Pi_{(A,(A,B))} &= \{\, (a_i, (a_i, b_j)) \mid (a_i, b_j) \in \Pi_{(A,B)} \,\} \\ \Pi_{(B,(A,B))} &= \{\, (b_j, (a_i, b_j)) \mid (a_i, b_j) \in \Pi_{(A,B)} \,\} \quad . \end{aligned}$$

Note that if the original two frames and compatibility relation were part of a Markov gallery, then the expanded gallery, with frame $\Theta_{(A,B)}$ and compatibility relations $\Pi_{(A,(A,B))}$ and $\Pi_{(B,(A,B))}$ substituted for the original compatibility relation $\Pi_{(A,B)}$, is also a Markov gallery. It defines exactly the same relationship among the original frames.

Figure 3.7 is an expanded version of the previously presented Markov gallery. Each of the original compatibility relations has been replaced by a frame and two compatibility relations, as just described. Figures 3.8 and 3.9 show the frame and compatibility relations that connect the EDUCATION frame to the JOB frame.

Given an omnidirectional analysis over the original gallery, it too can be transformed. The analysis is simply expanded to include an additional compound node between the nodes

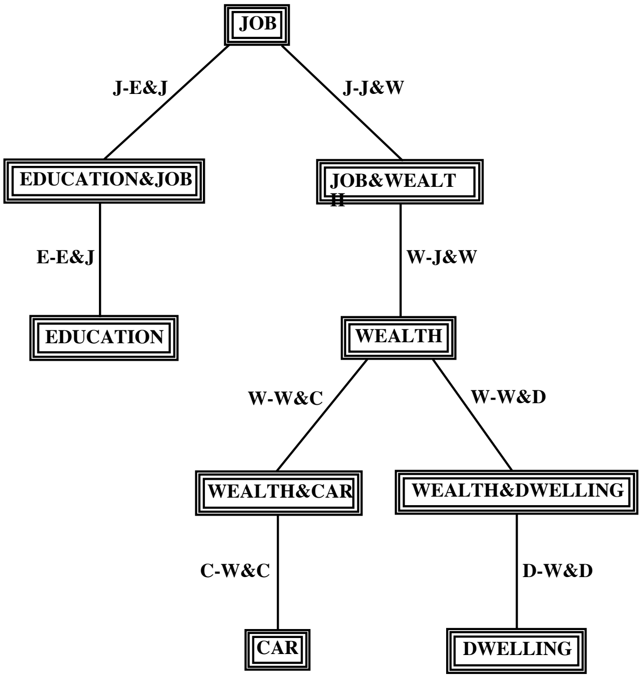


Figure 3.7: Expanded Markov gallery

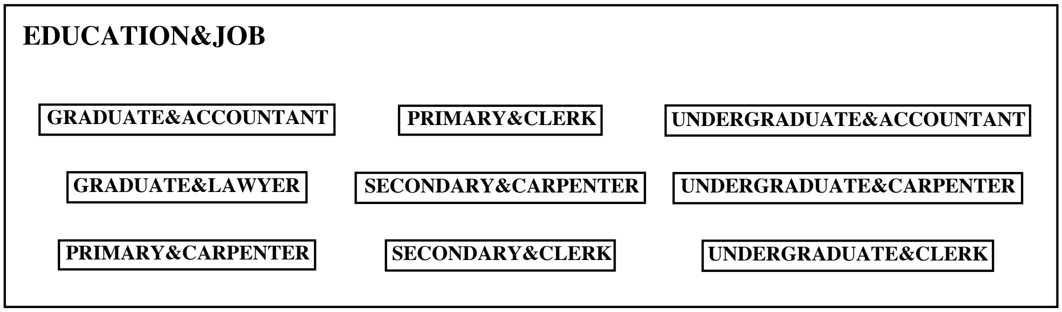


Figure 3.8: Frame from the expanded Markov gallery

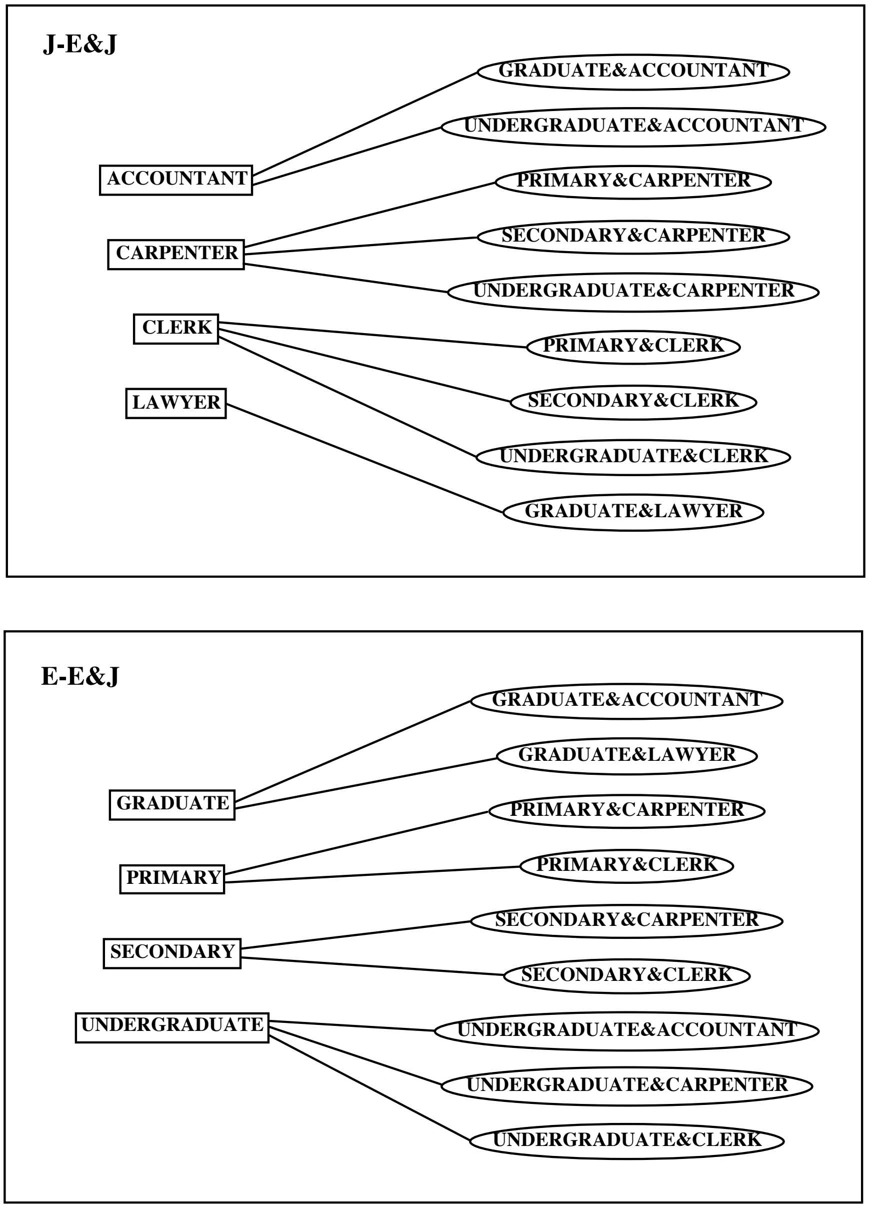


Figure 3.9: Compatibility relations from the expanded Markov gallery

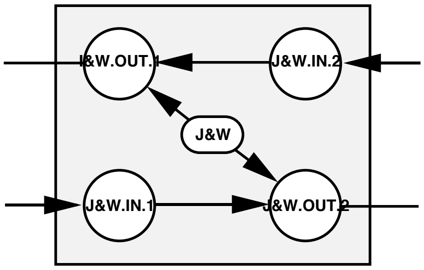


Figure 3.10: Compatibility compound node

corresponding to frames Θ_A and Θ_B; this new compound node corresponds to frame $\Theta_{(A,B)}$. Once modified, this analysis can take probabilistic information into account about the relationship between frames Θ_A and Θ_B. This information, expressed as a mass distribution over $\Theta_{(A,B)}$, provides a means of weighting translations to favor some elements of $\Theta_{(A,B)}$ over others.

If analogous substitutions are made for all compatibility relations in a Markov gallery and its omnidirectional analyses, all compatibility relations can be so weighted. Each of these *compatibility distributions* over the newly inserted *joint frames* can be viewed as a component of an overall prior distribution. Thus, the prior distribution is the result of applying Dempster's rule to these compatibility distributions. Note that these compatibility distributions are formally no different than any other distribution in an analysis. Since an omnidirectional analysis over a Markov gallery is guaranteed to produce the same results as the corresponding collapsed single-frame analysis, it follows that the omnidirectional analysis is equivalent to a single-frame analysis that includes a prior distribution composed of these compatibility distributions. If this prior distribution is an additive probability distribution, then, before any additional evidence is considered, all evidential intervals are equivalent to the marginal probabilities; if all additional bodies of evidence assert the truth of a proposition in their respective frames, then the combined probabilistic result for any proposition is its conditional probability, given these asserted propositions.

The compound nodes corresponding to these compatibility distributions have two ports; in general, we are not interested in output nodes for these frames since these frames are artifacts of the weighting technique. Therefore, we choose to represent these compatibility compound nodes in a slightly different fashion: with exactly two ports, no output node, and rectangular shape (Figures 3.10 and 3.11). When a Markov gallery and omnidirectional analysis are being transformed to include compatibility distributions, it is up to the designer to determine which relations are to be evidential.

3.3.3 Translation as Evidential Compatibility

Based upon this framework, it is easy to see how translation is a specialization of other evidential reasoning operations. Compatibility relations are Boolean valued constraints on the overall frame. Given a frame $\Theta_{(1,2,\ldots,n)}$ where

$$\Theta_{(1,2,\ldots n)} \subseteq \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n \quad ,$$

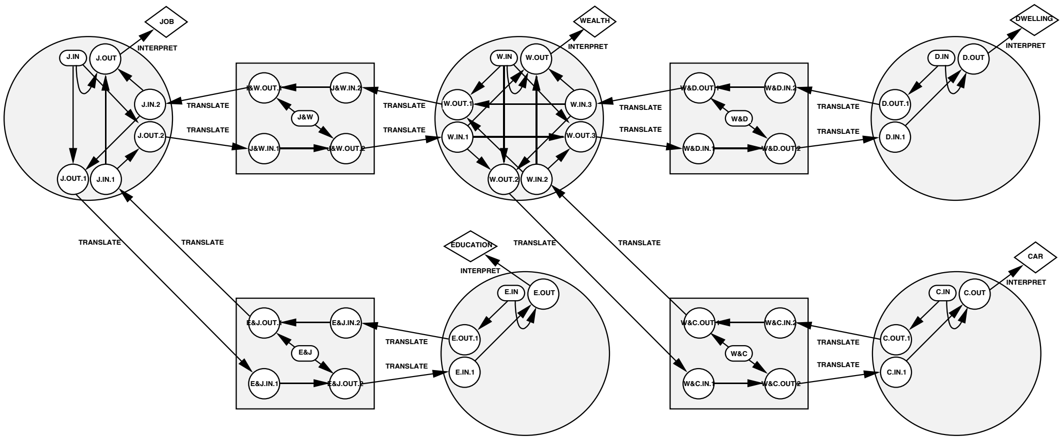


Figure 3.11: Omnidirectional analysis with compatibility distributions

a corresponding Markov tree might be defined. A joint frame $\Theta_{(i,j)}$ can be substituted for each compatibility relation, $\Pi_{(i,j)}$, in that tree. In this case, instead of defining the joint frame to be equivalent to the compatibility relation $\Pi_{(i,j)}$, we define it to be the full cross product of the two frames connected by the compatibility relation,

$$\Pi_{(i,j)} \subseteq \Theta_{(i,j)} = \Theta_i \times \Theta_j \quad .$$

Using the following mass distribution over each joint frame, $\Theta_{(i,j)}$,

$$m_{(i,j)}(X_k) = \begin{cases} 1, & X_k = \Pi_{(i,j)} \\ 0, & \text{otherwise} \end{cases} ,$$

the modified omnidirectional analysis computes exactly the same results as the original one. In turn, the modified analysis computes exactly the same results as a single-frame analysis over the full cross-product frame, $\Theta_1 \times \Theta_2 \times \ldots \times \Theta_n$. Thus the effect of translation via compatibility relations is equivalent to the introduction of a Boolean valued prior distribution, over the cross-product frame, that captures the constraints represented by the compatibility relations; these are the same constraints captured by $\Theta_{(1,2,\ldots,n)} \subseteq \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n$. Further, the conflict generated during this combination is the same conflict generated during translation.

3.3.4 Discounting as Evidential Compatibility

Recall that discounting lowers the apparent information content of a mass distribution in accordance with a discount rate $r \in [0, 1]$:

$$m_i^{\%}(X_k) = \begin{cases} (1-r)\, m_i(X_k), & X_k \neq \Theta_i \\ r + (1-r)\, m_i(\Theta_i), & \text{otherwise} \end{cases} .$$

Within a Markov gallery, consider introducing an additional frame $\Theta_i^{\%}$ for every frame Θ_i, $\Theta_i^{\%} = \Theta_i$. Then introduce a cross-product frame $\Theta_{(i,i)}$ between each such pair of frames.

$$\Theta_{(i,i)} = \Theta_i \times \Theta_i$$

If the following mass distribution is used as the compatibility distribution over $\Theta_{(i,i)}$,

$$m_{(i,i)}(X_k) = \begin{cases} r, & X_k = \Theta_{(i,i)} \\ 1 - r, & X_k = \{\, (\theta_j, \theta_j) \mid \theta_j \in \Theta_i \,\} \\ 0, & \text{otherwise} \quad , \end{cases}$$

where

$$r \in [0, 1] \quad ,$$

then the effect will be the same as having discounted the distribution m_i by r. In other words, if m_i is defined over $\Theta_i^{\%}$, translated to $\Theta_{(i,i)}$, combined with $m_{(i,i)}$, and translated to Θ_i, the result is the same as having discounted m_i, defined over Θ_i, by r.

Here, it is easy to see that the discount rate r is the degree to which the result is independent of the evidence i.e., the degree to which the evidence provides no information.

3.3.5 Constructing Omnidirectional Analyses

Frequently the compatibility distributions in an omnidirectional analysis are based upon subjective estimates or weak objective information. Testing and tuning is required to enhance the reliability of the analytic conclusions. To aid this process, it is useful to add a discounting node to each compatibility compound node. This discounting node is placed directly after the input node for the associated frame; see Figure 3.12. It serves to reduce the impact of the associated compatibility distribution. Together these discounting nodes can be used to balance the relative impact of the compatibility distributions. Similar discounting nodes are useful in the other compound nodes, as in Figure 3.13; they serve to balance the relative impact of the different sources of information. Collectively, these provide a useful and intuitive means of tuning an omnidirectional analysis, as in Figure 3.14.

Omnidirectional analyses can be constructed by assembling compound nodes in accordance with a Markov gallery. Higher dimensional compound nodes can be used in place of lower dimensional ones, during the construction phase, to ease the addition of new variables. Often both the gallery and analyses are constructed in parallel. New frames are added to the gallery as new compound nodes are added to the analysis. As construction progresses, qualitative conditional independence is periodically checked by examining chains of frames and compatibility relations to ensure that the chains of compatible frame elements are possible combinations in the domain of application. If all such combinations are valid and all the valid combinations are included, the gallery is a Markov gallery for the target domain. If some combinations are invalid or some valid combinations are excluded, either the gallery

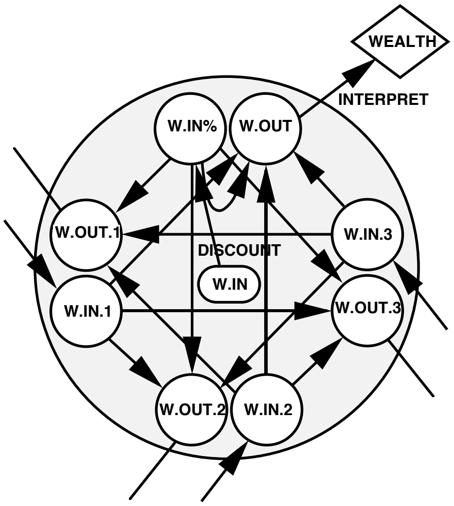


Figure 3.12: Discounted compatibility compound node

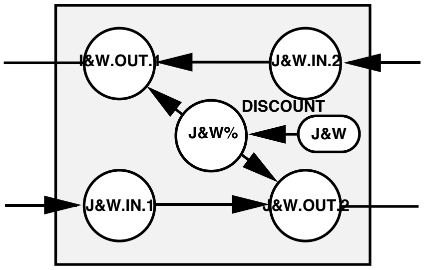


Figure 3.13: Discounted compound node

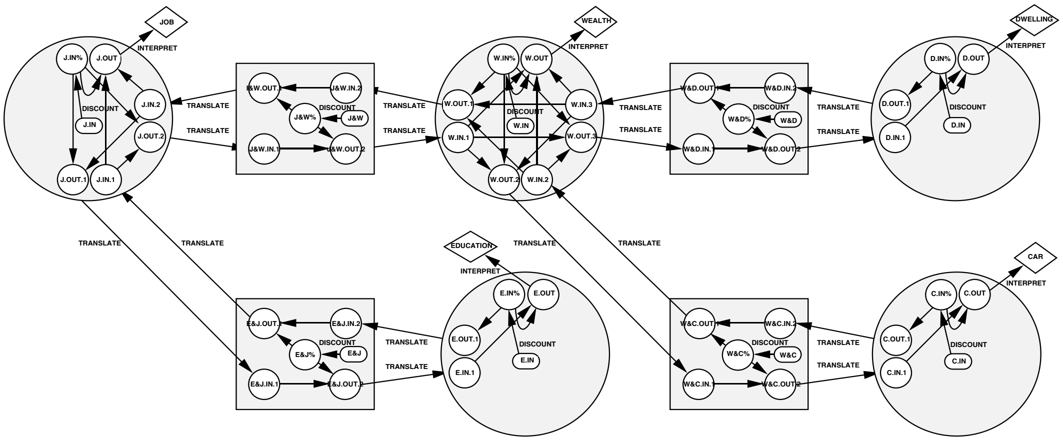


Figure 3.14: Discounted omnidirectional analysis

can be corrected, or it can be used as an approximation. As an approximation, it is as good as the implied all-encompassing frame. For example, if the true frame is

$$\Theta_{(1,2,\ldots,n)} \subseteq \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n \quad ,$$

while the implied frame, assuming Π is the set of compatibility relations in the gallery, is

$$\begin{aligned}\hat{\Theta}_{(1,2,\ldots,n)} &\subseteq \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n \\ &= \{\, (\theta_1, \theta_2, \ldots \theta_n) \mid \theta_1 \in \Theta_1,\, \theta_2 \in \Theta_2, \ldots, \theta_n \in \Theta_n,\, \forall\, \Pi_{(i,j)} \in \Pi,\, (\theta_i, \theta_j) \in \Pi_{(i,j)} \,\} \quad ,\end{aligned}$$

then the approximate answers will differ from the true answers in exactly the way that reasoning based upon $\hat{\Theta}_{(1,2,\ldots,n)}$ differs from reasoning based upon $\Theta_{(1,2,\ldots,n)}$.

3.4 Conditional Probabilities in Omnidirectional Analyses

Analogous structures to omnidirectional analyses, based upon classical probability theory, have been proposed [Pea88]. These *Bayesian networks* use conditional probabilities to capture the probabilistic relationships among related variables. Given a probabilistic estimate of the value of a variable, the values of its neighboring variables are calculated based upon these conditional probabilities using Bayes' rule. The relationship between belief propagation in Bayesian networks and Markov trees has been previously reported [SS88]. Although our omnidirectional analyses cannot directly incorporate conditional probabilities, if conditional probabilities are interpreted as constraints on joint distributions, they can be incorporated.

Given two variables, represented by frames Θ_A and Θ_B, a conditional probability might be used to express a probabilistic constraint between these two variables. Each such conditional probability $P(a_i|b_j)$ is equivalent to the quotient of two prior probabilities,

$$P(a_i|b_j) = \frac{P(a_i, b_j)}{P(b_j)} \quad .$$

Assuming that the underlying prior distribution is an additive distribution and that a cross-product frame $\Theta_{(A,B)}$ has been established, then this quotient can be expressed in terms of either support or plausibility:

$$P(a_i|b_j) = \frac{Spt\Big(\{\,(a_i, b_j)\,\}\Big)}{Spt\Big(\{\,(a_k, b_j) \mid (a_k, b_j) \in \Theta_{(A,B)}\}\Big)} = \frac{Pls\Big(\{\,(a_i, b_j)\,\}\Big)}{Pls\Big(\{\,(a_k, b_j) \mid (a_k, b_j) \in \Theta_{(A,B)}\}\Big)} \quad .$$

Alternatively, this quotient can be expressed directly in terms of the original two frames and compatibility mappings that led to the cross-product frame.

$$P(a_i|b_j) = \frac{Spt\Big(\Gamma_{A\mapsto(A,B)}(\{a_i\}) \cap \Gamma_{B\mapsto(A,B)}(\{b_j\})\Big)}{Spt\Big(\Gamma_{B\mapsto(A,B)}(\{b_j\})\Big)} = \frac{Pls\Big(\Gamma_{A\mapsto(A,B)}(\{a_i\}) \cap \Gamma_{B\mapsto(A,B)}(\{b_j\})\Big)}{Pls\Big(\Gamma_{B\mapsto(A,B)}(\{b_j\})\Big)} \quad .$$

In Bayesian networks, conditional probabilities, $P(a_i|b_j)$ for all $a_i \in \Theta_A$, $b_j \in \Theta_B$, are given for each such pair of variables. Each of these conditional probabilities can be interpreted as a constraint on the underlying distribution. In general, there are many mass distributions that fit these constraints, including many additive distributions. The problem, then, for omnidirectional analyses, is determining which mass distribution to use.

3.4.1 Mass Distributions from Conditional Probabilities

If one has evidence represented as a mass distribution over $\Theta_{(A,B)}$, then acquires new evidence that conclusively establishes the truth to lie within $B_j \subset \Theta_{(A,B)}$, one should adopt the conditional beliefs that result from using Dempster's rule to combine the original mass distribution, $m_{(A,B)}$, with another that attributes all of its mass to B_j, $m^{B_j}_{(A,B)}$. On the other hand, if one is first given this conditional mass distribution, $m_{(A,B)|B_j}$, then told that the truth might lie outside of B_j, one would find, in general, that there are many mass distributions that obey these conditionals. If a single mass distribution is to be selected as a representation of what is known, then the choice depends upon additional information about the circumstances that gave rise to the conditionals.

One approach, called *conditional embedding* [Sme78], requires that the evidence on which $m_{(A,B)|B_j}$ is based does not impugn any of the possibilities outside of B_j i.e., $\Theta_{(A,B)} - B_j$. That is, $m_{(A,B)|B_j}$ is assumed to be correct if B_j happens to be true and is irrelevant if the truth lies outside of B_j. This suggests that mass attributed by $m_{(A,B)|B_j}$ to A_i, where $A_i \subset B_j$, instead be attributed to $A_i \cup (\Theta_{(A,B)} - B_j)$:

$$m_{A,B}(X_k) = \begin{cases} m_{(A,B)|B_j}(A_i), & X_k = A_i \cup (\Theta_{A,B} - B_j) \\ 0, & \text{otherwise} \end{cases} ,$$

where $A_i \subset B_j$. As a consequence of this definition,

$$m_{(A,B)} \oplus m^{B_j}_{(A,B)}(A_i) = m_{(A,B)|B_j}(A_i) = P(A_i|B_j) \quad ,$$

where $m^{B_j}_{(A,B)}(B_j) = 1$. Further, of all the mass distributions that conform to the given conditionals, the evidential intervals implied by this distribution are the weakest; their supports are less than or equal to and their plausibilities are greater than or equal to those implied by the other distributions [Sme78].

Given partitions $A_1, \ldots, A_i, \ldots, A_m$ over Θ_A and $B_1, \ldots, B_j, \ldots, B_n$ over Θ_B, if a different observation gave rise to each set of conditionals $P(A_i|B_j)$ for each B_j, then each set can be represented by a mass distribution over $\Theta_{(A,B)}$ through conditional embedding. If these observations were independent of one another, then these mass distributions can be combined using Dempster's rule. The result is a mass distribution that conforms to all of the conditional probabilities and is vacuous at the margin B_j [Sme78]. Since it is not uncommon for the same phenomenon to have been studied under a number of distinct conditions B_j and for each of these studies to have been independently conducted, this method has practical utility. For example, using this method, a medical study conducted to determine the likelihoods of various symptoms for a given pathology can be combined with similar independent studies for other pathologies to predict a patient's pathology from his symptoms.

Further, if the chosen partitions correspond to the elements of frames Θ_A and Θ_B and the resulting distribution is combined via Dempster's rule with an additive distribution over Θ_B, then the results will be the same as those obtained by Bayesian nets, given the same data. Thus, conditional omnidirectional analyses reduce to Bayesian nets when the conditional probabilities $P(a_i|b_j)$ are embedded in cross product frames $\Theta_{(A,B)}$ and the other inputs are additive probability distributions over the marginal frames (see Appendix A).

3.4.2 Mass Distributions Versus Conditional Probabilities

Uncertain reasoning theories based on conditionals are not easily related to theories based on logic [DP90]. The root of the problem is that conditionals are not part of logic. Since mass distributions are by definition distributions over logical statements, conditionals cannot be directly represented by mass distributions.

One means of incorporating conditionals into evidential reasoning is to interpret conditionals as constraints on mass distributions. However, as we have already discussed, there is no unique means of so doing. The essence of the conditional embedding method is to interpret conditional probabilities as probabilities of material implication, that is $P(a_i|b_j)$ is interpreted as $P(b_j \Rightarrow a_i)$ which, in turn, is interpreted as $P(\neg b_j \vee a_i)$. Thus, conditional probabilities are converted into probabilities over logical statements. Dempster's combination of the resulting distributions conforms to the conditionals when further combined with additive distributions. However, other calculations will not necessarily conform. For example, if the evidential intervals for the margins a_i are calculated from the combination of the conditional distributions, the bounds are generally wider than those implied by taking the conditionals as simultaneous constraints over a single underlying distribution. The primary problem is that $a_i|b_j$ is not generally part of logic; attempts to include it have made it logically dissimilar to $\neg b_j \vee a_i$. Material implication is defined for all truth values of a_i and b_j, being false when b_j is true and a_i is false, and true for all other combinations; when b_j is true, the conditional has the same truth value as a_i, but is undefined when b_j is false [DP90]. Therefore, it is not surprising that conditional embedding is sometimes at variance with conditional reasoning.

In addition, the use of Dempster's rule requires that the distributions represent independent bodies of evidence. If the conditional probabilities do not derive from independent observations, then a different approach is required. For example, Shafer [Sha82] discusses the case where all of the conditional probabilities arise from a single frequency distribution; other cases have been discussed in the literature [Dem68, Sha82, Sha76b, Sme78].

From a belief function perspective, the problem is that conditional probabilities are only a partial account of an item of evidence, and mass distributions are not meant to be constructed from such elements. In belief functions, we expect to represent each independent item of evidence directly by a mass distribution [Sha82]. If conditional probabilities are going to be fully incorporated, then the theory of belief functions, and consequently evidential reasoning, must be extended to include these new objects. This approach is already under investigation [Spi90].

However, this problem only arises when we insist on capturing evidence in terms of conditional probabilities. If we use mass distributions over logical statements to capture evidence, the problem does not exist. Thus, if we deliberately seek probabilities of material implication, or other logical statements, rather than conditional probabilities, the problem is avoided. Of course, this may not always be practical, but based upon our experience, we believe mass distributions to be well suited to the representation and application of many forms of expert knowledge.

Chapter 4

Gister-CL's User Interface

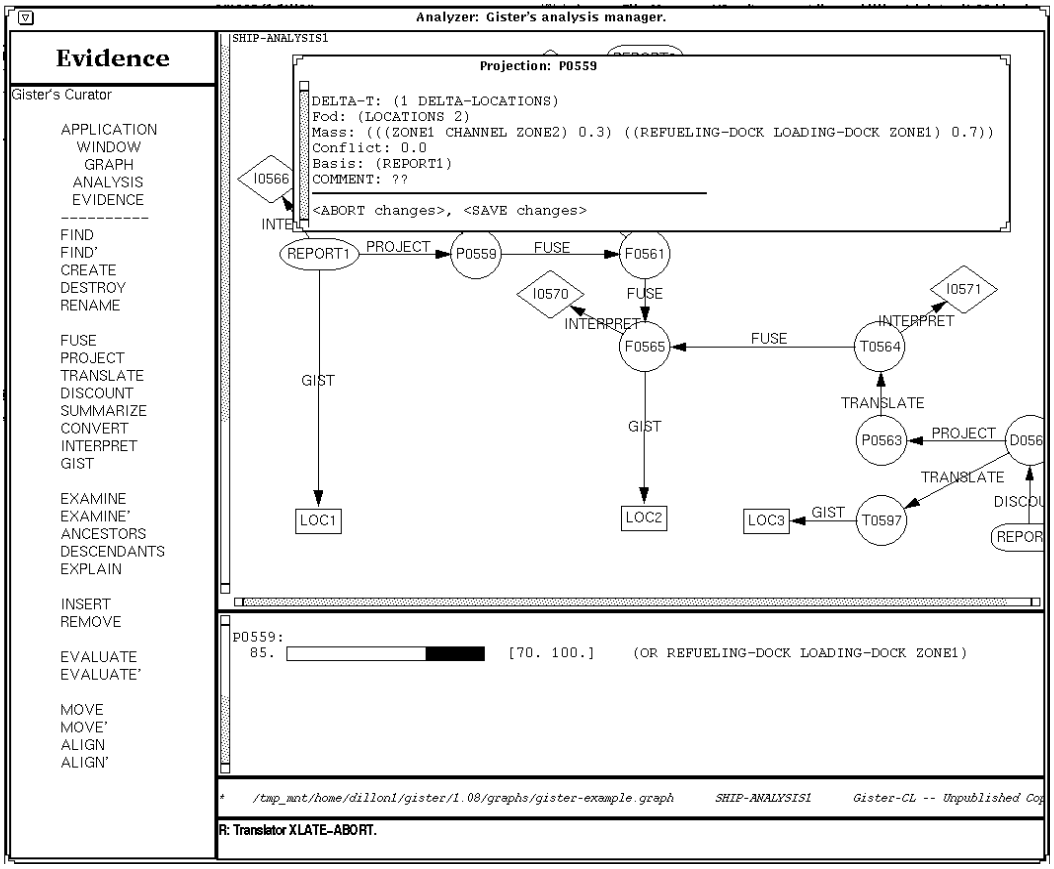
To support the construction, modification, and interrogation of evidential structures, we have developed Gister-CL. Gister-CL provides an interactive, menu-driven, graphical interface for the user to manipulate analyses, frames, compatibility relations, and galleries. The user simply makes menu selections to add an evidential operation to an analysis, to modify operation parameters, or to change any portion of a gallery.

4.1 Basic User Interface Interactions

Gister-CL has two main subsystems, the *Analyzer* and the *Curator*. The Analyzer supports the construction, modification, and inspection of evidential analyses. These analyses consist of interrelated primitive and derivative bodies of evidence concerning the probable state of the world. The Curator provides similar functionality for the background knowledge that logically delimits the possible states of the world. This background knowledge is represented by galleries of frames and compatibility relations.

The user interfaces to both of these subsystems, and Grasper-CL, share the same basic structure. The Gister-CL display window is divided into several regions or panes as shown in Figure 4.1. Each pane displays specific types of information during Gister-CL's operation. Although the layout of panes within the window is normally fixed, it can be altered to a degree; see Section 4.3.1. The largest region is the graph pane, where analyses, frames, compatibility relations, and galleries are displayed as Grasper-CL graphs. This pane is scrollable in case the current graph is too large to be displayed in its entirety. To the left of the graph pane is the command pane, which displays a menu of Gister-CL modes, plus all the commands within the currently selected mode. Above the command pane is the mode pane, which displays the name of the current editing mode. Below the graph pane is a interactor pane, where COMMON LISP expressions can be typed in by the user. In response, these expressions are evaluated, and their results are printed. This facility provides interactive access to Gister-CL's programmer interface, discussed in the following chapter, and COMMON LISP in general.

The primary means of interaction is through the two command menus on the left side of the screen. When a system takes control, it replaces the contents of these menus. The



```
DELTA-T: (1 DELTA-LOCATIONS)
Fod: (LOCATIONS 2)
Mass: (((ZONE1 CHANNEL ZONE2) 0.3) ((REFUELING-DOCK LOADING-DOCK ZONE1) 0.7))
Conflict: 0.0
Basis: (REPORT1)
COMMENT: ??
<ABORT changes>, <SAVE changes>
```

```
P0559:
  85.                  [70. 100.]    (OR REFUELING-DOCK LOADING-DOCK ZONE1)
```

```
*   /tmp_mnt/home/dillon1/gister/1.08/graphs/gister-example.graph     SHIP-ANALYSIS1     Gister-CL -- Unpublished Co
```

```
R: Translator XLATE-ABORT.
```

Figure 4.1: Gister-CL's graphical interface

upper menu is the *noun menu.* The user determines with what class of objects he wishes to work and selects the appropriate noun from the menu. Once a noun has been selected, the associated menu of verbs appears in the lower menu. A selection from this *verb menu* invokes the operation corresponding to the selected verb on the previously selected noun. The user then designates the appropriate nodes, edges, and the like for the selected operation.

A number of verb commands have two forms; the default or standard form and an alternative or expanded form. In this manual, the names of the expanded and standard forms differ only by the presence of absence of a trailing prime (') e.g., SELECT for the standard form and SELECT' for the expanded. Some implementations will simply include both forms in the verb menus; others use a different mouse gesture on the standard form that appears in the menu to trigger the expanded form. Novice users should confine themselves to the standard verb forms until they become more experienced.

Mouse gestures adhere to a number of conventions, to insure uniform interaction throughout the system. These conventions are consistent with Grasper-CL. Moving the mouse-driven cursor over any menu item will cause the item's documentation to appear in the mouse documentation line at the bottom of the window. This is the easiest way to find out what a particular command will do. Any menu item that is visible on the screen may be legally selected.

Since different types of machines have mice with differing numbers of buttons, Gister-CL's mouse interactions are not uniform across machines. Unless otherwise stated, all mouse interactions utilize the left mouse button. For example, selecting a menu item from a command menu is accomplished by positioning the mouse cursor over a menu item and clicking the left button on mice with multiple buttons or clicking the (only) button on single button mice.

Throughout Gister-CL, the hierarchical nature of these menus is observed. The noun menu is always at the root of the hierarchy, followed by one of the verb menus, and finally a (possible) temporary menu at the deepest level. On multiple button mice, the user can pop up a level in this hierarchy at any time by clicking the right mouse button, thereby terminating the ongoing operation. Use of the right button is encouraged as it can eliminate a lot of excessive mouse movement.

Verb commands prompt the user for a variety of information, which the user can supply through type-in to the interactor pane, by clicking on objects in the graph pane, and by clicking and type-in to pop-up menus that the verb commands create. For example, the EVIDENCE TRANSLATE command asks the user to click on an evidence node to be translated, then asks the user to click on a location for the new translated evidence node, and then pops up a menu that allows the user to select the target frame for the translation.

Other pop-up menus have a more complex format that displays a number of program parameters, which the user can alter. Some parameters have a small fixed number of possible values, with the current value displayed in bold face. The user can click on a different alternative value to select it. Other parameters can take on a value that the user types in by first either left clicking on the current value to replace it or middle clicking to edit it. Type-in *must* be terminated by the Return key, and Emacs-like control characters provide editing capabilities during type-in. When all parameters have the desired values, the user has several options at the bottom of the menu for exiting. Typically there is one option that save the changes and another that discards them.

The second mechanism for manipulating graph structures is invoked by pointing directly at graph entities. Nodes and edge labels can be repositioned by pressing and holding the left mouse button and dragging them to new positions. While under the control of one of the two Gister-CL subsystems, the standard menu hierarchy can be temporarily suspended to allow quick access to single Grasper-CL operations. To do this requires a multiple button mouse; the user clicks the right button on a space label, node, or edge, and a temporary menu of Grasper-CL verbs for the selected object appears. As always, the documentation for these operations appears in the mouse documentation line. When a verb is selected, the corresponding Grasper-CL operation is performed and, when completed, the Gister-CL interface will return to the same point in the hierarchy as when the right click occured.

Great care must be taken when manipulating Gister-CL entities through Grasper-CL operations. Although many of Gister-CL's data structures are accessible through Grasper-CL, it is up to the user to maintain the consistency of Gister-CL's data structures when manipulating them through Grasper-CL. Manipulation of these data structures through Gister-CL's interface is preferable, since it helps to maintain consistency. Only experienced users should use Grasper-CL operations to manipulate Gister-CL data structures.

Many Gister-CL analyses, frames, compatibility relations, and galleries are too large to fit in the graph window; therefore, the graph pane is a scrollable viewport onto the entire graph drawing. The graph pane can be scrolled left or right, up or down, by clicking the mouse in the scroll bars that border the graph pane. The birds-eye view window (or simply, the bird's eye) provides a low-resolution view of the entire graph that is scaled to fit completely in one window. The bird's eye is useful for visualizing and navigating through a large, complex graph. The bird's eye contains node icons, or edge shafts, or both (depending on the Grasper-CL shape parameter %Space-Birds-Eye-Contents) — different graphics will look best for different graphs. Node labels, edge labels, and edge arrowheads are never displayed in the bird's eye.

The user can manipulate the birds-eye window when in birds-eye mode, which is entered by middle clicking on the background of the graph pane. In birds-eye mode, the user can select a command from a menu of the following items:

- **Scroll Graph** — A rectangle within the bird's eye shows the position of the graph pane viewport within the entire graph. The user can reposition this rectangle to a new position in order to scroll the graph pane. After selecting this menu item, left click in the background of the birds-eye window and drag the rectangle to a new position.
- **Return To Graph** — This command exits birds-eye mode and returns to the top level.

The other birds-eye mode commands are still under development and are not yet documented.

Gister-CL, like all other applications built on Grasper-CL, has an APPLICATION command in its noun menu. Selecting this command caused a menu of applications to appear.

When running Gister-CL this menu will include at least the following entries: one for Gister's Analyzer, one for Gister's Curator, one for Grasper-CL, one to Exit from the graphic interface, and one to Kill the Lisp process. Selecting one of the applications, other than Exit or Kill, causes the corresponding system or subsystem to take control, replacing the noun menu with the commands appropriate to that application.

4.2 Invoking Gister-CL's Graphical Interface

You invoke the Gister-CL graphical interface by typing a form to a COMMON LISP listener, whose image already includes Gister-CL's system definition and Grasper-CL or its system definition. For information on installing Gister-CL and (accompanying) Grasper-CL, talk to your system administrator or see the *Gister-CL Installation Guide.* Type the following form to initialize Gister-CL in its Analyzer subsystem,

```
(cl-user:run-analyzer)     ;; or (er:run-analyzer)
```

or

```
(cl-user:run-curator)      ;; or (er:run-curator)
```

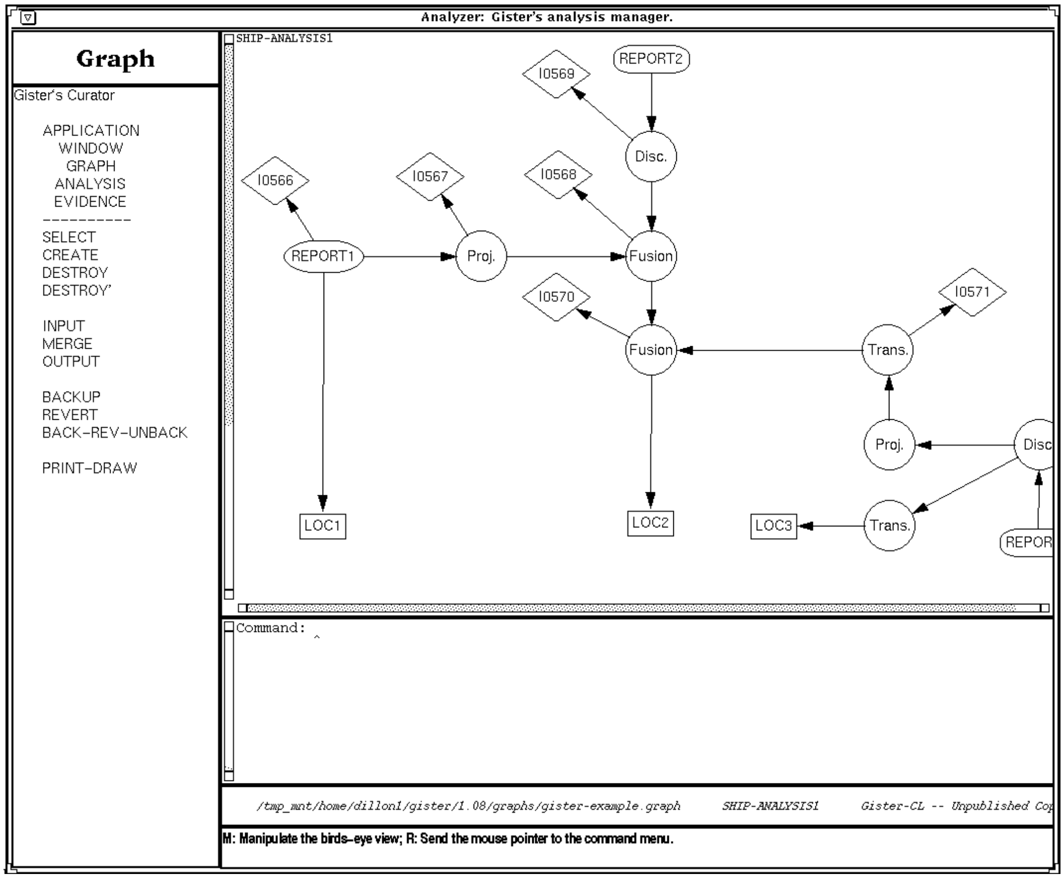
to initialize it in its Curator subsystem. Alternatively, type

```
(cl-user:run-gister)       ;; or (er:run-gister)
                           ;; or (sri:run-system 'gister)
```

to initialize Gister-CL in the subsystem appropriate to the currently selected Grasper-CL space; if that space represents an Analysis, the Analyzer is initialized, otherwise the Curator is initialized. Each of these initialization procedures have keyword arguments for the :height and :width of the window and for the position of the :top and :left extremities of the window on the screen.

4.3 The Analyzer

When the Analyzer is selected from the application menu, the Analyzer noun menu appears. The noun options are APPLICATION, WINDOW, GRAPH, ANALYSIS, and EVIDENCE (Figure 4.2). When these are selected, their associated verb menus are exposed as shown in Figure 4.3. However, if the space in the display window is not an analysis when ANALYSIS or EVIDENCE is selected, the Analyzer will request the user to either select or create an analysis, through a temporary menu, before proceeding. In addition, at the top of the noun menu appears an entry for quickly switching to Gister's Curator; selecting this entry is exactly equivalent to selecting the same entry from the application menu.



```
Command:
```

/tmp_mnt/home/dillon1/gister/1.08/graphs/gister-example.graph   SHIP-ANALYSIS1   Gister-CL -- Unpublished Cop

M: Manipulate the birds-eye view; R: Send the mouse pointer to the command menu.

Figure 4.2: Gister-CL's analyzer display

Window

Gister's Curator

APPLICATION
WINDOW
GRAPH
ANALYSIS
EVIDENCE

RESIZE
PANE-LAYOUT
BIRDS-EYE

Graph

Gister's Curator

APPLICATION
WINDOW
GRAPH
ANALYSIS
EVIDENCE

SELECT
CREATE
DESTROY
DESTORY'

INPUT
MERGE
OUTPUT

BACKUP
REVERT
BACK-REV-UNBACK

PRINT-DRAW

MAGNIFY

Analysis

Gister's Curator

APPLICATION
WINDOW
GRAPH
ANALYSIS
EVIDENCE

SELECT
SELECT'
CREATE
CREATE'
DESTROY
DESTROY'

RENAME
RENAME'
COPY
COPY'

BACKUP
BACKUP'
REVERT
REVERT'

EXAMINE
EXAMINE'

PRINT-DRAW
PRINT-DRAW'
REDRAW
RESCALE

EVALUATE
EVALUATE'
COMPILE
COMPILE'

Evidence

Gister's Curator

APPLICATION
WINDOW
GRAPH
ANALYSIS
EVIDENCE

FIND
FIND'
CREATE
DESTROY
RENAME

FUSE
PROJECT
TRANSLATE
DISCOUNT
SUMMARIZE
CONVERT
INTERPRET
GIST

EXAMINE
EXAMINE'
ANCESTORS
DESCENDANTS
EXPLAIN

INSERT
REMOVE

EVALUATE
EVALUATE'

MOVE
MOVE'
ALIGN
ALIGN'

Figure 4.3: The Analyzer's menus

4.3.1 The Window Options

The Analyzer WINDOW options are the same as the Grasper-CL WINDOW options. They affect the appearance of the window and entities displayed with it, but do not alter the entities themselves.

By default, the graphics system does not allow you to scroll outside the boundaries of an existing drawing. This restriction is problematic when you wish to create new nodes or edges that (for example) lie to the right of the rightmost area of the graph pane that can be exposed by scrolling. RESIZE allows the user to expand the size of the graph pane drawing plane in the positive direction along both coordinate axes, that is, rightward and downward. This command pops up a temporary menu that displays the current extreme X and Y coordinates. By default these extrema are the largest X and Y coordinates present in all nodes in the current space, which also represent a lower bound on the values that the user can assign. By increasing the X extremum, the user creates new drawing space on the right edge of the graph pane where new nodes and edges can be drawn.

The PANE-LAYOUT command alters the arrangement of the panes within the window. The user chooses a new arrangement of panes by name from a pop-up menu. The default arrangement (shown in Figure 4.2) is called "Left Menu with Small Lower Interactor". The arrangement called "Left Menu with Big Lower Interactor" increases the size of the interactor pane and decreases the size of the graph pane. Other arrangements, "Top Menu with Small Lower Interactor," "Top Menu with Small Upper Interactor," and "Top Menu with Small Right Interactor," place the menu of noun commands across the top; when a noun is selected with a click of the mouse, a momentary menu of verb commands is displayed; the momentary menu disappears once a verb is selected.

The birds' eye window provides the user with a low resolution view of whatever is displayed in the graph pane. The BIRDS-EYE command allows the user to turn the birds' eye window on and off and to set various parameters affecting its appearance including scale and content.

4.3.2 The Graph Options

The GRAPH option is the second entry on the noun menu. Its associated verbs are a subset of those available through the GRAPH option in Grasper-CL. The analyses, along with the graphically represented frames, compatibility relations, and galleries, make up the entire Gister-CL knowledge base and are all maintained as part of a single graph. OUTPUTting this graph saves the state of the entire Gister-CL system under a user-selected file on disk. The user can subsequently return to this state of the system by INPUTting the appropriate file. MERGEing the current graph with a graph previously input, augments the selected graph with the information from the other[1]. Other graphs previously input, can be SELECTed or the currently selected graph DESTROYed. Graphs other than the currently selected one can be DESTROY'ed. Another option in the verb menu allows the

---

[1] If the graphs being merged share Grasper-CL spaces with common names, the spaces in the currently selected graph are replaced.

user to BACKUP all the graphical structures in the entire graph, allowing the user to later revert portions of this graph back to their current state. The REVERT option in this menu reverts the entire graph. The BACK-REV-UNBACK option allows the user to selectively backup, revert, or unbackup (i.e., destroy the backup copy) any or the currently loaded graphs. The PRINT-DRAW option allows the selected graph to be (textually) printed or (graphically) drawn to paper or to a file; these are for documentation purposes only and cannot be used to recreate the graph. OUTPUT should be used if the graph needs to be recreated at some future time; INPUT is used to recreate a graph that was previously OUTPUT. Finally, MAGNIFY allows the user to select the magnification to be used for graph display.

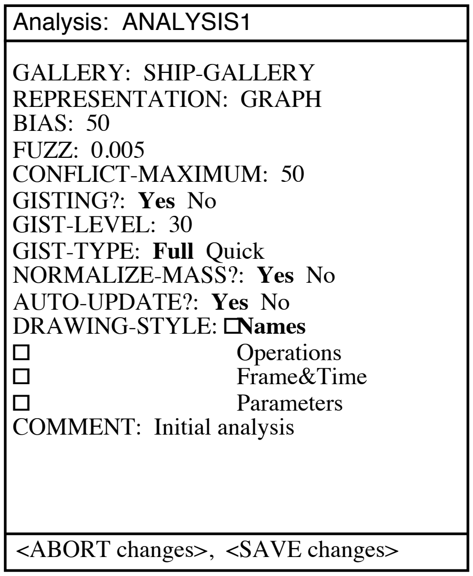
4.3.3 The Analysis Options

If the user wants to operate on the entire analysis, he selects ANALYSIS from the noun menu. In response, the analysis verb menu appears. Most of the options in this menu provide utilitarian services for maintaining multiple analyses. CREATE generates a null analysis, while COPY generates an analysis that is identical to the one currently selected. In either case, the user is asked to name the new analysis, but can choose to rename it at some future time through the RENAME option, and is given an opportunity to EXAMINE several analysis wide setting (see below). BACKUP is similar to COPY except that it copies to a hidden area, saving the current state of the selected analysis. Later, the user can restore the selected analysis to its backed-up state by selecting REVERT. DESTROY simply deletes the currently selected analysis and SELECT allows one to move to a different analysis, that references a common gallery; SELECT' allows one to move to an arbitrary analysis. All of the conclusions in an analysis are recalculated when EVALUATE is selected. REDRAW redraws the graph pane, RESCALE repositions the nodes and edges in an analysis (by additive and/or multiplicative factors), and PRINT-DRAW (textually) prints or (graphically) draws analyses on paper or to a file.

EXAMINE provides access and the ability to change several analysis wide setting. These include the name of the GALLERY that supports the analysis, the REPRESENTATION of the gallery elements to use, a BIAS for evidential decisions, a FUZZ factor used to determine what mass assignments are effectively zero, a CONFLICT-MAXIMUM that when exceeded in the course of an evidential operation will cause a warning to be issued, a GISTING? switch that controls whether gisting is automatically performed when bodies of evidence are examined, a default GISTING-LEVEL, a GIST-TYPE that determines if full gisting should be used or if the faster but less informative quick gisting should be used, an AUTO-UPDATE? switch that determines whether the analysis is to be automatically updated as modifications are made or await the use of the EVALUATE option[2], a DRAWING-STYLE that determines how the analysis is displayed, and a COMMENT string. When EXAMINE is selected, a form appears containing these settings (Figure 4.4); when EXAMINE' is selected, the used is asked to choose the method of examination: PRINTing to the interactor pane, EDITing in a text editing window, or editing in a POPUP form.

---

[2] If the AUTO-UPDATE? analysis switch is off, mass distributions are not recalculated when modifications are made.



Analysis: ANALYSIS1

GALLERY: SHIP-GALLERY
REPRESENTATION: GRAPH
BIAS: 50
FUZZ: 0.005
CONFLICT-MAXIMUM: 50
GISTING?: **Yes** No
GIST-LEVEL: 30
GIST-TYPE: **Full** Quick
NORMALIZE-MASS?: **Yes** No
AUTO-UPDATE?: **Yes** No
DRAWING-STYLE: ☐**Names**
☐ Operations
☐ Frame&Time
☐ Parameters
COMMENT: Initial analysis

<ABORT changes>, <SAVE changes>

Figure 4.4: An example analysis form

COMPILE transforms the interpretive analysis structures into self-contained independent subroutines (in C or Lisp). The resulting subroutines take numeric arguments (scalar or vector) as inputs, corresponding to the mass assigned propositional statements in selected primitive bodies of evidence, and produce numeric results corresponding to the support and plausibility assignments made by interpretation nodes. After selecting COMPILE, the user is asked to select one or more of the target languages; then the user is asked to specify the files where the resulting subroutines will be written; finally the user is asked to select any input nodes that are to be treated as constants (the resulting subroutines will have no inputs corresponding to these selected nodes). The key difference, between the computations performed by the resulting subroutines and the computations performed when analyses are evaluated, is that the logical questions that are posed relative to the gallery during analysis evaluation are compiled away in the subroutines. That is, the logical questions pertaining to the gallery are posed at compile time, their answers help shape the resulting numerical subroutines, but they are not posed during subroutine execution. As a direct result, the subroutines are more efficient, composed exclusively of numeric operations. With this feature, the user can make use of Gister-CL, with all of its development features, to develop an evidential line of reasoning and, when the user is satisfied, the COMPILE option can be used to produce an independent subroutine, embodying the evidential line or reasoning, for inclusion within other software systems.

Unless otherwise noted, all of the primed analysis options perform the same operations on a user selected subset of all of the analyses in the currently selected graph.

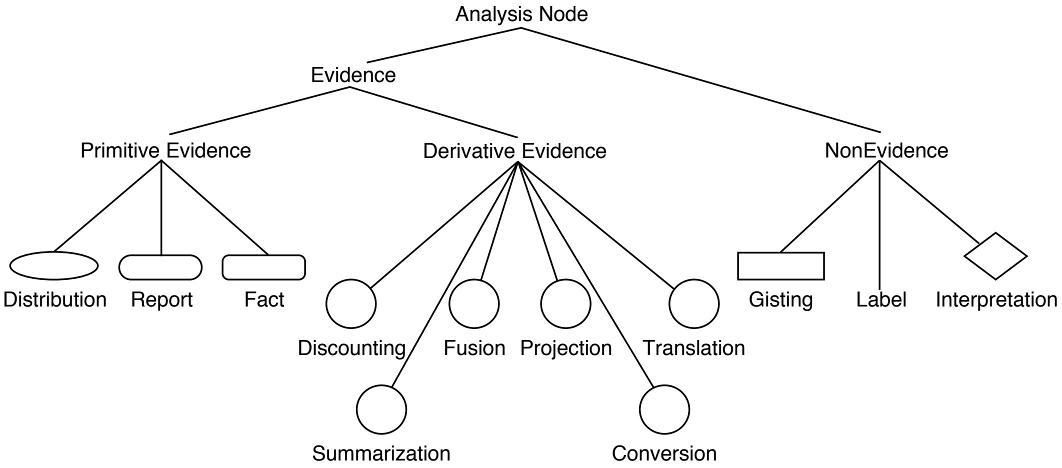


Figure 4.5: The analysis node hierarchy

4.3.4 The Evidence Options

The most frequently used Analyzer operations are accessed through the EVIDENCE menu. All of the nodes in an analysis are created and manipulated through these operations.

Analysis Nodes

The nodes in an analysis fall into several different hierarchically related categories (Figure 4.5). The first distinction is between evidence and nonevidence nodes. Evidence nodes have an associated mass distribution and nonevidence nodes do not. There are three categories of nonevidence nodes: rectangularly-shaped gisting nodes, diamond-shaped interpretation nodes, and label nodes that consist of a text string with no icon. The next distinction is between primitive and derivative evidence. Derivative evidence nodes derive their mass distributions from distributions in other (evidence) nodes, while the mass distributions in primitive evidence nodes do not rely upon information in any other nodes. Derivative evidence nodes are all circular and include conversion, discounting, fusion, projection, summarization, and translation nodes. Elliptically-shaped distribution and conditional-distribution nodes, capsule-shaped report nodes, and rectangular fact nodes, with slightly rounded corners, are all primitive evidence nodes. The mass distributions associated with standard distribution nodes are directly provided by the user, while the mass distributions associated with conditional distribution nodes are derived from a given distribution and a conditioning statement through conditional embedding; the mass distributions associated with report nodes are derived from a propositional statement and a credibility value[3] provided by the user; the mass distribution associated with a fact node attributes all of its mass to a single proposition provided by the user.

---

[3] A credibility value can be either a single number representing a point or a pair of numbers representing a closed interval.

All analysis nodes have COMMENTs and all analysis nodes, with the exception of labels, have an FOD and a BASIS. The FOD (frame of discernment) consists of a frame paired with a time, and a BASIS is a set of primitive evidence nodes. The FOD references the possibilistic information that underlies the information stored at each node; the BASIS refers to the primitive bodies of evidence that support the other derived information at each node.

Evidence nodes all have calculated mass distributions that are stored under MASS. A distribution node's mass distribution is the fully normalized version (i.e., normalized propositions and numbers) of its user supplied distribution (USER-MASS); a report node's mass distribution is derived from its propositional STATEMENT and CREDIBILITY; a fact node's mass distribution is derived from its propositional STATEMENT. Their FODs are formed from their FRAME and TIME; their propositional statements are normalized relative to their FODs before being included in their MASS distributions. For report nodes, an amount of mass equal to its CREDIBILITY, or to its lower bound, is associated with its normalized STATEMENT; if an upper bound is given, the difference between it and 100 is associated with the negation of the normalized STATEMENT; the remainder is associated with the true proposition (i.e., the disjunction of all possibilities) from its FOD. The mass distribution of a derivative evidence node is produced by applying its associated evidential operation to the mass distributions of the nodes connected to it by inwardly pointing arcs. A derivative evidence node's evidential operation is its TYPE and labels its inwardly pointing arcs.

Other entries contained in a derivative evidence node are unique to its type and represent either additional inputs or outputs of the associated operation. Discounting nodes have a DISCOUNT-RATE as an input; summarization nodes have an input SUMMARY-MINIMUM; the projection node input is DELTA-T; the translation node input is DELTA-F. A number of evidence nodes have CONFLICT as an additional output. The FODs for projection, translation, and conversion nodes are calculated based upon DELTA-T, DELTA-F, and DELTA-FOD, respectively, in conjunction with the FODs of their (inwardly) connected nodes. Conversion nodes support arbitrary changes to the GALLERY, FRAME, TIME, and REPRESENTATION of their FOD and corresponding MASS distributions. Discounting, fusion, and summarization nodes all directly inherit their FODs from their (inwardly) connected nodes.

Interpretation, gisting, and label nodes are nonevidence nodes that do not represent bodies of evidence. Therefore, they cannot be used as the basis for further evidential operations. Interpretation nodes have a set of PROPOSITIONS for which they calculate their supports, plausibilities, and representative likelihoods, relative to the mass distributions of the nodes connected to them by INTERPRET arcs. These results are paired with their respective propositions and appear in ascending order of their representative likelihoods in CONCLUSIONS. Gister nodes each have an optional GIST-LEVEL and GIST-TYPE. Based upon these values, or their values at the analysis level, and the mass distributions of the nodes connected to it by GIST arcs, a GIST and GIST-DISPLAY are calculated. The GIST-DISPLAY is a version of the GIST better suited for displaying to the user.

Manipulating Analysis Nodes

To view the contents of any of these analysis nodes, the user selects EXAMINE from the verb menu, then a node to be viewed. At the end of this procedure, a filled-in form appears[4]. The specific form used depends on the node's type (Figure 4.6). Once a form is displayed, the information in it can be modified by clicking on the entry that the user wants to change. Those entry keys that appear in all capital letters correspond to entries that are provided by the user; all other entries are derived[5]. A left click removes the current entry and awaits the user to provide a new one, while a middle click allows the user to edit the current entry[6]. When all of the desired changes have been made, the user clicks on the SAVE box. If the user changes an entry that is normally calculated by the Analyzer, the change will be lost the next time that node is evaluated. The modified node and all of its descendants are reevaluated (in accordance with the AUTO-UPDATE? analysis switch) after the user clicks on the SAVE box.

Left clicks on some user provided entries cause temporary menus to appear. In particular, some propositional STATEMENTs are collected from the user utilizing a menu containing a frame's aliases and elements. Clicking on these entries selects (or deselects) them for inclusion in the proposition. Selecting the END option exits this menu with the currently selected elements and aliases defining the proposition; selecting the EDIT option gives the user an opportunity to edit the proposition using Emacs-like control characters before it is returned. Other entries collect mass distributions (USER-MASS) from the user using these propositional menus along with menus for mass collection. The user is repeatedly asked to select a proposition followed by a portion of mass is to be assigned that proposition (from 0 to 100 percent). If a number is given, that portion of mass is assigned to that proposition; otherwise the remaining unassigned mass will be attributed to it. This repeats until a complete mass distribution has been defined.

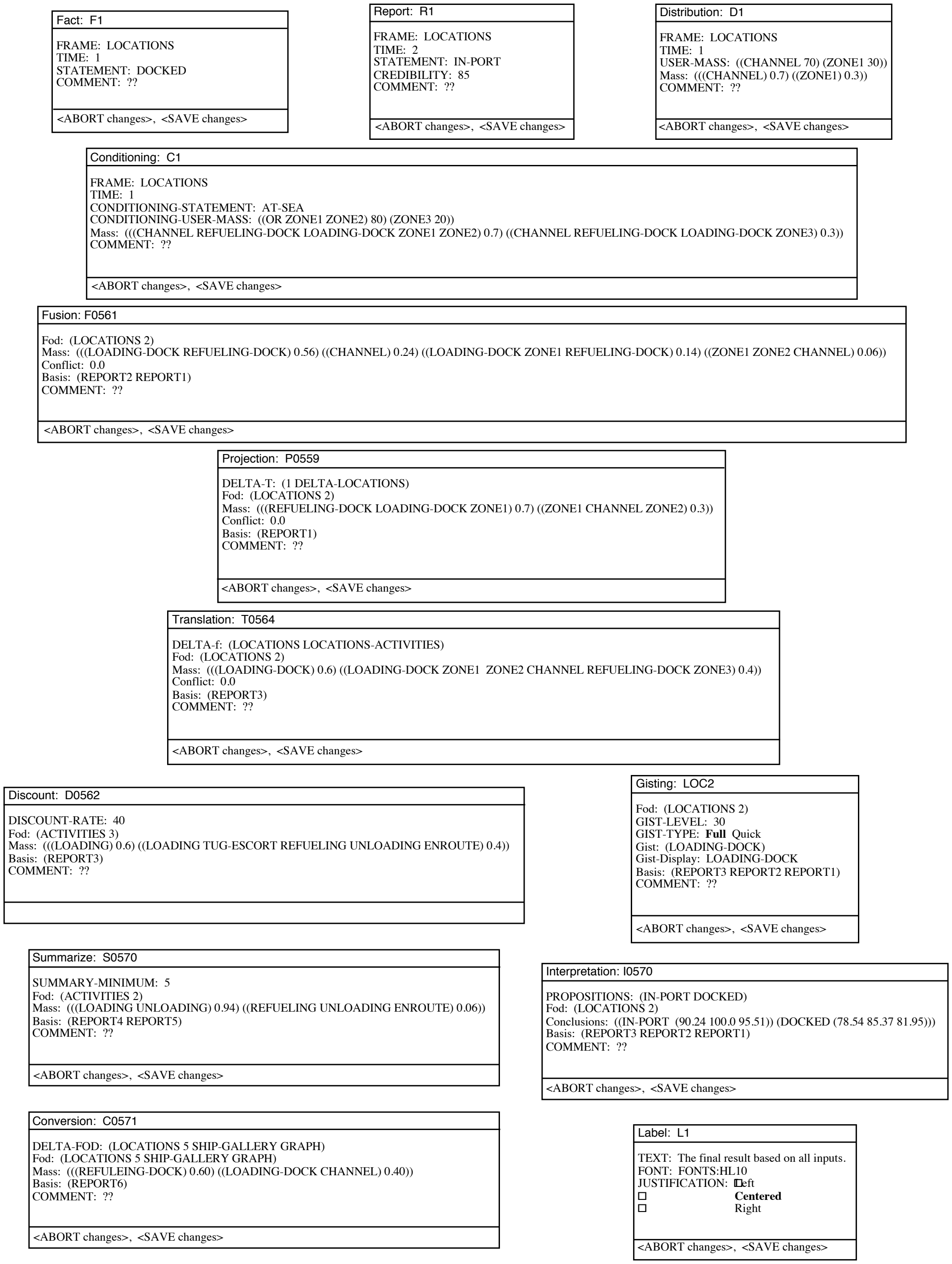
When EXAMINE' is invoked, it first asks the user to choose the type of information that is to be examined. Once the type of information has been selected, the used is asked to choose the method of examination: PRINTing to the interactor pane, EDITing in a text editing window, or editing in a POPUP form. Then the user selects a node whose information is to be viewed. PRINTing causes the information to be displayed, but with no way to modify it; EDITing prints the information to the interactor pane and allows the user to modify it though Emacs-like control characters until editing is terminated by the Return key; POPUP responds exactly as the nonprimed examine option only restricted to the selected type of information.

---

[4] If the selected node is an evidence node and the GISTING? switch is appropriately set, the gist of the selected node's mass distribution and that proposition's support, plausibility, and representative likelihood are graphically displayed in the Lisp Listener window. If the selected node is an interpretation node, the support, plasuibility, and representative likelihood of each of its PROPOSITIONS are graphically displayed in the order determined by their representative decision likelihoods. These likelihoods depend upon the setting of the analysis' BIAS.

[5] Although Gister-CL will allow system derived entries to be modified by the user, those modifications will be lost the next time the corresponding node is evaluated.

[6] Care must taken by the user when modifying entries, since few additional checks are made to verify the validity of these changes.



Fact: F1

FRAME: LOCATIONS
TIME: 1
STATEMENT: DOCKED
COMMENT: ??

<ABORT changes>, <SAVE changes>

Report: R1

FRAME: LOCATIONS
TIME: 2
STATEMENT: IN-PORT
CREDIBILITY: 85
COMMENT: ??

<ABORT changes>, <SAVE changes>

Distribution: D1

FRAME: LOCATIONS
TIME: 1
USER-MASS: ((CHANNEL 70) (ZONE1 30))
Mass: (((CHANNEL) 0.7) ((ZONE1) 0.3))
COMMENT: ??

<ABORT changes>, <SAVE changes>

Conditioning: C1

FRAME: LOCATIONS
TIME: 1
CONDITIONING-STATEMENT: AT-SEA
CONDITIONING-USER-MASS: ((OR ZONE1 ZONE2) 80) (ZONE3 20))
Mass: (((CHANNEL REFUELING-DOCK LOADING-DOCK ZONE1 ZONE2) 0.7) ((CHANNEL REFUELING-DOCK LOADING-DOCK ZONE3) 0.3))
COMMENT: ??

<ABORT changes>, <SAVE changes>

Fusion: F0561

Fod: (LOCATIONS 2)
Mass: (((LOADING-DOCK REFUELING-DOCK) 0.56) ((CHANNEL) 0.24) ((LOADING-DOCK ZONE1 REFUELING-DOCK) 0.14) ((ZONE1 ZONE2 CHANNEL) 0.06))
Conflict: 0.0
Basis: (REPORT2 REPORT1)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Projection: P0559

DELTA-T: (1 DELTA-LOCATIONS)
Fod: (LOCATIONS 2)
Mass: (((REFUELING-DOCK LOADING-DOCK ZONE1) 0.7) ((ZONE1 CHANNEL ZONE2) 0.3))
Conflict: 0.0
Basis: (REPORT1)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Translation: T0564

DELTA-f: (LOCATIONS LOCATIONS-ACTIVITIES)
Fod: (LOCATIONS 2)
Mass: (((LOADING-DOCK) 0.6) ((LOADING-DOCK ZONE1 ZONE2 CHANNEL REFUELING-DOCK ZONE3) 0.4))
Conflict: 0.0
Basis: (REPORT3)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Discount: D0562

DISCOUNT-RATE: 40
Fod: (ACTIVITIES 3)
Mass: (((LOADING) 0.6) ((LOADING TUG-ESCORT REFUELING UNLOADING ENROUTE) 0.4))
Basis: (REPORT3)
COMMENT: ??

Gisting: LOC2

Fod: (LOCATIONS 2)
GIST-LEVEL: 30
GIST-TYPE: **Full** Quick
Gist: (LOADING-DOCK)
Gist-Display: LOADING-DOCK
Basis: (REPORT3 REPORT2 REPORT1)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Summarize: S0570

SUMMARY-MINIMUM: 5
Fod: (ACTIVITIES 2)
Mass: (((LOADING UNLOADING) 0.94) ((REFUELING UNLOADING ENROUTE) 0.06))
Basis: (REPORT4 REPORT5)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Interpretation: I0570

PROPOSITIONS: (IN-PORT DOCKED)
Fod: (LOCATIONS 2)
Conclusions: ((IN-PORT (90.24 100.0 95.51)) (DOCKED (78.54 85.37 81.95)))
Basis: (REPORT3 REPORT2 REPORT1)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Conversion: C0571

DELTA-FOD: (LOCATIONS 5 SHIP-GALLERY GRAPH)
Fod: (LOCATIONS 5 SHIP-GALLERY GRAPH)
Mass: (((REFULEING-DOCK) 0.60) ((LOADING-DOCK CHANNEL) 0.40))
Basis: (REPORT6)
COMMENT: ??

<ABORT changes>, <SAVE changes>

Label: L1

TEXT: The final result based on all inputs.
FONT: FONTS:HL10
JUSTIFICATION: ☐Left
☐ **Centered**
☐ Right

<ABORT changes>, <SAVE changes>

Figure 4.6: Example analysis node forms

Selecting EVALUATE from the verb menu followed by a node forces the selected node and all of its descendants to be reevaluated; selecting EVALUATE' followed by a node restricts the reevaluation to just the selected node. This option is particularly useful when the AUTO-UPDATE? analysis switch is off.

Portions of analyses can be destroyed and the remaining effected portions reevaluated through the DESTROY verb. Selecting DESTROY followed by a node, destroys that node and all of its descendants up to the first descendant fusion nodes. These fusion nodes are reevaluated along with their descendants (in accordance with the AUTO-UPDATE? analysis switch) once the appropriate portion of the analysis has been destroyed.

CREATE is used to create fact, report, distribution, conditional-distribution, gisting, and label nodes. After selecting CREATE, a temporary menu appears with these options. Selecting GISTING followed by a location in the analysis and a name, creates a gisting node with that name at that location. If the user selects LABEL from the temporary menu followed by a location in the analysis, a blank label form pops up for editing; it includes a string of TEXT to be displayed with the FONT and JUSTIFICATION to be used in its display. If the user selects FACT, REPORT, DISTRIBUTION, or CONDITIONAL-DISTRIBUTION from the temporary menu, a location in the analysis, followed by typing in a name, a blank form appears for the user to fill out. Exiting from this form results in a new primitive node.

PROJECT, TRANSLATE, DISCOUNT, SUMMARIZE, CONVERT, and INTERPRET each invoke an evidential operation that adds to an analysis, and each such operation depends upon a single input node and an additional parameter. After one of these verbs has been selected, the users next selects the node that contains the evidence to which the operation is to be applied, the location for the new node that will result, and the additional parameter. For PROJECT, this parameter is a positive or negative number of time units and an optional set of projection relations (DELTA-T); for TRANSLATE, a frame from the gallery and an optional set of translation relations (DELTA-F); for DISCOUNT, a discount percentage between 0 and 100 (DISCOUNT-RATE); for SUMMARIZE, a threshold percentage between 0 and 100 (SUMMARY-MINIMUM); for CONVERT this parameter (DELTA-FOD) consists of a vector of four possible changes to the GALLERY, FRAME, TIME, and REPRESENTATION of the FOD; and for INTERPRET, a set of propositions (PROPOSITIONS). In response to any of these operations, the Analyzer creates a new node containing the appropriate result. For all but INTERPRET, the result consists of an FOD and a MASS entry; for INTERPRET it is an FOD and a list of conclusions (CONCLUSIONS) that pairs each of its propositions with its evidential interval and representative decision likelihood.

Fusion nodes are created by selecting FUSE from the verb menu, followed by any number of nodes representing evidence over the same FOD, and then, a location for the new node. The Analyzer responds by drawing a new node that is connected by FUSE arcs to the selected nodes. The new fusion node contains the result of applying Dempster's rule to the associated mass distributions and the level of conflict. This conflict level indicates how incompatible the combined opinions are; it ranges between 0 and 100, with 0 indicating that the fused opinions are completely compatible. If this is close to 100, the user should examine the analysis carefully to ascertain and correct the source of conflict.

GIST allows gisting nodes to be created in much the same way as fusion nodes. One or more evidence nodes are selected, followed by a location, followed by a name for the node. As a result, a new gisting node is created at that location including the selected evidence nodes. Two of the options allow a gisting node to have conclusions INSERTed or REMOVEd. Each gisting node has a GIST that is the product of applying the gisting operation to its evidence (i.e., the mass distribution(s) stored at the node(s) connected to the cell via a GIST arc and the GIST-LEVEL (stored in the gisting node or the default stored for the analysis). To insert evidence into a gisting node, the user selects INSERT from the verb menu, then the evidence nodes to be inserted, then the gisting node itself. This causes the gisting node to be reevaluated, thereby recalculating its gist. The removal of a node from a gisting node is accomplished similarly. REMOVE is selected, then a gisting node, and finally the node to be removed. INSERT and REMOVE also can be used in conjunction with FUSION nodes to add additional bodies of evidence (with the same fod and nonintersecting bases) to a fusion operation or to remove bodies of evidence from a fusion operation.

A number of remaining operations enable the user to better understand evidential analyses by helping him to unravel the chains of reasoning. The ANCESTORS option highlights those contributing bodies of primitive evidence and intermediate conclusions that impinge on the conclusions represented by a selected node. The converse option, DESCENDANTS, highlights those conclusions that depend upon the evidence at a selected node. In other words, the descendants of a body of evidence are those conclusions that may be sensitive to changes in the selected body of evidence, while the ancestors are those that may change the selected conclusion if they themselves are changed. If one of these operations has been executed, the scrolling facility can be used to move both vertically and horizontally so as to view the entire chain of highlighted nodes. In both cases, the highlighted nodes are unhighlighted by repeating the operation.

Utilizing the ANCESTORS option, the user can easily identify which primitive and derivative bodies of evidence contribute to a selected conclusion. However, this does not reveal the relative impact of these bodies of evidence on the conclusion. This is the role of Gister-CL's explanation facility that is invoked by selecting the EXPLAIN option followed by an evidence or interpretation node that is to be explained. The Analyzer responds by highlighting that node's tree of ancestors; the explanations will be in terms of the impact of the leaves of the ancestor tree on the selected node. The depth of the tree, and thereby the terms of the explanation, can be trimmed back to nodes higher in the ancestor tree by simply selecting them; in response, the highlighted tree will be trimmed back to the selected nodes. When the user is satisfied with the depth of the highlighted ancestral tree, he selects a location on the background; in response, Gister-CL performs a sensitivity analysis.

If the root of the ancestral tree is an evidence node, when the sensitivity analysis is complete, the user is presented with a temporary menu of options. If "Influential" is selected, then those leaves are highlighted whose individual absence would substantially impact the evidential results at the root node; "Inconsequential" highlights those leaves whose absence would have little impact on the results. Selecting "Agree" highlights those leaves that confirm the root consensus while "Disagree" highlights those that are conflicting. The final option in this temporary menu, "Scatter Plot", produces a two dimensional plot. The horizontal axis is *specificity* and the vertical axis is *consonance*; the plot is a unit square with the (0,0) origin at the bottom left. The root node's specificity and consonance is marked by a vertical and a horizontal line. For each leaf in the selected ancestral tree, a point is plotted. This point corresponds to the specificity and consonance of the root consensus that would result if the evidence at that leaf were removed. Thus, the disposition of these points, relative to the cross hairs of the root consensus, reflects the impact of each of these leaves on the consensus. Since these points and their labels may be overlapping in the plot, the mouse can be used to reposition the labels; while a label is selected its corresponding point is highlighted.

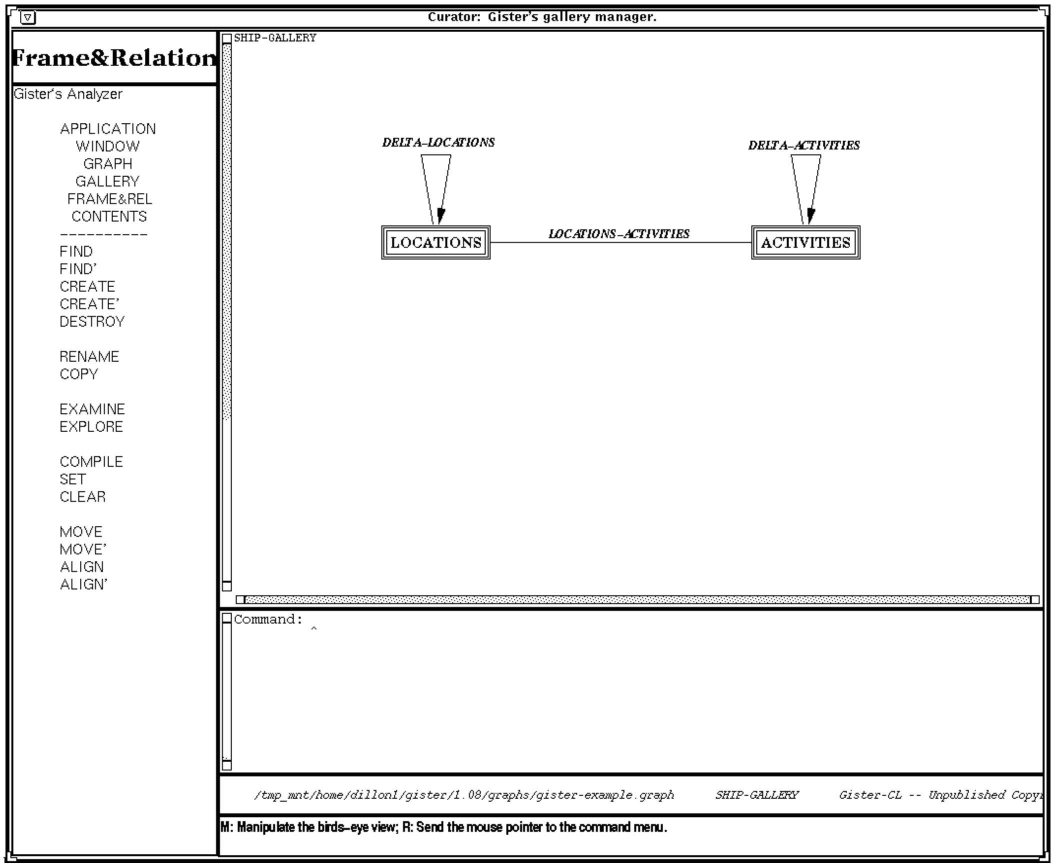
On the other hand, if the root of the selected ancestral tree is an interpretation node, the explanation facility will successively display support-plausibility triangle plots for each proposition included in the interpretation node's PROPOSITIONS. These two-dimensional plots use support and plausibility as as their unit length axes, forming two sides of an isosceles triangles that rests on its hypotenuse. The support and plausibility pairs that result at the interpretation node, when each of the leaf nodes are removed from the ancestral tree and when all of the evidence is considered, are plotted on the triangle. The pair corresponding to all of the evidence is plotted as two lines emanating from each of the axes; the other pairs are each plotted as labeled points. Thus the relative position of the points to the intersection of the lines reflects the impact of each leaf body of evidence on the proposition. The labels can be repositioned just as in the scatter plots.

Together these facilities support a “what if” style of reasoning, which makes it possible to construct and compare several alternative analyses before deciding on the optimum line of reasoning.

The remaining command options are the same as those found in Grasper-CL. FIND and FIND' allow the user to quickly find nodes or arcs based upon their names. RENAME allows the user to edit the names of selected nodes. MOVE, MOVE', ALIGN, and ALIGN' provide the user with a means of repositioning groups of nodes and arcs.

4.4 The Curator

The Curator is the manager of Gister-CL's galleries. Within Gister-CL, it is invoked by selecting Gister's Curator from the application menu or from the top of the Analyzer's command menu; there is an analogous entry at the top of the Curator's command menu for returning to the Analyzer. The Curator's noun menu includes the following options: APPLICATION, WINDOW, GRAPH, GALLERY, FRAME&REL (i.e., frame or relation), and CONTENTS (Figure 4.7). Upon selecting FRAME&REL or CONTENTS, if a gallery is not already displayed, a gallery from the currently selected graph is drawn in the graph pane; if more than one gallery is in the current graph, the user may be asked to select one from a temporary menu; if the current graph does not contain a gallery, the user is asked for a name so that a new one can be created. Selecting WINDOW, GRAPH, GALLERY, or FRAME&REL immediately exposes its associated verb menu; the verb menu for CONTENTS does not appear until after both CONTENTS and an item from the gallery (a node or an arc from the gallery in the graph pane) are selected (Figure 4.8).



```
/tmp_mnt/home/dillon1/gister/1.08/graphs/gister-example.graph     SHIP-GALLERY     Gister-CL -- Unpublished Copy
M: Manipulate the birds-eye view; R: Send the mouse pointer to the command menu.
```

Figure 4.7: Gister-CL's Curator display

Window

Gister's Analyzer

APPLICATION
WINDOW
GRAPH
GALLERY
FRAME&REL
CONTENTS

RESIZE
PANE-LAYOUT
BIRDS-EYE

Graph

Gister's Analyzer

APPLICATION
WINDOW
GRAPH
GALLERY
FRAME&REL
CONTENTS

SELECT
CREATE
DESTROY
DESTORY'

INPUT
MERGE
OUTPUT

BACKUP
REVERT
BACK-REV-UNBACK

PRINT-DRAW

MAGNIFY

Gallery

Gister's Analyzer

APPLICATION
WINDOW
GRAPH
GALLERY
FRAME&REL
CONTENTS

SELECT
CREATE
DESTROY
DESTROY'

RENAME
RENAME'
COPY
COPY'

BACKUP
BACKUP'
REVERT
REVERT'

EXAMINE

PRINT-DRAW
PRINT-DRAW'
REDRAW
RESCALE

COMPILE
COMPILE'
SET
SET'
CLEAR
CLEAR'

Frame&Relation

Gister's Analyzer

APPLICATION
WINDOW
GRAPH
GALLERY
FRAME&REL
CONTENTS

FIND
FIND'
CREATE
CREATE'
DESTROY

RENAME
COPY

EXAMINE
EXPLORE

COMPILE
SET
CLEAR

MOVE
MOVE'
ALIGN
ALIGN'

Contents

Gister's Analyzer

APPLICATION
WINDOW
GRAPH
GALLERY
FRAME&REL
CONTENTS

TO-GALLERY

FIND
FIND'
CREATE
CREATE'
DESTROY
DESTROY'
RENAME

EXAMINE
EXPLORE

PRINT-DRAW
REDRAW
RESCALE

MOVE
MOVE'
ALIGN
ALIGN'

Figure 4.8: The Curator's menus

4.4.1 The Window and Graph Options

Both the WINDOW and GRAPH options are the same as those found in the Analyzer. The WINDOW options include RESIZEing the graph pane, changing the PANE-LAYOUT of the window, and setting parameters for the BIRDS-EYE display. GRAPHs can be SELECTed, CREATEd, DESTROYed, INPUT, MERGEd, OUTPUT, BACKedUP, REVERTed, BACKedup-REVerted-UNBACKedup, PRINTed-or-DRAWn, or MAGNIFYed. Each of these perform the same basic operation as their counterpart under the Analyzer or Grasper-CL.

4.4.2 The Gallery Options

The Curator's GALLERY options are reminiscent of the Analyzer's ANALYSIS options. The verb menus are the same except that the Curator's includes options for COMPILEing, SETting, and CLEARing, and does not include the EVALUATE options. Each of the common options performs the same basic function, the Curator's on galleries and the Analyzer's on analyses, except that some of the GALLERY options influence other portions of the graph beyond the space that is visible in the graph pane. This reflects the structure of a gallery. A gallery is represented by a collection of Grasper-CL spaces: one where each frame is represented by a node and each compatibility relation by an arc, and another space for each of the gallery's frames and compatibility relations that represents their contents. Thus, when a gallery is DESTROYed, COPYed, BACKedUP, or REVERTed, all of these spaces are taken into account. The other GALLERY options, SELECT, CREATE, RENAME, EXAMINE[7], PRINT-DRAW, REDRAW, and RESCALE, do not take other spaces into account; they are completely analogous to their ANALYSIS counterparts.

The remaining GALLERY options manipulate the different frame and compatibility representations. At any time, the contents of each gallery component (i.e., every frame and compatibility relation) is captured using one or more different representational schemes. The GRAPH representation is the most commonly used and is the basis for several of the other representations in the sense that these other representations can be automatically generated through compilation of the GRAPH representation. The COMPILE command option asks the user which of these other representations is the target and then compiles every gallery component that includes a GRAPH representation, producing a new version of the contents in the target representation. Once multiple representational options exist for gallery components, the default representation to be utilized can be selected by using the SET option. After selecting SET the user is asked to select a default representation; every gallery component that has this as a representational option has its default representation changed. Finally, representational options that have been produced through compilation can be destroyed through the CLEAR option. The user is asked which representation is to be cleared; the selected representation is removed from all gallery components. In addition, CLEAR can be used to destroy the default propositional statements that are associated with each gallery component by selecting "Propositions" from the temporary menu rather

---

[7] The GALLERY EXAMINE option is included primarily for upwards compatibility with forthcoming features. At present, there is little useful information available through this option.

than a representation. All of these gallery options have primed counterparts that do the same for a selected subset of the galleries in the currently selected graph.

4.4.3 The Frame-Relation Options

The FRAME&REL options are used to populate the selected gallery with FRAMEs and compatibility RELations. Using the CREATE option, the user selects a location in the gallery, followed by a name, to create a frame with that name at the selected location. It will default to using the GRAPH representation. Compatibility reations are created by selecting a frame (i.e., a node in the gallery), an optional sequence of intermediate points, another frame, and a name, after choosing the CREATE option. In response, an arc representing a compatibility relation with the designated name is created between the selected frames. The arc goes through the intermediate points. If the two frames are different, the compatibility relation is a translation relation (i.e., one used to translate evidence from one frame to another) and the arc has no arrowhead. If the two selected frames are the same, the user is asked if the compatibility relation is to be a projection relation or a translation relation. A projection relation is one used to project evidence over time and is represented by an arc with an arrowhead. These compatibility relations will default to the GRAPH representation unless they connect frames that do not include the GRAPH representation as an option; in that case, if the two frames have a common selected representation, that will be the representation of the compatibility relation; otherwise, the representation will default to be FUNCTIONAL.

Four additional options are available through CREATE'. The "Create Frames or Relations [Representation]" option allows the user to choose the representation for the frames and relations that are to be created, rather than using the default representation. Another option, "Relocate Relation Edge," allows a previously created (relation) edge to be relocated. After selecting "Relocate Relation Edge" from the temporary menu, the user selects a frame, an optional sequence of intermediate points, and a second frame. If there is more than one compatibility relation between the two selected frames, the user will be asked to choose one. In either case, the appropriate edge will be redirected between the selected nodes passing through the intermediate points.

The remaining options under CREATE' create or augment the graph representations of frames and compatibility relations. Therefore, they require that the gallery components involved have graph representations. For example, the "Macro Create Aliases" option adds aliases to a selected frame's graph representation; the aliases added are those elements and aliases from a neighboring frame. After selecting "Macro Create Aliases", the user is asked to select a frame followed by one of its compatibility relations. It is the elements and aliases from that frame that are added as aliases.

Another option is invoked by selecting "Macro Create Frames or Relations" from the CREATE' temporary menu. If this is followed by the selection of a sequence of compatibility relations (terminated by selecting a location) where each relation shares a common frame with the preceding relation in the sequence, either a single new relation or new frame (with connecting relations) is created, as determined by the user's response to another temporary menu. If a new relation is created, it connects the frames, connected to the first and last selected compatibility relations, that were not common to their neighboring relations in the sequence. The new relation is defined in such a way as to produce the same conclusion, when translating/projecting across it, as would result from a multiple step translation/projection using the original relations in sequence. On the other hand, if the user selects to create a new frame, then the new frame is defined to be the dual of the compatibility relation that would have been created in the former case; this new frame is connected by a translation relation to each of the frames that connect the selected compatibilities relations in sequence.

Finally, if a collection of frames are selected (terminated by selecting a location), after selecting "Macro Create Frames or Relations" from the CREATE' temporary menu, then a new frame is created that corresponds to the cross product of the selected frames. Each of the selected frames are connected to the new frame by a trivial translation relation. Together these macro creation options allow the user to move freely between the dual representations for compatibility relations (as discussed in Section 2.1.2).

Frames can also be created using the COPY option. After selecting COPY from the FRAME&REL menu, the user selects an existing frame, a location, and a name. An exact copy of the selected frame, including its different representations, is created under the new name at the selected location. The RENAME option allows these or other frames and compatibility relations to be renamed by designating the entity to be renamed and a new name to be used. The Curator guarantees the uniqueness of frame and relation names across the entire graph.

Another option allows frames and compatibility relations to be DESTROYed. After selecting DESTROY, the user designates a frame (i.e., node) or relation (i.e., edge); the designated frame or relation is destroyed; if the designated frame has connecting relations, they are destroyed as well.

Frames and relations are examined by selecting them while operating under the EXAMINE option. This exposes a temporary menu that indicates the frame's or relation's default REPRESENTATION, the different REPRESENTATION-OPTIONS that have been defined for it, an optional SHORT-NAME that is an abbreviated version of its name that is used by some ANALYSIS drawing styles, a COMMENT string, and, for frames, a PROPOSITION that is/was used during the EXPLOREation option. All but the REPRESENTATION-OPTIONS are modifiable through this menu.

The COMPILE, SET, and CLEAR options are identical to those found in the GALLERY menu, expect that these apply to singularly selected frames or compatibility relations. The user select the option, followed the frame or compatibility relation, followed by the target representation.

The EXPLORE option allows the user to enter a propositional statement at a selected frame, project or translate that proposition, and then examine the result. In this way, the user can come to understand how the gallery is functioning. After selecting EXPLORE, the user selects a frame at which to begin. The PROPOSITION currently stored at that frame is displayed in a temporary menu. The user has the option of editing that proposition before exiting the menu. Upon exiting the proposition is stored at the frame, and depending on if compatibility relations are connected to that frame and their type, the user will be given up to three options: "Translate", "Project" some number of time units forward or backward, and "Project[+1]" i.e., project one unit forward. If "Translate" is selected and there is more than one frame connected by a translation relation, the user will be asked to designate the target frame and, then, if there is more than one translation relation leading to that target frame, the user will be asked to select the relation; if "Project" is selected, the user will be asked for a positive or negative number of time units, followed by a choice of projection relations if there is more than one; if "Project[+1]" is selected, the user will be asked to choose a projection relation if more than one exists. Following this, the translation or projection is performed, the resulting proposition is stored at the target frame and displayed in a temporary menu, allowing the user to quit or modify it and then translate or project again. All of these operations are performed using the default REPRESENTATION at each frame and relation.

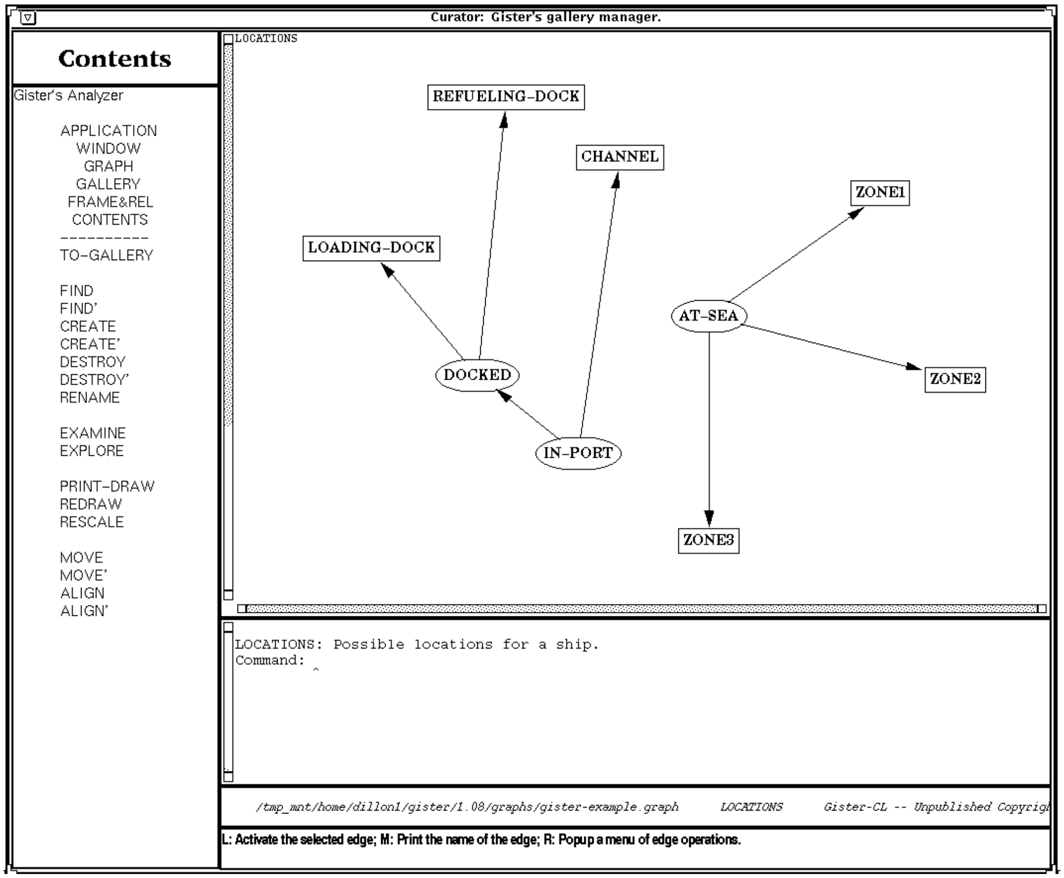
The remaining options are the same as those found in Grasper-CL and the Analyzer. FIND and FIND' allow the user to quickly find nodes or arcs based upon their names. MOVE, MOVE', ALIGN, and ALIGN' provide the user with a means of repositioning groups of nodes and arcs.

4.4.4 The Contents Options

The user employs the CONTENTS options to populate frames and compatibility relations created under the FRAME&REL options. After selecting CONTENTS from the noun menu, the user is asked to select a frame (i.e., node) or compatibility relation (i.e., edge) in the gallery. What happens next depends on the default REPRESENTATION of the selected item. Except were otherwise noted, the following discussion assumes that the GRAPH representation is being used.

After selecting CONTENTS followed by a frame or relation, the graph pane changes to the contents of the selected entity and the verb menu appears. The verb menu is the same no matter what type of entity from the gallery is selected, but it represents different operations depending on the selected entity. If a frame is selected under the CONTENTS option, its elements appear as rectangular nodes and its aliases as elliptical nodes with directed arcs pointing to their disjuncts (Figure 4.9). To create a new element, the CREATE option is selected followed by a location and a name; to create a new alias, the CREATE option is selected followed by some number of elements or aliases, a location, and a name. There are three CREATE' options invoked by making the appropriate selection from a temporary menu: (1) "Relocate Alias Edge" that works just like relocating a relation edge under FRAME&REL CREATE', (2) "Add Alias Edge" asks the user to select an alias followed by a new element or alias to be included in its definition, and (3) "Creating Element and Update Relations" which creates a new element while simultaneously updating the compatibility relations in which this frame participates (the new element is connected to every possible element or no elements in the related frames, at the discretion of the user).

To modify the name given a frame element or alias, the user selects RENAME followed by the appropriate node and edits or replaces the current name. To destroy frame elements or aliases, the user selects DESTROY followed by the appropriate node. In the case of aliases, the arcs connected to them are destroyed as well. The DESTROY' option allows alias edges to be individually destroyed by pointing at two nodes in sequence (at least one

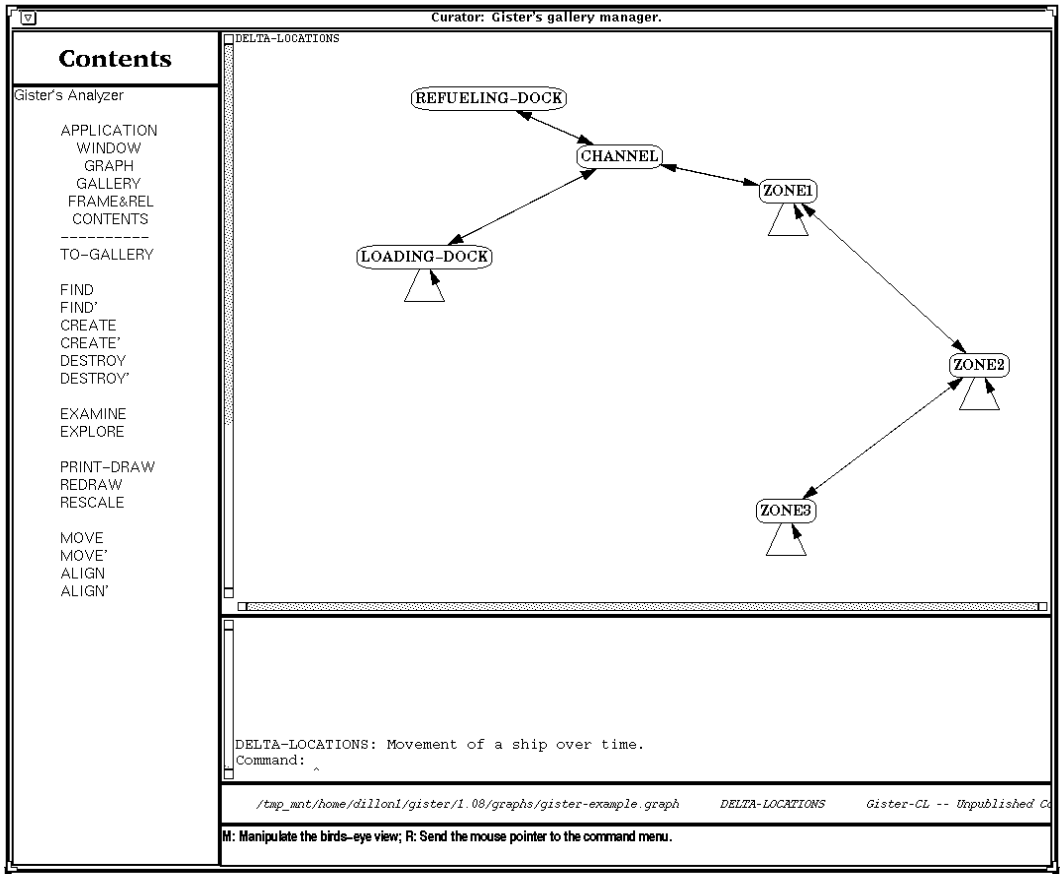


```
LOCATIONS: Possible locations for a ship.
Command:
```

/tmp_mnt/home/dillon1/gister/1.08/graphs/gister-example.graph LOCATIONS Gister-CL -- Unpublished Copyrig

L: Activate the selected edge; M: Print the name of the edge; R: Popup a menu of edge operations.

Figure 4.9: Contents of a frame



```
DELTA-LOCATIONS: Movement of a ship over time.
Command:
```

/tmp_mnt/home/dillon1/gister/1.08/graphs/gister-example.graph DELTA-LOCATIONS Gister-CL -- Unpublished C

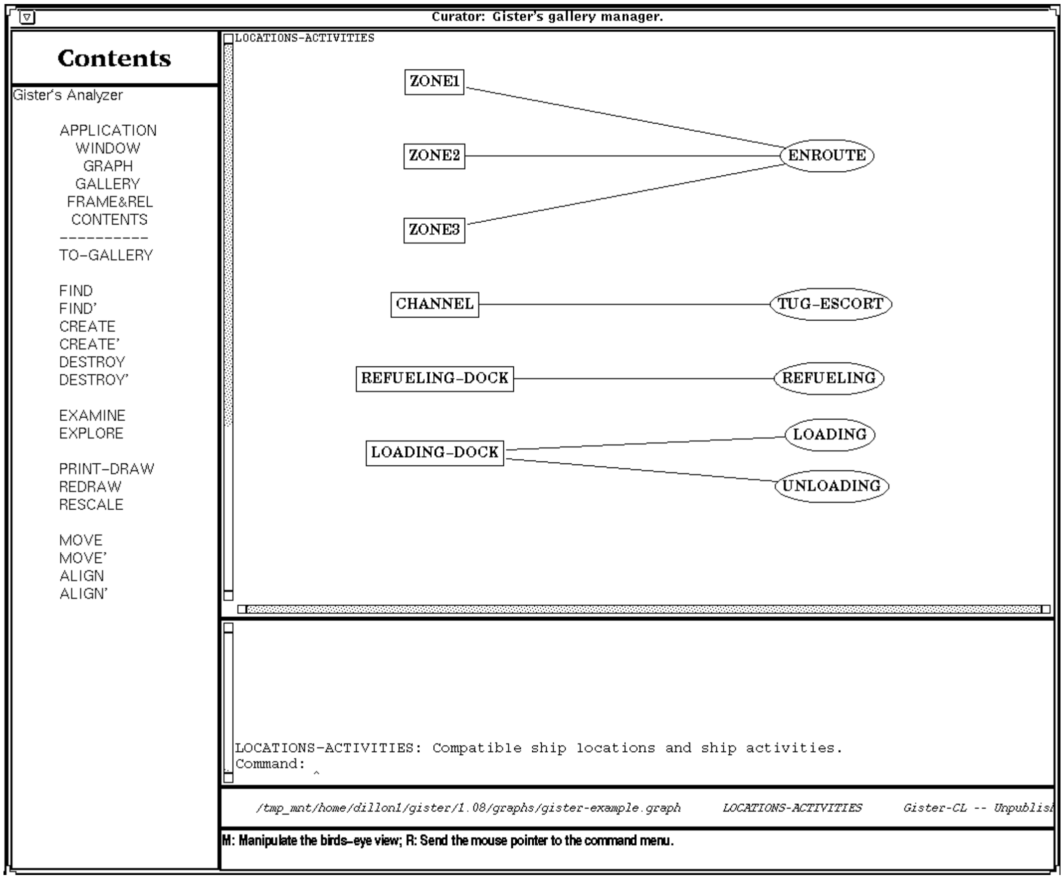
M: Manipulate the birds-eye view; R: Send the mouse pointer to the command menu.

Figure 4.10: Contents of a projection relation

of which is an alias); the alias edge between the nodes is destroyed. Alias edges can also be destroyed by selecting DESTROY and then the alias edges.

If the user selects a projection relation after selecting CONTENTS, the elements from the frame it references are displayed as capsule-shaped nodes and the individual elements of the projection relation are displayed as directed arcs among the nodes (Figure 4.10). Using the CREATE option, a new relationship is added by selecting a node, an optional sequence of intermediate points, and another node; an arc is created from the first to the second through the intermediate points. If the CREATE' option is used instead, three additional options are available: (1) "Relocate Relationship Edge" that allows the user to redirect an edge through different intermediate points, (2) "Create All Possible Relationships" that fully connects every node with every other node, and (3) "Create Multiple Relationships" that allows the user to select a base node followed by any number of nodes to which the base node is to be related.

Using the DESTROY and DESTROY' options, elements of a projection relation can be destroyed. The DESTROY option allows the user to either designate the edge to be destroyed directly or by selecting the nodes at either end of the edge. If more than one edge exists between the selected nodes, the edge pointing away from the first node and toward the



```
LOCATIONS-ACTIVITIES: Compatible ship locations and ship activities.
Command:
```

/tmp_mnt/home/dillon1/gister/1.08/graphs/gister-example.graph *LOCATIONS-ACTIVITIES* *Gister-CL -- Unpublis*

M: Manipulate the birds-eye view; R: Send the mouse pointer to the command menu.

Figure 4.11: Contents of a translation relation

second is the one destroyed. The DESTROY' option supports the destruction of all edges in the relation, i.e., "Destroy All Relationships", or through "Destroy Element's Relationships" either "All Outpointing Edges" from a designated node (i.e., all edges pointing away from that node), "All Inpointing Edges" from a designated node, or "All Edges" connected to a designated node (regardless of direction).

The last option to be consider after having selected CONTENTS is the selection of a translation relation. A translation relation is drawn with rectangular nodes representing the elements from one of the two related frames, elliptical nodes for the elements from the other frame, and capsule-shaped nodes for elements that are identically named in both frames (Figure 4.11 contains a modified version of the LOCATIONS-ACTIVITIES relation used in an earlier chapter; REFUELING-DOCK has been renamed REFUELING and LOADING-DOCK has been renamed LOADING in the LOCATIONS frame; therefore, both REFUELING and LOADING are contained in both the LOCATIONS and ACTIVITIES frames.). Arcs connect compatible elements from different frames. If an element is in both frames, it can have an arc that connects it to itself (i.e., to its alter ego), to other elements that are in both frames, or to any other element.

The CREATE and CREATE' options for translation relations are the same as those available for projection relations. The DESTROY and DESTROY' options are analogous, except that multiple arc operations from a single node use the frames of the nodes to which the edges lead as the distinguishing factor, rather than the direction of the edges.

The remaining CONTENTS options are the same across frames, projection relations, and translation relations. EXAMINE allows the contents of all frame and relation entities to be viewed by selecting them after selecting EXAMINE. PRINT-DRAW, REDRAW, and RESCALE are completely analogous to their GALLERY counterparts. FIND and FIND' allow the user to quickly find nodes or arcs based upon their names. MOVE, MOVE', ALIGN, and ALIGN' provide the user with a means of repositioning groups of nodes and arcs.

Finally, at the head of the CONTENTS verb menu is a TO-GALLERY command. Selecting this option returns the user to the gallery from which the gallery component being displayed has come.

4.5 Learning Gister-CL's User Interface

Although this exposition on the operation of Gister-CL may make it seem complex, the system has proved fairly easy to learn and use. Although we have done no formal testing, it is our experience that novices quickly become productive after just a brief introduction (a few hours). We attribute this to the graphic nature of the interface, which attempts to exploit the intuitions of the user. In addition, a documentation line at the bottom of the screen furnishes useful guidance throughout all the phases of operation. In general, people seem to grasp the basic concepts quickly.

It is recommended that a novice begin by INPUTing a previously developed gallery and set of analyses. First, use the SELECT and EXAMINE options in the Curator and Analyzer to explore the gallery and analyses. Second, use the Analyzer's EVIDENCE options to modify an analysis. Try modifying some reports or primitive evidence nodes using EXAMINE; develop a distinct line of reasoning from the same reports and primitive evidence using FUSE, PROJECT, TRANSLATE, DISCOUNT, SUMMARIZE, and INTERPRET; use ANALYSIS REVERT to restore the analysis to its original state. Third, modify portions of the gallery using the Curator's FRAME&REL and CONTENTS options. Using CONTENTS CREATE, add a new element and alias to a frame and update the compatibility relations in which that frame participates; using FRAME&REL CREATE and CONTENTS CREATE, create a new frame with a projection relation and a translation relation. Fourth, use the Analyzer to reEVALUATE an analysis under the modified gallery; try TRANSLATEing some bodies of evidence to the newly created frame and PROJECT the results; create a completely new analysis based upon new reports and new lines of reasoning; use GRAPH REVERT to restore the entire graph to its original state. Finally, use the Curator and Analyzer to develop a completely new application.

Chapter 5

Gister-CL's Programmer Interface

All of the operations that are supported by Gister-CL's user interface are also accessible through its programmer interface. The programmer interface consists of a number of specialized Gister-CL procedures that work in conjunction with Grasper-CL's programmer interface. All Gister-CL procedures and data structures reside in the `ER` package[1]. All symbols listed in the documentation are exported from the `ER` package.

Gister-CL's data structures are restricted subclasses of Grasper-CL's data structures that are constrained to have certain characteristics. As such, the basic concepts that underlie Grasper-CL (such as nodes, edges, graphs, current graph, and current space) apply also to Gister-CL. One important consequence of this architecture is that a user can employ Grasper-CL as well as Gister-CL to manipulate and define these data structures. However, it is up to the programmer to maintain the consistency of Gister-CL's data structures when manipulating them through Grasper-CL. Manipulation of these data structures through Gister-CL's specialized procedures is preferable, since they help to maintain consistency. However, even these procedures assume that they are given consistent arguments. It is up to the programmer to guarantee this!

Further documentation is available for users who are interested in learning more about Grasper-CL. The *Grasper-CL User's Guide* (in [KLS93]) provides a thorough overview of the foundations and use of Grasper-CL. The *Grasper-CL Programmer's Manual* (also in [KLS93]) provides descriptions of the COMMON LISP procedures that comprise Grasper-CL.

Implementing an evidential reasoning system can be divided into two independent subproblems: How to represent mass distributions and perform numeric calculations on them? How to represent propositions and perform logical inferences? Thus, Gister-CL implements evidential reasoning as two independent components: the Analyzer, that manipulates and interprets mass distributions and, the Curator, that performs logical reasoning. In the following discussion we attempt to highlight this by recasting the mathematical definitions of

---

[1] `ER`, prior to version 2.05, was a nickname for the `Grasper-CL` package. For versions 2.06 and beyond, the `ER` package is distinct from the `Grasper-CL` package; although it "uses" the `Grasper-CL` package. To update a graph output under version 2.05 or earlier, edit the first line of the graph file changing it from (in-package :gsp) to (in-package :er); input it into Gister-CL; then output it.

the evidential reasoning operations in terms of propositional logic, rather than set theory. The best suited implementations depends upon the characteristics of the domain of application and upon the characteristics of the host computational environment. We outline some implementation alternatives as examples, but many others are possible.

5.1 Mass Distributions and Numeric Calculations

The fundamental numeric representation for evidential reasoning is the mass distribution. A mass distribution divides a unit of mass over propositions from a frame of discernment. Thus, the number of possible nonzero assignments is bounded by the number of possible propositional statements in the selected frame. For typical applications of evidential reasoning, the frames of interest are discrete and the number of nonzero assignments relatively small. Therefore, we often use a tabular representation for mass distributions.

The tabular representation utilized in Gister-CL is a linked list of pairs. Each pair corresponds to a nonzero mass assignment within a mass distribution. One element of the pair is a proposition from the selected frame of discernment and the other element is a number corresponding to the portion of mass assigned to that proposition. This tabular representation of a mass distribution is suitable regardless the representation used for the propositions. Thus, a mass distribution m_{A}, that assigns nonzero mass to propositions $\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_{\mathsf{z}}$, can be represented by the following linked list[2]:

$$((\mathsf{A}_1, n_1), (\mathsf{A}_2, n_2), \ldots, (\mathsf{A}_z, n_z)) \quad ,$$

where

$$\begin{aligned} \sum_i n_i &= 1 \\ n_i &= m_{\mathsf{A}}(\mathsf{A}_i) \\ n_i &> 0 \\ \mathsf{A}_i &\neq \mathsf{FALSE} \quad . \end{aligned}$$

In the literature, frames of discernment and the logical operations over them have typically been defined in terms of set theory. Using this approach, a frame of discernment consists of a set of mutually exclusive and exhaustive answers or values for some question or variable. For example, if the possible answers for a question A are $a_1, a_2, \ldots, a_n$, then propositional statements pertaining to the answer to A are in one-to-one correspondence with subsets of these answers. If we call the set of possible answers Θ_A, then a propositional statement A_i pertaining to the answer to question A corresponds to A_i, a subset of Θ_A.

$$A_i \subseteq \Theta_A = \{a_1, a_2, \ldots, a_n\} \quad .$$

---

[2] The linked list is written using LISP syntax.

Given this correspondence between propositions A_i and sets A_i, propositional statements and connectives have their corresponding sets and operations:

$$\begin{array}{rcl}
\mathsf{FALSE} & \Longleftrightarrow & \emptyset \\
\mathsf{TRUE} & \Longleftrightarrow & \Theta_A \\
\mathsf{A}_i & \Longleftrightarrow & A_i \\
||\mathsf{A}_i|| & \Longleftrightarrow & |A_i| \\
\neg\mathsf{A}_i & \Longleftrightarrow & \Theta_A - A_i \\
\mathsf{A}_i \wedge \mathsf{A}_j & \Longleftrightarrow & A_i \cap A_j \\
\mathsf{A}_i \vee \mathsf{A}_j & \Longleftrightarrow & A_i \cup A_j \\
\mathsf{A}_i \Rightarrow \mathsf{A}_j & \Longleftrightarrow & A_i \subseteq A_j \\
\mathsf{A}_i \Leftrightarrow \mathsf{A}_j & \Longleftrightarrow & A_i = A_j \quad .
\end{array}$$

Using this correspondence, we can define evidential reasoning operations in terms of propositional logic, rather than set theory. So doing makes it clear that any implementation will suffice for the propositional logic, so long as it is consistent with the rules of logic. In the following sections we give the logical definitions for several of the evidential reasoning operations and some comments about their implementation.

5.1.1 Interpretation

The interpretation operation calculates the truthfulness of a given statement based upon a given body of evidence. It produces an estimate of both the positive and negative effects of the evidence on the truthfulness of the statement. To interpret a body of evidence m_{A} relative to the statement A_j, we calculate its support and plausibility to derive its evidential interval as follows:

$$\begin{array}{rcl}
Spt(\mathsf{A}_j) & = & \sum\limits_{\mathsf{A}_i \Rightarrow \mathsf{A}_j} m_{\mathsf{A}}(\mathsf{A}_i) \\
Pls(\mathsf{A}_j) & = & 1 - Spt(\neg\mathsf{A}_j) \\
{[Spt(\mathsf{A}_j), Pls(\mathsf{A}_j)]} & \subseteq & [0,1] \quad .
\end{array}$$

Given a tabular representation of a mass distribution m_{A}, the evidential interval for a selected proposition A_i can be calculated in one pass over m_{A}. For each pair in m_{A}, determine if the proposition in the pair implies A_i; if it does, add the mass from the pair to the support for A_i; if it does not, then determine if the proposition in the pair implies $\neg\mathsf{A}_i$; if it does, add the mass from the pair to the support for $\neg\mathsf{A}_i$; otherwise, ignore that pair. Upon completion of the pass down m_{A}, the accumulated value for the support of A_i is the lower bound of the evidential interval, and the difference between 1 and the accumulated value for $\neg\mathsf{A}_i$ is the upper bound.

5.1.2 Fusion

The fusion operation pools multiple bodies of evidence into a single body of evidence that emphasizes points of agreement and deemphasizes points of disagreement. When two mass distributions m^1_{A} and m^2_{A} representing independent opinions are expressed relative to the same frame of discernment, they can be fused using Dempster's Rule of Combination. Dempster's rule pools mass distributions to produce a new mass distribution, m^3_{A}, that represents the consensus of the original disparate opinions. Dempster's rule is defined as follows:

$$\begin{aligned} m^3_{\mathsf{A}}(\mathsf{A}_i \wedge \mathsf{A}_j) &= m^1_{\mathsf{A}} \oplus m^2_{\mathsf{A}}(\mathsf{A}_i \wedge \mathsf{A}_j) \\ &= \frac{1}{1-\kappa} \sum_{\mathsf{A}_i \wedge \mathsf{A}_j} m^1_{\mathsf{A}}(\mathsf{A}_i) m^2_{\mathsf{A}}(\mathsf{A}_j) \\ \kappa &= \sum_{\neg(\mathsf{A}_i \wedge \mathsf{A}_j)} m^1_{\mathsf{A}}(\mathsf{A}_i) m^2_{\mathsf{A}}(\mathsf{A}_j) \\ &< 1 \ . \end{aligned}$$

Assuming that both m^1_{A} and m^2_{A} are tabularly represented, then fusion can be implemented by an algorithm that maps down m^2_{A} once for each pair in m^1_{A}. By so doing, we consider every pair from m^1_{A} in combination with every pair from m^2_{A}. For each such combination, we first determine if the conjunction of the propositions from the pairs is FALSE. If it is, we add the product of the numeric portions of the pairs to the conflict accumulator (i.e., κ); if it is not, we add a new pair, consisting of the conjunction of the propositions and the product of their associated masses, to the tabular representation of the result m^3_{A}. Once all combinations have been considered, we need to normalize the result. This is accomplished by multiplying the numeric portion of each pair in the result by $\frac{1}{1-\kappa}$. Finally, we scan the tabular result looking for pairs with equivalent propositions. Any such pairs can be replaced with a single pair consisting of that proposition and the sum of the masses from those pairs. If this is not done, then the size of the tabular representation for the mass distribution can become large, increasing the computational cost of applying evidential operations to the result.

5.1.3 Discounting

The discounting operation adjusts a body of evidence m_{A} to reflect the credibility of its source, expressed as a discount rate $r \in [0, 1]$. If a source is completely reliable ($r = 0$), discounting has no effect. If a source is completely unreliable ($r = 1$), discounting strips away all apparent information content so that all evidential intervals based upon the resulting distribution are equivalent to the unit interval, except that $[1, 1]$ is always associated with the proposition corresponding to the entire frame; otherwise, discounting lowers the apparent information content in proportion to the source's unreliability so that the evidential intervals are proportionally widened:

$$m^{\%}_{\mathsf{A}}(\mathsf{A}_i) = \begin{cases} (1-r)\, m_{\mathsf{A}}(\mathsf{A}_i), & \mathsf{A}_i \neq \mathsf{TRUE} \\ r + (1-r)\, m_{\mathsf{A}}(\mathsf{TRUE}), & \text{otherwise} \end{cases} .$$

If m_{A} is tabularly represented, then it can be discounted by replacing the numeric portion n_i of each pair by $(1-r)n_i$ and adding the pair (TRUE, r) to the tabular representation. If m_{A} already has a pair with a proposition equivalent to TRUE, then the numeric portion of that pair is replaced by $r + (1-r)n_i$, rather than adding a new pair.

5.1.4 Translation and Projection

These operations move a body of evidence away from its original context to a related one, to assess its impact on dependent hypotheses. If a body of evidence is to be interpreted relative to a question expressed over a frame other than the one over which the evidence is expressed, a path of compatibility relations connecting the two frames is required. The mass distribution expressing the body of evidence is then repeatedly translated from frame to frame, via compatibility mappings, until it reaches the ultimate frame of the question. In translating m_{A} via compatibility mapping $\Gamma_{\mathsf{A}\mapsto\mathsf{B}}$, the following computation is applied to derive the translated mass distribution m_{B}:

$$\begin{aligned} m_{\mathsf{B}}(\mathsf{B}_j) &= \frac{1}{1-\kappa} \sum_{\Gamma_{\mathsf{A}\mapsto\mathsf{B}}(\mathsf{A}_i)=\mathsf{B}_j} m_{\mathsf{A}}(\mathsf{A}_i) \\ \kappa &= \sum_{\Gamma_{\mathsf{A}\mapsto\mathsf{B}}(\mathsf{A}_i)=\mathsf{FALSE}} m_{\mathsf{A}}(\mathsf{A}_i) \\ &< 1 \ . \end{aligned}$$

This same method is applied to move mass distributions among frames that represent states of the world at different times i.e., for projecting.

Given a tabularly represented mass distribution m_{A} and a functional representation of the compatibility mapping $\Gamma_{\mathsf{A}\mapsto\mathsf{B}}$, then the proposition in each pair from m_{A} is replaced by the result of applying $\Gamma_{\mathsf{A}\mapsto\mathsf{B}}$ to that proposition, provided the result is not equivalent to FALSE. When the result is equivalent to FALSE, the pair is deleted from m_{A} after the numeric portion of the pair is added to κ. Once each pair has been so processed, the resulting mass distribution is normalized by multiplying each mass assignment by $\frac{1}{1-\kappa}$. Just as in the case of fusion, duplicate pairs for equivalent propositions are coalesced as a final step.

5.1.5 Summarization

Summarization collects all of the extremely small amounts of mass (determined by a threshold $t \in [0,1]$) attributed to propositions, then attributes the sum to the disjunction of those propositions. The resulting mass distribution is slightly less informative than the original in that some evidential intervals based upon this resulting mass distribution will be wider than those based upon the original, but it remains consistent with the original in that the intervals based on the resulting distribution contain those based on the original. Thus:

$$\begin{aligned} m_A^+(\mathsf{A}_i) &= \begin{cases} m_A(\mathsf{A}_i), & \mathsf{A}_i \neq \mathsf{S} \\ s + m_A(\mathsf{S}), & \text{otherwise} \end{cases} \\ \mathsf{S} &= \bigvee_{0<m_A(\mathsf{A}_i)<t} \mathsf{A}_i \\ s &= \sum_{0<m_A(\mathsf{A}_i)<t} m_A(\mathsf{A}_i) \quad . \end{aligned}$$

This can be accomplished with a pass down a tabular representation of the mass distribution, removing those pairs whose numeric portion are below the selected threshold, and then adding a new pair corresponding to the disjunction of the propositions and the summation of the numeric portions of the discarded pairs. In adding this new pair, if the mass distribution already includes a pair with an equivalent proposition, then that pair is replaced by one with the same proposition and the sum of the discarded mass and the mass of the pair being replaced; otherwise, the new pair is simply included as a new pair in the mass distribution.

5.1.6 Gisting

The gist of a mass distribution is the most pointed statement from the frame whose support meets or exceeds a selected level. This definition uses cardinality as a measure of specificity (i.e., pointedness). As a logical operation, cardinality needs to be defined as a measure that is proportional to the number of possible worlds in which the given proposition is true. The gist G of a mass distribution, m_A, is defined relative to a gist level, $g \in [0, 1]$:

$$\begin{aligned} \mathsf{G} &= \bigvee_{\mathsf{A}_i \in G} \mathsf{A}_i \\ G &\subseteq \{\mathsf{A}_i | m_A(\mathsf{A}_i) > 0\} \quad , \end{aligned}$$

for all $\mathsf{A}_i, \mathsf{A}_j \in G, \mathsf{A}_k \notin G$

$$\begin{aligned} Spt(\mathsf{A}_i) = Spt(\mathsf{A}_j) &\geq g \\ |\mathsf{A_i}| &= |\mathsf{A_j}| \\ Spt(\mathsf{A}_i) > Spt(\mathsf{A}_k) \quad &or \quad |\mathsf{A_k}| > |\mathsf{A_i}| \quad . \end{aligned}$$

Gisting require a search through subsets of focal elements in a mass distribution. To speed this search, the algorithm determines the smallest number of focal elements, taken in descending order of mass, whose mass total exceeds the threshold. This becomes the candidate to beat. To beat this candidate, a set of focal elements with equivalent or lower cardinality and greater mass total than the threshold must be found. If a candidate has higher cardinality, then neither it nor any superset of it needs to be considered. Thus, those subsets need not be considered.

5.2 Propositions and Logical Inferences

The preceding discussion on the implementation of evidential reasoning operations assumes an implementation of propositional logic. However, the two implementations are completely independent of one another. In fact, depending upon the characteristics of the domain of application, different logical implementations are best utilized. Since Gister-CL was designed to be suitable for multiple domains of application, it supports the use of different logical reasoning implementations.

Each logical reasoning implementation needs to support the same operations. These are the operations assumed in the numerical calculation, given two propositions A_i and A_j,

- Negation: $\neg\mathsf{A}_i$
- Disjunction: $\mathsf{A}_i \vee \mathsf{A}_j$
- Conjunction: $\mathsf{A}_i \wedge \mathsf{A}_j$
- Implication: $\mathsf{A}_i \Rightarrow \mathsf{A}_j$
- Biimplication: $\mathsf{A}_i \Leftrightarrow \mathsf{A}_j$
- Translation: $\Gamma_{\mathsf{A}\mapsto\mathsf{B}}(\mathsf{A}_i)$
- Cardinality: $||\mathsf{A_i}||$

The implementation of these operations depends upon the selected representation for propositions. Each such representation must include two distinguished propositions, the true proposition, TRUE, and the false proposition, FALSE, in addition to all other propositions that are included in the associated frame of discernment.

There are two propositional forms supported for each representation: the *normal form* and the *display form*. The normal form is the internal representation of a proposition utilized by Gister-CL in support of its evidential reasoning operations; the display form is the external representation utilized when communicating with the user. The user enters propositions utilizing the display form and Gister-CL uses this form for its output to the user. The display forms of propositions in any of the logical representations include elementary units, aliases, and logical combinations expressed in prefix notation. The range of legal forms for a proposition P is defined by the following:

P = E |
A |
(P ... P) |
(NOT P) |
(OR P ... P) |
(AND P ... P)
(IMPLIES P P) .

E = an elementary unit from a frame.

A = UNKNOWN |
an alias from a frame.

5.2.1 Set Logic

Probably the most straightforward implementation of propositional logic for evidential reasoning is based upon set theory. We use linked lists to represent sets; each list is maintained as a sequence of unique symbols without regard to their order. Using this representation, the false proposition is the null list; the true propositions is any list that includes exactly one symbol for each element in the frame of discernment; all other propositions are represented as lists containing symbols for some subset of the elements in the frame.

For example, suppose that we are trying to determine who committed a burglary and we have already determined that the only suspects are Enrique, Janet, John, and Tom. In this case, the false proposition is represented by the null list and the true proposition, corresponding to the frame of discernment, is represented by a list that includes a symbol corresponding to each suspect:

$$\mathrm{SUSPECTS} = (\mathrm{ENRIQUE}, \mathrm{JANET}, \mathrm{JOHN}, \mathrm{TOM}) \quad .$$

If the evidence to be considered includes the sex and predominant hand of the burglar, then the following additional sets might be defined as lists:

$$\begin{aligned} \mathrm{MALE} &= (\mathrm{ENRIQUE}, \mathrm{JOHN}, \mathrm{TOM}) \\ \mathrm{FEMALE} &= (\mathrm{JANET}) \\ \\ \mathrm{RIGHT\text{-}HANDED} &= (\mathrm{ENRIQUE}, \mathrm{TOM}) \\ \mathrm{LEFT\text{-}HANDED} &= (\mathrm{JANET}, \mathrm{JOHN}) \quad . \end{aligned}$$

Using this representation, it is a fairly simple task to define negation as set complementation, conjunction as set intersection, implication as set inclusion, disjunction as set union, and biimplication as set equality.

Another frame might be used to capture the possible blood types of the burglar. If the types consist of a blood group (i.e., A, B, AB, O) coupled with an Rh factor (i.e., +, -), then this frame is captured by the following lists:

$$\begin{aligned}
\text{BLOOD-TYPES} &= (\text{A+}, \text{A-}, \text{B+}, \text{B-}, \text{AB+}, \text{AB-}, \text{O+}, \text{O-}) \\[1em]
\text{A} &= (\text{A+}, \text{A-}) \\
\text{B} &= (\text{B+}, \text{B-}) \\
\text{AB} &= (\text{AB+}, \text{AB-}) \\
\text{O} &= (\text{O+}, \text{O-}) \\[1em]
\text{Rh+} &= (\text{A+}, \text{B+}, \text{AB+}, \text{O+}) \\
\text{Rh-} &= (\text{A-}, \text{B-}, \text{AB-}, \text{O-}) \quad .
\end{aligned}$$

If we want to use evidence about the burglar's blood type in combination with the other evidence, then we need to define the compatibility mappings between the two frames. This can be done in terms of a compatibility relation represented by an association list. Here the association list consists of an unordered list of pairs, where the first element of each pair is an element from SUSPECTS and the second element is an element from BLOOD-TYPES. A pair is included just in case the blood type is a possibility for that suspect. If the available information is that Enrique's blood type is B+, Janets's blood group is A (i.e., her Rh factor is unknown), Johns's blood group is B, and nothing is known about Tom's blood type, then the following association list might be used to represent the SUSPECTS-BLOOD-TYPES compatibility relation.

$$\begin{aligned}
\text{SUSPECTS-BLOOD-TYPES} = (\ &(\text{ENRIQUE}, \text{B+}) \\
&(\text{JANET}, \text{A+}), (\text{JANET}, \text{A-}), \\
&(\text{JOHN}, \text{B+}), (\text{JOHN}, \text{B-}), \\
&(\text{TOM}, \text{A+}), (\text{TOM}, \text{A-}), (\text{TOM}, \text{B+}), (\text{TOM}, \text{B-}), \\
&(\text{TOM}, \text{AB+}), (\text{TOM}, \text{AB-}), (\text{TOM}, \text{O+}), (\text{TOM}, \text{O-})\ ) \quad .
\end{aligned}$$

Given this representation of the compatibility relation, translation is easily implemented. A set over the BLOOD-TYPES frame is translated to the SUSPECTS frame by matching each element in the set against the second element of each pair in the association list; for each match, if the first element of the pair is not already in the list to be returned, it is added to the list. When all of the elements in the set have been considered in combination with each pair from the compatibility relation, the list is returned. To move information from the SUSPECTS frame to the BLOOD-TYPES frame, the same procedure is invoked, but the roles of the elements in the pairs are interchanged; they match against the first element and return the second.

Finally, to ease user interaction with this implementation, it is useful to add a normalization operation that will allow the user to enter aliases for more complex expressions. For example, an association list might be maintained for each frame that pairs aliases with the set that they represent. Thus, for the SUSPECTS frame, aliases might be so defined for MALE, FEMALE, RIGHT-HANDED, LEFT-HANDED, and other useful shorthands. The normalization operation simply substitutes the associated full expressions for each alias in the user's inputs, before doing any further processing.

Within Gister-CL this SET representation is straightforwardly implemented by storing the ELEMENTS list of frame elements and the ALIASES association list on each frame node in the gallery, and the ELEMENTS association list on each compatibility arc in the gallery. The advantage of this representation is its simplicity. However, as the number of elements in the propositions linearly increases, the computational costs of the operations (e.g., union, intersection) exponentially increase. Therefore, this representation is practicality limited to those domains where the number of elements in the frames are small.

5.2.2 Vector Logics

An alternative representation uses binary vectors to represent sets. The advantage of this representation is that the computational costs of the logical operations for a given frame are fixed; they do not depend upon the particular propositions, and on average they are substantially less than the cost for the unordered linked list representation.

Here one chooses an arbitrary linear ordering for the elements in a frame. Propositions are represented by vectors of zeros and ones. The dimension of these vectors is determined by the number of elements in the frame. A one appears as the i-th element of a vector, representing a proposition, if the i-th element, from the linearly ordered elements of the frame, is a member of the set corresponding to that proposition; otherwise, the i-th element is a zero. Therefore, the true proposition is a vector of all ones and the false proposition is a vector of all zeros. Using this representation for the SUSPECTS frame, assuming the linear order to be

ENRIQUE, JANET, JOHN, TOM,

we have

$$\begin{aligned}
\text{ENRIQUE} &= [1,0,0,0] \\
\text{JANET} &= [0,1,0,0] \\
\text{JOHN} &= [0,0,1,0] \\
\text{TOM} &= [0,0,0,1] \\
\\
\text{MALE} &= [1,0,1,1] \\
\text{FEMALE} &= [0,1,0,0] \\
\\
\text{RIGHT-HANDED} &= [1,0,0,1] \\
\text{LEFT-HANDED} &= [0,1,1,0] \quad .
\end{aligned}$$

Similarly, if the linear order over BLOOD-TYPES is

A+, A-, B+, B-, AB+, AB-, O+, O-,

then the frame is represented by the following:

$$\begin{aligned}
\text{A+} &= [1,0,0,0,0,0,0,0] \\
\text{A-} &= [0,1,0,0,0,0,0,0] \\
\text{B+} &= [0,0,1,0,0,0,0,0] \\
\text{B-} &= [0,0,0,1,0,0,0,0] \\
\text{AB+} &= [0,0,0,0,1,0,0,0] \\
\text{AB-} &= [0,0,0,0,0,1,0,0] \\
\text{O+} &= [0,0,0,0,0,0,1,0] \\
\text{O-} &= [0,0,0,0,0,0,0,1] \\
\\
\text{A} &= [1,1,0,0,0,0,0,0] \\
\text{B} &= [0,0,1,1,0,0,0,0] \\
\text{AB} &= [0,0,0,0,1,1,0,0] \\
\text{O} &= [0,0,0,0,0,0,1,1] \\
\\
\text{Rh+} &= [1,0,1,0,1,0,1,0] \\
\text{Rh-} &= [0,1,0,1,0,1,0,1] \quad .
\end{aligned}$$

Each of the logical operations requires a single pass down the vectors: negation inverts the vector, changing ones to zeros and zeros to ones; conjunction returns a one where both vectors have ones and zeros elsewhere; one vector implies a second if the second vector has a one wherever the first has a one; a vector representing the disjunction of two vectors has a one wherever either of the two have a one; if two vectors are identical, then they imply each other. Also, it is useful to define a normalization operation, based upon an association list of aliases paired with their vector representations, to ease user interaction.

Again, we can define the compatibility mapping in terms of a compatibility relation. If the two frames to be related have dimensions n and m, then the compatibility relation is represented by an n by m matrix. Position i, j in this matrix corresponds to relationship between the i-th element of the n-dimensional frame and the j-th element of the m-dimensional frame; if a zero is at that position, the elements are incompatible; if a one is at that position, the elements are compatible. In the case of the SUSPECTS-BLOOD-TYPES compatibility relation, we have

$$\text{SUSPECTS-BLOOD-TYPES} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad .$$

Given a vector defined over the SUSPECTS frame, treat it as a column vector, find the positions with ones in them, and select the corresponding rows from SUSPECTS-BLOOD-TYPES. Taking the disjunction of these rows produces the translation of the SUSPECTS vector to the BLOOD-TYPES frame. Similarly, a vector over the BLOOD-TYPES frame can be translated to the SUSPECTS frame by taking the disjunction of the columns from SUSPECTS-BLOOD-TYPES that have corresponding ones in the BLOOD-TYPE vector, and converting the resulting column vector to a row vector.

The vectors and matrices in this vector representation can be implemented utilizing any of a number of data structures. These include binary arrays, binary linked lists, and integers interpreted by their binary representation. Correspondingly, Gister-CL supports three vector-like representations: BVECTOR, BNUM, and BLIST. Each of these representations stores, at each frame in the gallery, the selected linear ordering for that frame's elements as a linked list of symbols under ELEMENTS and an association list of aliases under ALIASES. BVECTOR represent the compatibility matrices as two-dimensional binary arrays stored under ELEMENTS; BNUM represents them as a linked list of integers, one integer corresponding to each row of the matrix, stored under ELEMENTS; BLIST uses a linked list of linked binary lists corresponding to a list of the matrix rows, also stored under ELEMENTS. The best choice depends upon the characteristics of the domain of application and the associated memory management overhead.

5.2.3 Graph Logic

In implementing evidential reasoning, we have found that frames, as with the other formal elements in this theory, can be represented straightforwardly as graphs consisting of nodes connected by directed edges. Because they are graphs, these formal elements are easily understood, and they provide an intuitive basis for man-machine interaction. To support graphs as a data type in COMMON LISP, we developed Grasper-CL. Grasper-CL implements graphs as hash tables indexed by node names. List structures, associated with each node, represent the connecting arcs and association lists of values for each node.

In the GRAPH representation, a frame is represented by a named subgraph (i.e., space) that includes a node for each element of the frame and may include additional nodes representing aliases. Each of these additional nodes has edges pointing to elements of the frame (or other aliases) that make up that alias. Figures 5.1 and 5.2 are the GRAPH representations for the SUSPECTS and BLOOD-TYPES frames. The required logical operations are implemented in the same manner as for the set logic representation, except that the true proposition and the normalized aliases are derived from the graph structure rather than from an association list.

A compatibility relation is represented as a subgraph including the nodes from the frames it relates, with edges connecting compatible elements. For example, in the SUSPECTS-BLOOD-TYPES compatibility relation (Figure 5.3), ENRIQUE is connected to B+ indicating that his blood type is known to be B+, while JOHN is connected to B+ and B- indicating that his blood group is know to be B, but his Rh factor is unknown. Propositions, represented by sets, are translated one element at a time and then unioned together

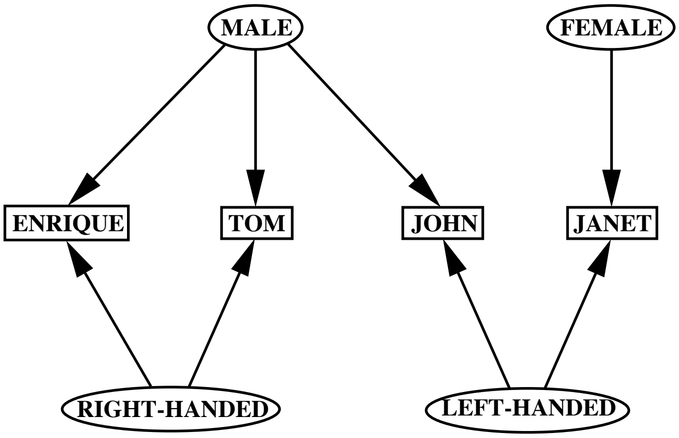


Figure 5.1: SUSPECTS frame of discernment

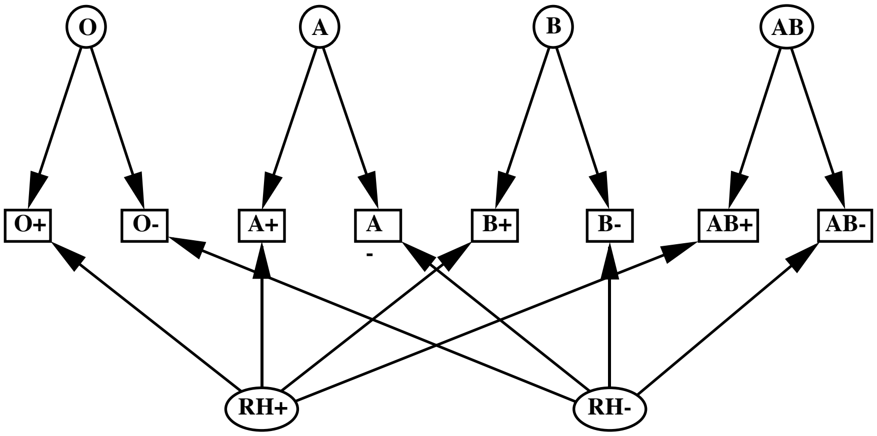


Figure 5.2: BLOOD-TYPES frame

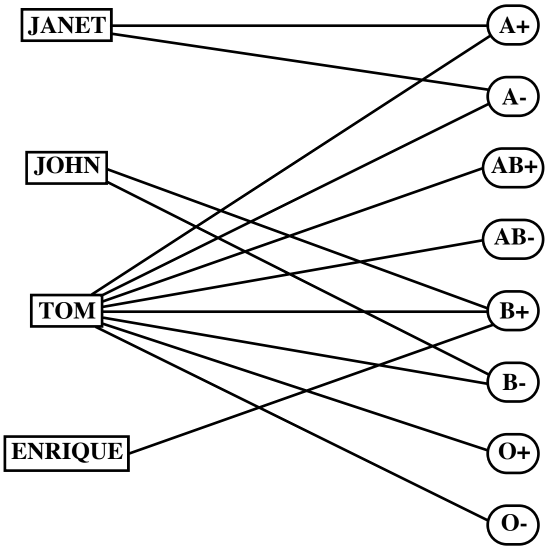


Figure 5.3: SUSPECTS-BLOOD-TYPES compatibility relation

to get the overall result. An element is translated by simply referencing the graphical compatibility relation and returning all of the nodes connected to that element.

To ease the transition from one logical representation to another, the Curator includes a cross-compiler. Utilizing this facility, a frame or relation in the GRAPH representation can be automatically converted to another discrete representation i.e., BLIST, BNUM, BVECTOR, SET. This allows the user to make use of the GRAPH representation for ease of interaction and another for performance.

5.2.4 Continuous Logic

Some questions of interest are better expressed relative to a continuous numeric scale rather than a discrete set of possible answers. For example, the height of a person might be best expressed relative to a continuous scale. To support this type of logical reasoning, Gister-CL supports the CONTINUOUS representation. Each frame utilizing this representation has a range of ELEMENTS that consists of a pair of numbers representing the lower and upper bounds (inclusive) of the scale (e.g., (0, 90) corresponds to the closed interval [0, 90]). Propositional statements representing single elements from this frame correspond to single numbers within this range (e.g., 72); propositions corresponding to (open) subintervals of this range are each represented by a pair of numbers, the lower and upper bounds of the subinterval (e.g., (65 72)); other propositional statements correspond to ordered lists of numbers and intervals and are formed by taking logical combinations of the points and open intervals. For example, the statement (OR (65 72) 72), in display form, corresponds to (65, 72], the interval open at 65 and closed at 72, and is represented, in normal form, by the list ((65 72) 72); (AND (55 70) (OR 60 (65 72) 72)) corresponds to the disjunction of the element 60 and the open interval from 65 to 70, and is represented by the list (60 (65 70)). Similarly, the list (45 (48 49) 50 (55 60)) represents the disjunction of the points

45 and 50 and the open intervals (48 49) and (55 60). Just like other representations, CONTINUOUS frames can include an ALIASES association list, where alias symbols are paired with propositional statements from the frame.

CONTINUOUS representations for compatibility relations capture the relationships among CONTINUOUS frames in terms of two functions. Each of these functions takes a single numeric element from one of these frames as its input and returns the compatible portion of the other frame's numeric scale as its output. A pair of functions is stored as the ELEMENTS of each such compatibility relation; one mapping in one direction along the relation and the other mapping in the opposite direction. For example, the pair of functions might consist of one that adds 1 to its argument (i.e., the COMMON LISP function 1+) and another that subtracts one (i.e., the COMMON LISP function 1-). When translating or projecting a proposition across a CONTINUOUS compatibility relation, the appropriate function is applied to each numeric value in the proposition. Thus, applying 1+ to the proposition (45 (48 49) 50 (55 60)) results in (46 (49 50) 51 (56 61)), and applying 1- to this result returns us to the original proposition. As a consequence of this method of application, the relationships represented by CONTINUOUS compatibility relations must be monotonic i.e., the projection of an interval must correspond to the range bounded by the projection of its extremes; nonmonotonic relationships can be represented utilizing the FUNCTIONAL representation described below. The functions associated with CONTINUOUS compatibility relations are not limited to returning point values. For example, the application of a function that returns an open interval, two units wide, centered on the value given to the function, when applied to (45 (48 49) 50 (55 60)) would return ((44 46) (47 51) (54 61)).

The implementation of the logical operations for this representation are more complex than those discussed earlier. Both disjunction and conjunction walk down the given propositions from beginning to end, taking advantage of their ordered representation. The front elements of the propositions are tested for overlap; the appropriate result for those elements are calculated doing the continuous equivalent of union or intersection; the result is conjoined with the appropriate recursive call. Negation is computed by taking the disjunction of the compliment of each element in the proposition relative to the range of the frame. Implications are computed as material implication using disjunction and negation. Cardinality is the sum of cardinality of a proposition's elements using the width of intervals and a fixed (very small) value for points[3]. Translation and projection are calculated by mapping down the given proposition, applying the user supplied function to points, applying the user supplied function to the lower and upper bounds of intervals and forming a new interval consisting of the lower and upper bound of the results (or a single point if the lower and upper bounds are equal), and taking the disjunction of the results.

5.2.5 Functional Logic

The FUNCTIONAL logic is intended to allow the user to specify a logic for frames and relations that performs in user defined ways. The contents of a frame utilizing FUNCTIONAL logic has a slot for each operation that needs to be supported. The user simply inserts their functions into these slots. Gister-CL will utilize these user supplied functions

---

[3] This is stored in the global variable *SINGLE-ELEMENT-CARDINALITY.CONTINUOUS

when performing the corresponding operations on such frames; if no user supplied function is provided in a slot then Gister-CL uses the SET logic definition for that operation.

FUNCTIONALly represented compatibility relations capture the relationships among frames in much the same way as with CONTINUOUS compatibility relations. A pair of functions determines how propositions are translated or projected, one for each direction. FUNCTIONAL compatibility relations can be placed between FUNCTIONAL frames or between frames with other logical representations. In that case, there is an additional slot on the compatibility relation that specifies the sequence of representations to be used during translation: the representation of the initial frame in which the proposition to be translated is expressed, FUNCTIONAL indicating that the FUNCTIONAL representation of the compatibility relation is to be used during translation, and a final representation of the target frame in which the result is to be represented. Thus, FUNCTIONAL compatibility relations can be used to connect arbitrary frames[4] [5].

5.2.6 Hybrid Logic

Gister-CL's HYBRID logic provides a means for the user to define a frame as a linear sequence of other frames; each frame in the sequence is interpreted as being independent from the others. Propositions are represented as linked lists of (sub)propositions from the component frames, each (sub)proposition being captured in the representation dictated by its corresponding frame in the gallery. HYBRID frames are generated through the Curator's "Macro Create Frames or Relations" option under CREATE' in the FRAME&REL menu. They are linked to other frames through FUNCTIONal compatibility relations.

5.2.7 Utilizing Multiple Logical Representations

Since we anticipated that different problems would best be solved using different implementations of logical inference, every mass distribution in the Analyzer is accompanied by a referenced frame of discernment. When a logical question needs to be answered, this frame of discernment along with the question and compatibility relation, when appropriate, is passed to the Curator. Thus, the Curator knows which frame and compatibility relation is to be used in generating the response.

Within the gallery, each frame and compatibility relation references the logical representation that has been used to implement it. When questions are posed by the Analyzer, the Curator calls the procedures corresponding to the referenced representation. The actual implementation uses object-oriented programming to route the questions to the appropriate

---

[4] Note that the functions stored for FUNCTIONAL compatibility relations are defined over propositions (i.e., not restricted to elements) and, therefore, can be used to capture nonmonitonic relationships among CONTINUOUSly represented frames.

[5] Two pairs of functions are supported by Gister-CL for creating FUNCTIONAL compatibility relations between graph and continuous frames, and between graph and functional frames. TRANSLATE.GRAPH-TO-CONTINUOUS and TRANSLATE.CONTINUOUS-TO-GRAPH is one pair and TRANSLATE.GRAPH-TO-FUNCTIONAL and TRANSLATE.FUNCTIONAL-TO-GRAPH is the other. Both can make use of RELATION-CONTENTS-MODE.XREP-YREP as the RELATION-CONTENTS-MODE operation.

procedures. Thus, adding new representations to the Curator involves adding new objects and methods to the system. As a result, new logical representations are easily added without otherwise modifying the Curator or Analyzer.

5.2.8 Fuzzy Logic

Given Gister-CL's object-oriented implementation, different forms of possibilistic reasoning can be incorporated within its galleries. In particular, we have experimented substituting fuzzy logics for Boolean-valued logics i.e., fuzzy versions of the GRAPH and SET logics, called FGRAPH and FSET. Each of these is just like its Boolean-valued counterpart only the propositional statements have a fuzzy membership number, from $[0.0, 1.0]$, paired with them. Similarly, each element of a frame or compatibility relation has a fuzzy membership number. Fuzzy methods for all of the logical operations are straightforwardly substituted for the Boolean-valued ones.

5.3 The Analyzer

The procedures in this section do not have any direct impact on the Gister-CL display. The programmer must use Grasper-CL's DRAWing and ERASEing procedures to keep the display synchronized with Gister-CL's data structures.

Delta-f and delta-t describe changes in fods. If they are atomic, delta-f is the target frame for translation and delta-t is the time shift (a positive or negative integer) for projection. Otherwise, these atomic arguments are paired with the compatibility relations to be used in translating or projecting. When no compatibility relations are given, all compatibility relations from the gallery that can affect the change are utilized.

5.3.1 Creation Procedures

These operations create and augment analyses. Analyses are augmented by creating new evidential relationships. Creating a relationship typically causes nodes, node values, edges, and edge values to be added to a Grasper-CL space that represents an analysis. Relationships are created without checking the validity of the arguments; it is the responsibility of the programmer to guarantee their validity. The numeric parameters, discount-rate, gist-level, and summary-minimum, range from 0 to 100. Creation includes evaluation so long as the global switch *AUTO-UPDATE? is on.

- CREATE-ANALYSIS-SPACE [analysis gallery]
- CREATE-CONDITIONING-RELATIONSHIP [conditioning-name frame time statement user-mass-distribution analysis *&OPTIONAL location*]
- CREATE-CONVERSION-RELATIONSHIP [evidence-node delta-fod analysis *&OPTIONAL location*]

- CREATE-DISCOUNT-RELATIONSHIP [evidence-node discount-rate analysis *&OPTIONAL location*]
- CREATE-DISTRIBUTION-RELATIONSHIP [distribution-name user-mass-distribution frame time analysis *&OPTIONAL location*]
- CREATE-FACT-RELATIONSHIP [fact-name frame time statement analysis *&OPTIONAL location*]
- CREATE-FUSION-RELATIONSHIP [evidence-nodes analysis *&OPTIONAL location*]
- CREATE-GISTING-RELATIONSHIP [evidence-nodes analysis *&OPTIONAL gisting-name gist-level location gist-type*]
- CREATE-INTERPRETATION-RELATIONSHIP [interpretation-name evidence-node propositions analysis *&OPTIONAL location*]
- CREATE-PROJECTION-RELATIONSHIP [evidence-node delta-t analysis *&OPTIONAL location*]
- CREATE-REPORT-RELATIONSHIP [report-name frame time statement credibility analysis *&OPTIONAL location*]
- CREATE-SUMMARY-RELATIONSHIP [evidence-node summary-minimum analysis *&OPTIONAL location*]
- CREATE-TRANSLATION-RELATIONSHIP [evidence-node delta-f analysis *&OPTIONAL location*]

5.3.2 Evaluation Procedures

These operations cause evidential conclusions to be recalculated. The results are stored in the values of the appropriate evidential relationship nodes. The global switch *AUTO-UPDATE? inhibits mass distribution recalculation when it is off. *BIAS is used as the bias for interpretation evaluation.

- EVALUATE-ANALYSIS-SPACE [analysis]
- EVALUATE-CONDITIONING-RELATIONSHIP [conditioning-node analysis]
- EVALUATE-CONVERSION-RELATIONSHIP [conversion-node analysis]
- EVALUATE-DISCOUNT-RELATIONSHIP [discount-node analysis]
- EVALUATE-DISTRIBUTION-RELATIONSHIP [evidence-node analysis]
- EVALUATE-FACT-RELATIONSHIP [fact-node analysis]
- EVALUATE-FUSION-RELATIONSHIP [fusion-node analysis]

- EVALUATE-GISTING-RELATIONSHIP [gisting-node analysis]
- EVALUATE-INTERPRETATION-RELATIONSHIP [interpretation-node analysis]
- EVALUATE-NODE [node analysis]
- EVALUATE-NODE-AND-DESCENDANTS [node analysis]
- EVALUATE-PROJECTION-RELATIONSHIP [projection-node analysis]
- EVALUATE-REPORT-RELATIONSHIP [report-node analysis]
- EVALUATE-SUMMARY-RELATIONSHIP [summary-node analysis]
- EVALUATE-TRANSLATION-RELATIONSHIP [translation-node analysis]

5.3.3 Predicate Procedures

These operations return their first argument if the argument's type is the one being tested.

- ANALYSIS-SPACE? [space]
- CONDITIONING-NODE? [node space]
- CONVERSION-NODE? [node space]
- DISCOUNT-NODE? [node space]
- DISTRIBUTION-NODE? [node space]
- EVIDENCE-NODE? [node space]
- FACT-NODE? [node space]
- FUSION-NODE? [node space]
- GISTING-NODE? node [space]
- INTERPRETATION-NODE? [node space]
- PROJECTION-NODE? [node space]
- REPORT-NODE? [node space]
- SUMMARY-NODE? [node space]
- TRANSLATION-NODE? [node space]

5.3.4 Search Procedures

These procedures return lists of evidence nodes or analysis spaces.

- ANCESTOR-EVIDENCE [evidence-node analysis]
- DESCENDANT-EVIDENCE [evidence-node analysis]
- SET-OF-ANALYSES []

5.3.5 Evidential Procedures

These procedures support the basic evidential-reasoning operations. They all require the propositions included in the massfuns to be in normal form. Those that return massfuns are sensitive to the global switch *NORMALIZE? that inhibits the renormalization of the mass distributions when it is off. *FUZZ specifies the greatest difference there can be between two numbers that are considered to be equal by these operations. As a side effect of the two Dempster's rule procedures and the projection, translation, and normalization procedures, the global variables *CONFLICT and *HARMONY are set to the total mass attributed to the false proposition and the total mass attributed to other propositions before normalization.

Massfuns consist of a list of pairs where each pair consists of a proposition and mass assignment ranging from 0 to 1.0. All other numeric parameters (i.e., discount-rate, gist-level, and minimum-mass) must fall within this same range. The only exception is the time increment included in delta-t that must be a positive or negative integer.

- BINARY-DEMPSTER [massfun1 massfun2 fod]
- CONSONANCE [massfun fod]
- CONVERT [massfun new-fod fod]
- DISCOUNT [massfun discount-rate fod]
- GIST [massfun gist-level fod]
- INTERPRET [massfun propositions bias fod]
- NARY-DEMPSTER [massfuns fod]
- NORMALIZE [massfun fod]
- PROJECT [massfun delta-t fod]
- QUICK-GIST [massfun gist-level fod]
- SPECIFICITY [massfun fod]
- SUMMARIZE [massfun minimum-mass fod]
- TRANSLATE [massfun delta-f fod]

5.4 The Curator

The procedures in this section do not have any direct impact on the Gister-CL display. The programmer must use Grasper-CL's DRAWing and ERASEing procedures to keep the display synchronized with Gister-CL's data structures.

5.4.1 Creation Procedures

These procedures create and augment galleries with new components. Creating a new component in a gallery typically causes nodes, node values, edges, edge values, spaces, and space values to be added to the Grasper-CL graph that contains the gallery. These components are created without checking the validity of the arguments; it is the responsibility of the programmer to guarantee their validity.

- COPY-FRAME [frame new-name gallery location]
- COPY-GALLERY [gallery new-gallery-name]
- CREATE-ALIAS-RELATIONSHIP [name elements-and-aliases frame gallery location]
- CREATE-ELEMENT [name frame gallery location *&OPTIONAL add-all-edges?*]
- CREATE-FRAME [name gallery location *&OPTIONAL representation*]
- CREATE-GALLERY [name]
- CREATE-RELATION [frame1 relation-name frame2 gallery *&OPTIONAL add-all-edges? representation projection?*]
- CREATE-ALIASES-FROM-NEIGHBORING-FRAMES [frame gallery]
- CREATE-ALIASES-FROM-RELATION [target-frame relation gallery *&OPTIONAL source-frame fod new-aliases*]
- CREATE-RELATIONSHIP-EDGE [element1 element2 relation gallery]

5.4.2 Modification Procedures

These procedures remove entities from galleries, backup and revert galleries, and rename gallery entities.

- BACKUP-GALLERY [gallery]
- DESTROY-ALIAS [alias frame gallery]
- DESTROY-ELEMENT [element frame gallery]

- DESTROY-FRAME-AND-RELATIONS [frame gallery]
- DESTROY-GALLERY [gallery]
- DESTROY-RELATION [frame1 relation frame2 gallery]
- RENAME-ALIAS [alias new-name frame gallery]
- RENAME-ELEMENT [element new-name frame gallery]
- RENAME-FRAME [frame new-name gallery]
- RENAME-GALLERY [gallery new-name *&OPTIONAL rename-in-analyses?*]
- RENAME-RELATION [frame1 relation frame2 gallery]
- REVERT-GALLERY [gallery]

5.4.3 Search Procedures

These procedures return sets of entities from galleries.

- FRAMES-OF-RELATION [relation gallery]
- SET-OF-ALIASES [frame gallery]
- SET-OF-ALIAS-ELEMENTS [aliases frame gallery]
- SET-OF-ELEMENTS [frame gallery]
- SET-OF-FRAMES [gallery]
- SET-OF-GALLERIES []
- SET-OF-PROJECTIONS [gallery]
- SET-OF-RELATIONS [gallery]
- SET-OF-TRANSLATIONS [gallery]

5.4.4 Predicate Procedures

These operations return their first arguments if the argument's type is the one being tested.

- FRAME? [possible-frame gallery]
- GALLERY? [possible-gallery]
- PROJECTION-RELATION? [possible-projection gallery]
- RELATION? [possible-relation gallery]
- TRANSLATION-RELATION? [possible-translation gallery]

5.4.5 Logic Procedures

All of the logical questions that are posed in support of the evidential-reasoning operations are resolved by these procedures. They each reference the *fod* (i.e., frame of discernment) relative to which the question is to be answered. One procedure puts propositions in their normal form; another returns the given proposition's cardinality. Still others return the set of aliases and set of elements for a given fod.

- ALIAS-PROPS [fod]
- AND-PROPS [proposition1 proposition2 fod]
- CARDINALITY-PROP [proposition fod]
- CLEAR-FRAME [frame] [frame gallery representation-name]
- CLEAR-GALLERY [gallery representation-name]
- CLEAR-RELATION [relation gallery representation-name]
- COMPILE-FRAME [frame gallery representation-name]
- COMPILE-GALLERY [gallery representation-name]
- COMPILE-RELATION [relation gallery representation-name]
- CONVERT-PROP [proposition new-fod fod]
- DISPLAY-FORM-PROP [proposition fod]
- ELEMENT-PROPS [fod]
- ELEMENTS-PROP [prop fod]
- EQUAL-PROPS [proposition1 proposition2 fod]
- FALSE-PROP [fod]
- FALSE-PROP? [proposition fod]
- NORMAL-FORM-PROP [proposition fod]
- NOT-PROP [proposition fod]
- OR-PROPS [proposition1 proposition2 fod]
- PROJECT-PROP [proposition delta-t fod]
- TRANSLATE-PROP [proposition delta-f fod]
- TRUE-PROP [fod]
- TRUE-PROP? [proposition fod]

5.4.6 FOD Procedures

These procedures create fod's (i.e., frames of discernment), each consisting of a frame from a gallery and an associated time, and return their individual components.

- CREATE-FOD [frame time *&OPTIONAL gallery*]
- COMPATIBLE-FOD? [fod1 fod2]
- CONVERT-FOD [fod delta-fod]
- EQUAL-FODS? [fod1 fod2]
- FOD-GALLERY [fod]
- FOD-FRAME [fod]
- FOD-REPRESENTATION [fod]
- FOD-REPRESENTATION-NAME [fod]
- FOD-TIME [fod]
- PROJECT-FOD [fod delta-t]
- SET-OF-REPRESENTATION-NAMES []
- TRANSLATE-FOD [fod delta-f]

Chapter 6

Summary

We have developed a system, Gister-CL, that supports the construction, modification, and analysis of evidential arguments. Gister-CL supports an interactive, menu-driven, graphical interface that allows these structures to be easily understood and manipulated. The user simply selects from a menu to add an evidential operation to an analysis, to modify operation parameters, or to change any portion of a gallery. In response, Gister-CL updates the analyses, allowing the user to explore quickly the space of alternative arguments. Gister-CL also supports a programmer's interface that allows the sophisticated user to control all aspects of argument construction under program control.

Appendix A

Bayesian Network Example

To illustrate how evidential analyses can mimic Bayesian networks, we have implemented a running example taken from Pearl's book on probabilistic reasoning [Pea88]. The example concerns a burglar alarm, testimony about it sounding, and questions related to a possible burglary. Although the example is only paraphrased in the following discussion, the probabilities are drawn directly from Pearl's example and the probabilistic conclusions produced by the evidential analysis are identical to those of Pearl.

Pearl's example begins on page 35 of his book by hypothesizing that a home burglar alarm is sounding. He then examines the likelihood that a burglary has taken place. To support this analysis he provides the following additional information and assumptions.

- Based upon previous crime patterns, assume that there is a one in ten thousand chance that a given house will be burglarized on a given night.
- Assume that there is a 95% chance that an attempted burglary will trigger the alarm system.
- Based on previous false alarms, assume there is a slight (1%) chance that the alarm will be triggered by a mechanism other than an attempted burglary.

To capture this information we create a gallery with three frames. The ALARM frame has two elements, ALARM and NO ALARM; the BURGLARY frame has two elements, BURGLARY and NO BURGLARY; the joint frame, BURGLARY&ALARM, has four elements corresponding to the cross product of the elements in the other two frames. Compatibility relations connect ALARM to BURGLARY&ALARM and BURGLARY to BURGLARY&ALARM, in the obvious way. Next we create an analysis with four bodies of evidence. The first, named P(B) and defined over the BURGLARY frame, attributes 0.01% of its mass to BURGLARY and 99.99% to NO BURGLARY; the second, P(.|B) defined over the BURGLARY&ALARM frame, attributes 95% of its mass to ALARM|BURGLARY and 5% to NO ALARM|BURGLARY using conditional embedding (see Section 3.4.1); the third, P(.|¬B) defined over BURGLARY&ALARM, attributes 1% to ALARM|NO BURGLARY and 99% to NO ALARM|NO BURGLARY using conditional embedding; the fourth, A,

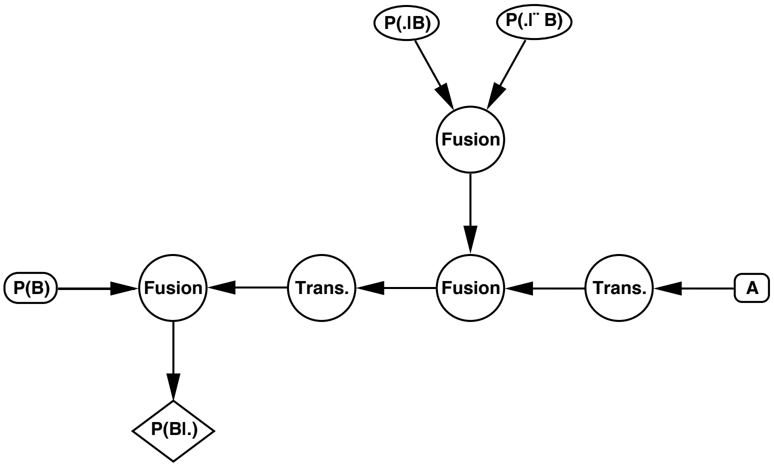


Figure A.1: Burglary analysis 1

attributes all of its mass to the ALARM proposition in the ALARM frame. Processing this information through the analysis in Figure A.1 results in a point probability for BURGLARY, at interpretation node P(B|.), corresponding to a 0.941% likelihood of a burglary, given the alarm is sounding; this is identical to Pearl's answer.

As Pearl's example continues on pages 42–46, we find that the home in question belongs to Mr. Holmes and that he is speculating about a possible burglary based upon a phone call from Dr. Watson. In this call, Dr. Watson states that Mr. Holmes' burglar alarm is sounding. However, since Dr. Watson is known to be a tasteless practical joker, Mr. Holmes telephones his neighbor Mrs. Gibbon. She says that his burglar alarm is sounding, but Mrs. Gibbon has occasional drinking problems and did not directly answer Mr. Holmes' questions about his burglar alarm. What is Mr. Holmes to conclude?

To incorporate this new information, we define four additional frames. The WATSON frame includes two elements, REPORTS ALARM and DOESN'T REPORT ALARM, as does the GIBBON frame. The ALARM&WATSON frame is the joint frame composed from the ALARM and WATSON frames and is connected by compatibility relations to each of these component frames. Similarly, the ALARM&GIBBON frame is the joint frame composed from, and connect by compatibility relations to, the ALARM and GIBBON frames. Given these additions to the gallery, the new information can be represented by four additional reports, one for each of these new frames. In the modified analysis (Figure A.2), report W corresponds to Dr. Watson's testimony and report G corresponds to Mrs. Gibbon testimony. Each attributes 100% belief to the proposition REPORTS ALARM, with W referencing the WATSON frame and G the GIBBON frame. The P(.|W) report attempts to capture Dr. Watson's reliability by conditionally embedding ALARM|REPORTS ALARM as 90% probable and NO ALARM|REPORTS ALARM as 10% probable over the ALARM&WATSON frame; report P(.|G) does the same for Mrs. Gibbon, conditionally embedding ALARM|REPORTS ALARM at 80% probable and NO ALARM|REPORTS ALARM at 20% probable over the ALARM&GIBBON frame. In this second analysis, these four reports combine to fulfill the role played by report A in the

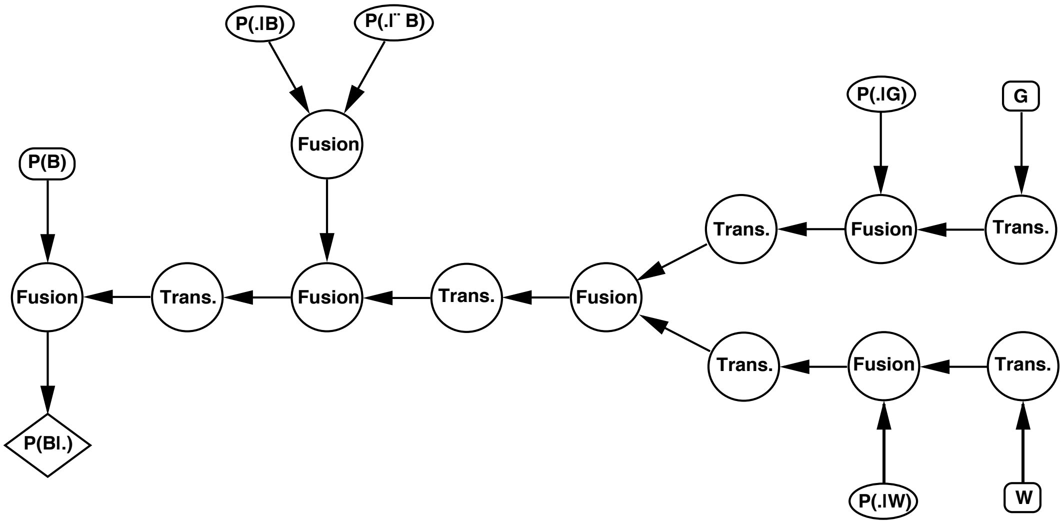


Figure A.2: Burglary analysis 2

first analysis. As a result, the likelihood of NO BURGLARY increases to 99.747% and the likelihood of BURGLARY decreases to 0.253%. Again, this result is identical to Pearl's.

Pearl's example continues (pages 47 and 48) with Mr. Holmes reasoning that his daughter would probably (70%) call him if the burglar alarm were sounding. If the burglar alarm is not sounding, then Mr. Holmes' daughter would certainly (100%) not call to report it. How likely is Mr. Holmes' daughter to call?

Again, we need to extend the gallery and modify the analysis to answer this new question. Two new frames are required: the DAUGHTER frame includes two elements, REPORTS ALARM and DOESN'T REPORT ALARM; the ALARM&DAUGHTER joint frame relates the DAUGHTER frame to the ALARM frame via two compatibility relations. The complete gallery is in Figure A.3. The report $P(.|A)$ conditionally embeds REPORTS ALARM|ALARM at 70% and DOESN'T REPORT ALARM|ALARM at 30% over frame ALARM&DAUGHTER, while report $P(.|\neg A)$ conditionally embeds DOESN'T REPORT ALARM|NO ALARM at 100%. To calculate the probability that Mr. Holmes' daughter will call and report the alarm (i.e., REPORTS ALARM in the DAUGHTER frame), the computational structure between P(B) and the ALARM fusion node is reversed to flow toward the ALARM node, and the ALARM node is conditionally translated, using $P(.|A)$ and $P(.|\neg A)$, to reach the DAUGHTER frame (Figure A.4). The interpretation node $P(D|.)$ calculates the likelihood that Mr. Holmes' daughter DOESN'T REPORT ALARM as being 81.20%; the likelihood that there is NO ALARM is calculated to be 73.15% at interpretation node $P(A|.)$.

As a final exercise, we extended the analysis to a full omnidirectional analysis (Figure A.5). This analysis answers all of the questions simultaneously. When given the same numerical entries as in Pearl's example, it calculates the same results.

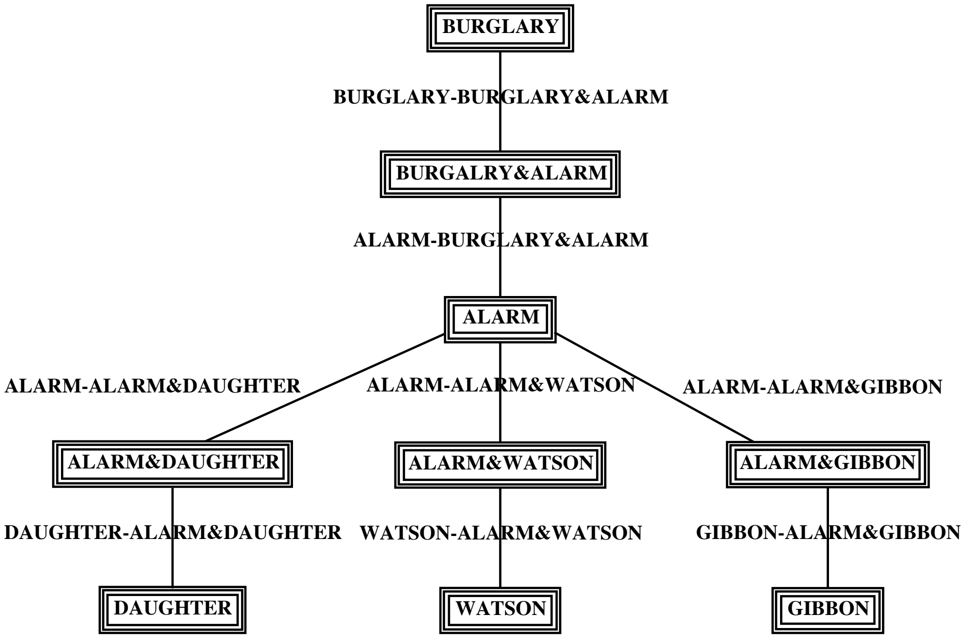


Figure A.3: Burglary gallery

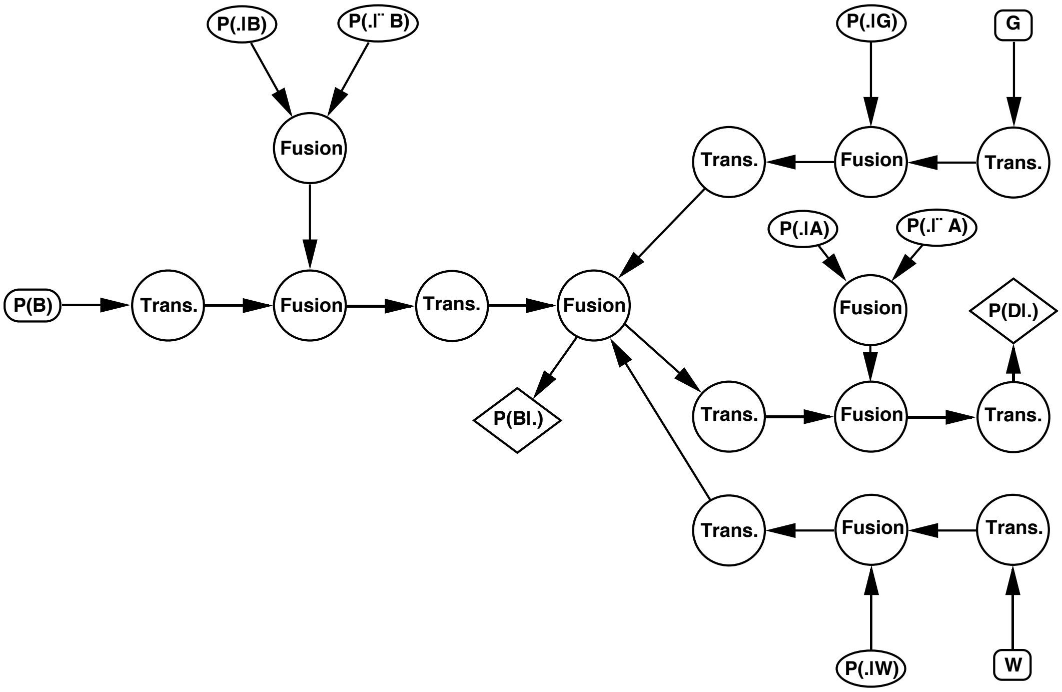


Figure A.4: Burglary analysis 3

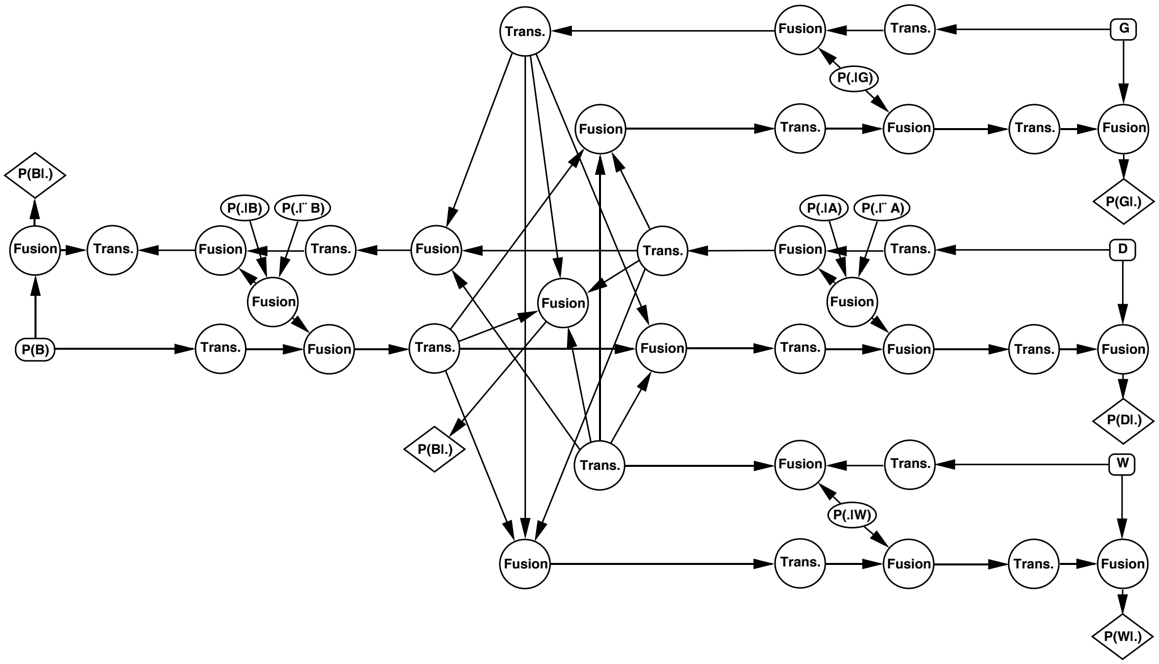


Figure A.5: Omnidirectional burglary analysis

Appendix B

Modifications since Version 2.0

This section highlights changes that have been made since the release of version 2.0. The text of this manual reflects the state of Gister-CL as of the version on the cover of the hardcopy or the version of Gister-CL under which the softcopy was found. The following is provided to aid users of earlier versions.

B.1 Version 2.01

This is the first version of Gister-CL that runs in Allegro Common Lisp and CLIM 2.0. Most of the changes to the system were made to make Gister-CL commpatible with Allegro Common Lisp and CLIM 2.0 and are generally transparent to the user. Changes visible to the user include:

- ANALYSIS EXAMINE' has been modified to edit values in a Text Editing Window rather than in the Lisp Listener Window.
- In preparation for the main package for Gister-CL becoming :ER (Evidential Reasoning), :ER is now a nickname for the :Grasper-CL package. User's should write new code using :ER as Gister-CL's main package.

B.2 Version 2.06

Beginning with versions 2.06, the `ER` package is distinct from the `Grasper-CL` package; although it "uses" the `Grasper-CL` package. To update a graph output under version 2.05 or earlier, edit the first line of the graph file changing it from (in-package :gsp) to (in-package :er); input it into Gister-CL; then output it.

Bibliography

[Bar81] Jeffery A. Barnett. Computational methods for a mathematical theory of evidence. In *Proceedings of the Seventh Joint Conference on Artificial Intelligence*, pages 868–875, Menlo Park, CA, August 1981. American Association for Artificial Intelligence.

[Dem68] Arthur P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, 30:205–247, 1968.

[DHN81] Richard O. Duda, Peter E. Hart, and Nils J. Nilsson. Subjective Bayesian methods for rule-based inference systems. In Bonnie Lynn Webber and Nils J. Nilsson, editors, *Readings in Artificial Intelligence*, pages 192–199. Tioga Publishing Company, Palo Alto, CA, 1981.

[DP90] Didier Dubois and Henri Prade. The logical view of conditioning and its application to possibility and evidence theories. *International Journal of Approximate Reasoning*, 4(1):23–46, 1990.

[Gar86] Thomas D. Garvey. Evidential reasoning for geographic evaluation for helicopter route planning. Technical Report 405, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, December 1986.

[GLF81] Thomas D. Garvey, John D. Lowrance, and Martin A. Fischler. An inference technique for integrating knowledge from disparate sources. In *Proceedings of the Seventh Joint Conference on Artificial Intelligence*, pages 319–325, Menlo Park, CA, August 1981. American Association for Artificial Intelligence.

[GS85] Jean Gordon and Edward H. Shortliffe. A method for managing evidential reasoning in hierarchical hypothesis spaces. *Artificial Intelligence*, 26:323–258, 1985.

[KLS93] Peter D. Karp, John D. Lowrance, and Thomas M. Strat. *The Grasper-CL Documentation Set.* Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, June 1993.

[Kon86] Chung Tung Augustine Kong. *Multivariate Belief Functions and Graphical Models.* PhD thesis, Department of Statistics, Harvard University, October 1986.

[LG83] John D. Lowrance and Thomas D. Garvey. Evidential reasoning: An implementation for multisensor integration. Technical Report 307, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, December 1983.

[LGS90] John D. Lowrance, Thomas D. Garvey, and Thomas M. Strat. A framework for evidential-reasoning systems. In Glenn Shafer and Judea Pearl, editors, *Uncertain Reasoning*, pages 611–618. Morgan Kaufman Publishers, Inc., San Mateo, CA, 1990.

[Low82] John D. Lowrance. *Dependency-Graph Models of Evidential Support.* PhD thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, MA, September 1982.

[LSG86] John D. Lowrance, Thomas M. Strat, and Thomas D. Garvey. Application of artificial intelligence techniques to naval intelligence analysis. Final Report SRI Contract 6486, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, June 1986.

[LSW$^+$91] John D. Lowrance, Thomas M. Strat, Leonard P. Wesley, Thomas D. Garvey, Enrique H. Ruspini, and David E. Wilkins. The theory, implementation, and practice of evidential reasoning. Final Report SRI Contract 5701, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, June 1991.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligence Systems.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[Rus86] Enrique H. Ruspini. The logical foundations of evidential reasoning. Technical Report 408, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, December 1986.

[Rus87] Enrique H. Ruspini. Epistemic logics, probability, and the calculus of evidence. In *Proceedings of the Tenth Joint Conference on Artificial Intelligence*, pages 924–931, Menlo Park, CA, August 1987. American Association for Artificial Intelligence.

[Sha76a] Glenn Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, NJ, 1976.

[Sha76b] Glenn Shafer. A theory of statistical evidence. In W. Harper and C. A. Hooker, editors, *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science*, volume II, pages 365–436. 1976.

[Sha82] Glenn Shafer. Belief functions and parametric models. *The Journal of the Royal Statistical Society*, 44(3):322–352, 1982.

[Sha86] Glenn Shafer. Belief functions and possibility measures. *The Analysis of Fuzzy Information*, 1, 1986.

[SL89] Thomas M. Strat and John D. Lowrance. Explaining evidential analyses. *International Journal of Approximate Reasoning*, 3(4):299–353, July 1989.

[Sme78] Philippe Smets. *Un Modèle Mathématico-Statistique Simulant le Processus du Diagnostic Mèdical.* PhD thesis, Free University of Brussels, Presses Universitaires del Bruxelles, 1978.

[Spi90] Marcus Spies. Combination of evidence with conditional objects and its application to cognitive modeling. In et al. I. Goodman, H. Nguyen, editor, *Conditional Logic in Expert Systems.* North Holland, 1990.

[SS86] Prakash P. Shenoy and Glenn Shafer. Propagating belief functions with local computations. *IEEE Expert*, Fall:43–52, 1986.

[SS88] Glenn Shafer and Prakash P. Shenoy. Bayesian and belief-function propagation. Working Paper 192, School of Business, University of Kansas, Lawrence, KA, April 1988.

[SSM87] Glenn Shafer, Prakash P. Shenoy, and K. Mellouli. Propagating belief functions in qualitative Markov trees. *International Journal of Approximate Reasoning*, 1(4):349–400, 1987.

[Str90] Thomas M. Strat. Decision analysis using belief functions. *International Journal of Approximate Reasoning*, 4(5):391–418, September 1990.

[Wes88] Leonard P. Wesley. *Evidential-Based Control in Knowledge-Based Systems.* PhD thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, MA, 1988.

[Yag83] Ronald R. Yager. Entropy and specificity in a mathematical theory of evidence. *International Journal of General Systems*, 9:249–260, September 1983.

Index