Lawrence Berkeley National Laboratory

AN EXPERTISE BASED ENERGY INFORMATION SYSTEM

Author
Rosenberg, S.

Publication Date
1980-04-01

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

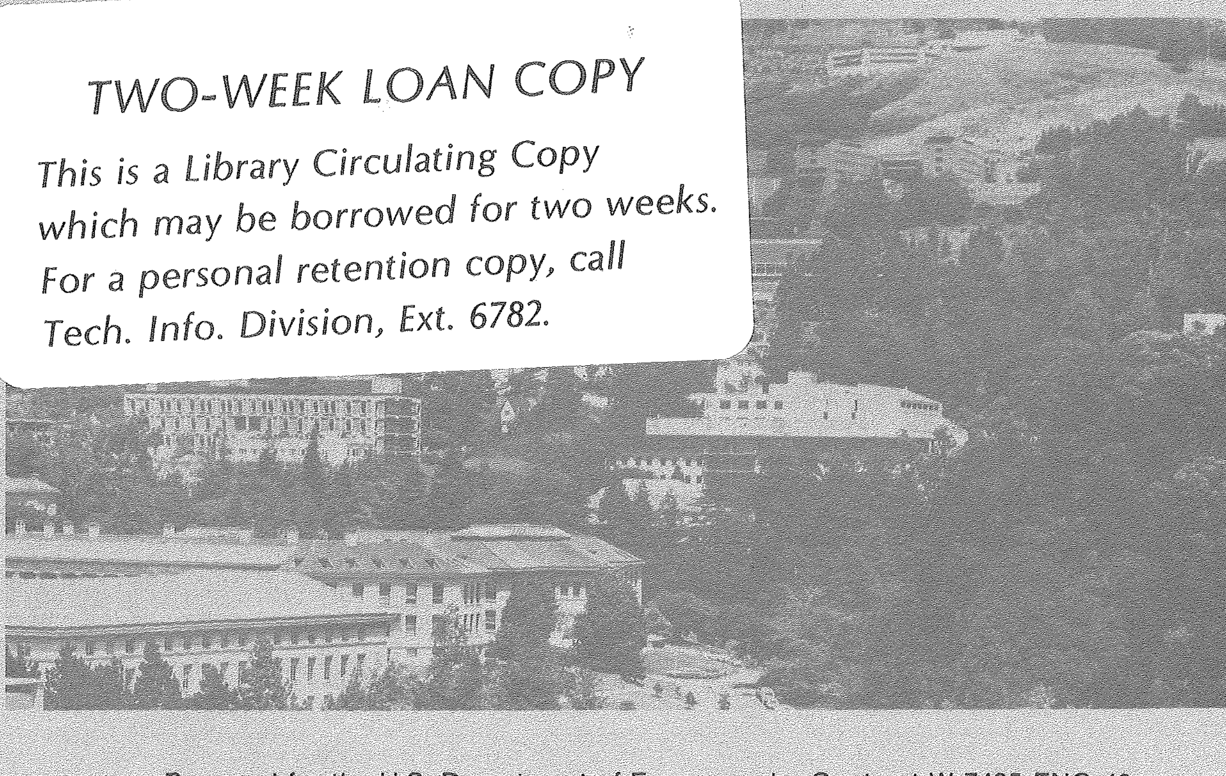
Employee & Information Services Division

To be presented at the American Society for Information Science 9th Mid-year Meeting, Pittsburg, PA, May 14-17, 1980

AN EXPERTISE BASED ENERGY INFORMATION SYSTEM

S. Rosenberg

April 1980



Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48

DISCLAIMER

Proceedings of the ASIS
Mid-Year Meeting
Pittsburgh, May 1980

LBL-10270

AN EXPERTISE BASED ENERGY INFORMATION SYSTEM

S. Rosenberg

Information Methodology Research Project
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Prepared for the U. S. Department of Energy
Technical Information Center
under contract No. W-7405-ENG-48

AN EXPERTISE BASED ENERGY INFORMATION SYSTEM

S. Rosenberg

Information Methodology Research Project
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

This paper describes an intelligent decision support system for energy information. The system draws on Artificial Intelligence technology, both in the design of the reasoning and representation components, and in the computer language. The computer language supports a powerful declarative semantics based on frame hierarchies. The design supports desirable user features such as substitution for incomplete data, data validation, caveat warnings, and simple noticing. A model of oil flow in the United States forms the semantic basis for the system. A reasoning compoment operating within this representation provides answers to queries; a capacity for analyzing energy scenarios; and complex monitoring of the database for developing trends.

AN EXPERTISE BASED ENERGY INFORMATION SYSTEM

S. Rosenberg

Information Methodology Research Project
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

The Information Methodology Research Project has, as one of its focuses, the goal of developing intelligent information systems for dealing with energy resources in the United States. We are currently engaged in designing one such system, for dealing with information concerning petroleum resources and use. We expect that solutions to representation and reasoning problems in this domain will generalize to other energy resources, such as coal, uranium, etc. Our long term goal is to aid the Department of Energy in developing a single, comprehensive information system covering all aspects of energy use. A first step in the process is the development of an intelligent information system within the petroleum domain to provide capabilities currently either unavailable or performed by human analysts. In this paper I will present a brief overview of our project. I will then focus on the development of a reasoning component, designed to help decision makers.

Our goals are ultimately quite practical; namely the transfer of A. I. "technology" into a real world domain. It is useful to start by considering the constraints this imposes on the design. Unlike existing expert systems, such as Prospector (Duda et al., 1978), we are not free to create ideal semantic representations. Over 200 databases dealing with energy resources are already maintained by the DOE. Our representation scheme must be able to use this existing knowledge. Similarly, there are limits on the types of information which can be collected. (Consider, for example, the controversy over the recent U. S. oil shortage--did the shortage really exist? Not enough relevant information exists to decide.) Of the data available, there are problems with validation, with information gaps, with variable definitions of terms, etc. The strength of our expert system will depend on the quality of the input. Indeed, one of our short term goals is to isolate shortcomings in existing information collection systems which hamstring the development of the intelligent support system. Since the system will do the same types of tasks as a human energy analyst does, information gaps which hinder it will also hinder the human. The task of creating a representation methodology capable of using actual data, while serving as the basis for an intelligent support system, provides a focus for defining the required types of abilities.

Several of our colleagues (Krishnan and Cahn, 1980) are developing precise formal models for the flow of energy resources. Any representation must capture the features of such models so that potential existing databases on energy resources can be mapped into it. Real data in our domain is often "messy." Crucial information is sometimes missing, incomplete, or invalid. To be useful, an intelligent support system must provide ways around these problems. Such methods as default procedures, cross validation checks, backtracking, caveats, and constraint monitoring provide the core of such a system. We have chosen to use FRL (frame representation language) as the basis for our representation. In FRL, a frame can make use of inheritence, default values, procedural attachments, etc. This augmented notion of what a data object is allows us to create the type of "friendly" representation we need.

A "friendly" representation takes the burden of performing routine, if sometimes complicated, functions, from the reasoning component. Some of these functions are quite simple, such as providing aggregated information, or default values. some are complex, such as adjusting the representation by changing deduced consequences when erroneous facts are corrected. Some we do not know yet how to do, such as handling fuzzy information. The net effect of such a friendly system is to allow a user or reasoning component to focus on doing higher level tasks, while leaving lower level information processing to the representation system. In effect, we propose that in many real world domains, semantic representations must function dynamically, drawing on interlaced procedural and world knowledge to provide a solid basis for higher level reasoning. The ability of human experts to fill information gaps, assess credibility, etc. forms a central aspect of their expertise. Consequently, developing such a flexible representation forms an essential basis for modeling our domain.

The basic semantic system is constructed using FRL (Roberts and Goldstein, 1977). FRL is a sophisticated, higher level language specifically designed for the representation of knowledge in a variety of domains. It provides a hierarchically organized, frames-based semantics with inheritance and procedural attachments among other features. FRL is in turn written in LISP (Moon, 1975), hence is compatible with normal LISP code.

FRL is a Frame Representation Language based on Minsky's (1975) notion of frames. Goldstein and Roberts (1977) have developed this working frame system which forms the basis for our knowledge representation. FRL has been used to implement NUDGE (Goldstein and Roberts 1977), a system for maintaining a person's schedule of activities in the face of individual preferences, conflicting constraints, and changing plans; PAL, a natural language front end for NUDGE (Bullwinkle 1977); TRIPPER, a knowledge base for places and travel around the country (Jeffrey 1977); a representation for the discourse structure of news articles (Rosenberg 1977); and COMEX (Stansfield 1977), a system for understanding discourse about the commodities market.

Symbol manipulating languages such as LISP provide mechanisms for organizing collections of properties for objects. IN LISP, such properties are associated with objects in property lists of attributes. Such attributes can be paired with arbitrary values in attribute/value pairs.

FRL extends the traditional characterization of properties as attribute/value pairs by allowing properties to be described by comments, abstractions, defaults, constraints, indirect pointers from other properties, and attached procedures. A value of a property becomes one of the range of potential descriptors. A frame can be thought of as a named collection of slots which form the semantic definition of a concept. These slots define the properties of the frame (i.e., they form a list of such properties). Each property can have many values. A slot (= property) can be specified further through the use of an arbitrary number of associated user and system defined "facets." One of these facets will be the traditional "value" of attribute/value pairs in property lists. Useful system defined facets are: Value, which contains the value of that slot; Default, which specifies a default value; Require, which specifies procedural constraints on the values for that slot; If-Needed, which specifies procedures that compute a value for the slot; and If-Added and If-Removed, which specify actions to be taken when a value is added or removed. Notice that many of these facets are procedural attachments. Each slot can have associated procedures which can perform calculations when required. Thus a frame in FRL is more than a simple datastructure.

FRL allows concepts (represented as frames) to be arranged in an inheritance hierarchy using the AKO (A Kind Of) slot. The value of this slot is a generic frame of which the current frame is a specialized instance. Thus the frame system forms a tree structure. Generic information is stored higher up in the hierarchy and shared by frames lower down; specialized frames specify new distinguishing knowledge. The generic knowledge, including computational procedures, is inherited automatically.

A small testbed model serves as a basis for developing the intelligent support system. The model serves as a domain for an "energy expert" capable of reasoning and making decisions about energy scenarios. Support, for example, an overseas supplier of crude oil decreases supplies to the U. S. One question which might be asked is whether supplies of gas and heating oil are sufficient. Ignoring for the moment quantitative calculations, a simple reasoning chain might be:

```
Can Supply be increased?
    Can imports be increased?
        NO, SINCE WE ASSUME A DROP IN IMPORTS.
    Can production be increased?
        Can local production be increased?
          NO.
        Can foreign production be increased?
          NO, IT IS DECREASING.

Can demand be reduced?
    Can exports be decreased?
        THERE ARE NO EXPORTS.
    Can consumption change?
        Can gas consumption be decreased?
          YES.
```

Therefore, demand for gas can be reduced to allow greater production of heating oil.

Some theorems used might look like:

```
Thm1    To adjust   Supply and Demand
           either   Supply = Demand
               or   increase supply
                    decrease demand
```

```
Thm2    To   increase supply
        or   increase import
             increase production

Thm3    To   increase production
             if location = domestic
                then fail
             else increase foreign-production

Thm4    To   increase imports
        or   increase production--foreign
             increase price

Thm5    To   decrease demand
        or   decrease exports
             decrease consumption

Thm6    To     decrease exports
        if     exports = 0, fail
        else   set exports = 0

Thm7    To     decrease consumption
        or if  gas not fixed
               decrease gas
           if  heating-oil not fixed
               decrease heating-oil

Thm8    To   decrease gas
             increase heating oil
```

All knowledge is represented as frames. Rules are expressed as productions (Newell and Simon 1972). These productions are in turn translated into rule frames with condition and action slots. The only indication that such declarative knowledge is a rule consists in the value of the generic pointer. Thus rules are semantically defined, but represented as declarative knowledge in the frame tree. As such, all features of a hierarchical frame representation are available, such as the use of inheritance, procedural attachments for dynamically calculating needed values, the ability to use semantic relations in determining an appropriate rule, and so on. Rules frames are considered to contain competence knowledge.

To use rules, a rule frame is interpreted as a procedure, with the slot values controlling the interpretation. Thus a condition slot causes a condition to be tested; the action slot specifies the action to be performed and so on.

The competence knowledge expressed in a rule frame can be used in many ways. Since frames can be used for representing complex situations in whose occurrence we are interested, we would like to have rules which, given a target frame, instantiate it as relevant information is added to the database. We will often want to notice constraint violations in our database. During deductive processes, relevant information may not yet exist in the database. We may want to suspend deduction, and have it automatically proceed as sufficient information is added (Rosenberg, 1979). We may wish to do hypothetical reasoning on possible energy scenarios. We may wish to apply a rule once, until a criterion is met, or until further notice. We may wish our rules to be triggered by the removal, not the addition, of appropriate information. Many other possibilities exist. All of these use the same competence knowledge contained in the rule frames.

By using variable interpretation we can use the same competence knowledge in all of these cases. We vary the interpretation of a rule frame for each of these uses, by adding the appropriate descriptive label for that instance of the rule. This type knowledge is then used by the interpreter in applying the competence knowledge in the rule frame.

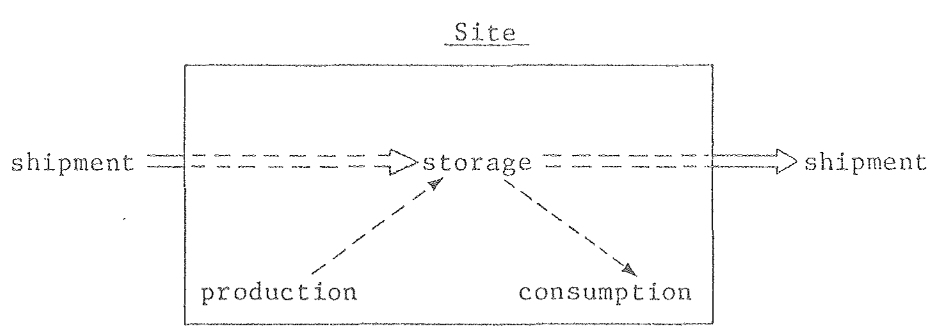
To sum up, all rules are expressed as declarative knowledge, in rule frames. Thus, everything is a frame. This allows manipulation of the contents of rules using the semantic power of RFL, together with any metarules required. Rules are applied by a process of variable interpretation which allows the same declarative knowledge to be used in several procedural modes. The most useful of these, for our purposes, are ordinary deduction, and to enable noticing of developing situations.

The test bed serves as the focus for developing reasoning and representation segments of the system. While semantically simple, semantic detail is being transferred from the more formal, if non-computational, models mentioned earlier, in an incremental process. The testbed provides a useful domain for the development of the reasoning module.

The basic semantic system is constructed using FRL (Roberts and Goldstein, 1977). The model organizes information around the following fundamental categories: site, area, and company. These represent the basic physical loci whose petroleum usage we wish to keep track of.

A site represents any actual physical location at which oil is handled, such as a port, tank farm, refinery or oil field. Areas, such as states, are considered to consist of a set of sites physically located within their boundaries, while companies, are represented by ownership of a collection of sites in arbitrary locations.

Since many types of activities can occur at sites, and by extension, within states and companies, it is necessary to develop a simple cannonical model capable of representing within one basic type the different activities of refining, storage, extraction, and so on. Conceptually, we can think of any site as being some instantiation of the following scheme:



We can accurately characterize sites, and, by extension, states and companies which are physical and social collections of sites) in terms of this structure. A site, as a physical location, always represents oil in storage. Shipments coming in add to this store. Production, if it exists at a site, also adds to the store. Shipments from the sites reduce the storage. Some oil is locally consumed, either because of shipment costs, refining costs, leakage, or end-use by consumers in the local service area of that site.

We extend this to the frames representing sites, states and companies by giving them each the following set of slots (= features) in common, in addition to other slots they might have. This preserves an essential conceptual identity in the way we treat sites, states and companies. (a) bought; (b) sold; (c) consumption; (d) production; (e) carryover.

(a) The bought slot contains cumulative information concerning the volume of oil shipped to that site (or company, or state). (b) The sold slot contains the cumulative amount of oil sent from that site (or company, or state). (c) Since these two slots contain aggregate information, summed over many shipments, they do not tell us how much oil is available for shipment or consumption at a site at any one instant. this is dynamically calculated through the use of the carryover slot. This has each new shipment or production added to it, and each consumption or ship-out subtracted. It contains the actual storage amount available at a site at any given time. (d) The consumption slot represents how much oil is locally consumed. (e) The production slot, how much is locally produced.

The site frame contains additional slots, among which are the source-from slot, which contains a list of shipments which have gone to that site, and source-to, which is a list of shipments from that site, allowing examination of the raw data, if needed. Ownership and locative relations among the sites, areas and companies are also encoded into the frames.

Sites are owned by companies (which may in turn be subsidiaries of other companies) and located in areas. It follows that information contained in frames for particular companies and states will be aggregated over the sites they own or contain. Sites represent the smallest physical grain in our system beyond the raw data.

Generic frames represent the semantics in our model. Instances of these generic frames are used to represent actual data. This data is entered in particular instances of generic frames, which inherit the semantic properties of the generic type. These instances represent the grain of our model. Besides the physical grain, there exists a time grain. Frame instances will have a particular date specified. We have chosen to use a grain of a month. The basic data elements for sites, areas and companies will consist of month by month instances of these generic types, showing the cumulative data for that month. Other frames provide chronological aggregation over the physical grain, by having a cumulative time span of a season or a year. Thus, we might ask about oil consumption in New England last winter.

The basic transactional unit in our model is a shipment. Raw data is entered as shipments between sites. A shipment has the following structure:

Shipment:
- price
- volume
- status
- bought
- sold
- date

This structure is sufficient to specify all transactions which occur in our model. Each shipment specifies the price and volume of the oil involved; where it came from (bought), and where it is going (sold). The status slot specifies which class this shipment falls into. (Shipments are marked as to whether they are imports, production, destined for export, are merely being transhipped (say from one state to another), or being shipped to the final consumers.)

Sites form the physical locus for all transfers of petroleum. All transactions between sites involve shipments of petroleum from one site to another, the adjustment of the cumulative statistics maintained for each site, and changes in the available carryover of each site.

Production and consumption are viewed as within site transactions that are also instantiated as shipments. To produce oil at a site representing an oil field, for example, and send it to another site, involves two shipments. The first shipment is from the production at the site to the site's storage, where it gets added to the site's carryover, and the second is from that site to another site.

Sites and shipments serve as the interface between the continuous aspects of production, consumption and transportation, and the discrete model. The two cannonical entities of shipments and sites provide a simple basic semantic structure for the entire system. These provide the two elements necessary for a transaction.

There is only one type of transaction possible, shipping oil, and this occurs only within or between sites. All other entities in the model represent aggregation of data. Aggregation occurs for the owner (company) of each site involved in a shipment, and for the location (state) of each site. This can in turn trigger further aggregation, as when a company is a subsidiary of another, or when an area is part of a larger geographical grouping (such as the New England states). All transactions are annotated by linking the elements involved (one shipment and either one or two sites). The shipment is added to the site frames' source slots. Thus, the raw data for all subsequent calculations and resulting changes is recorded at this level and can be examined if necessary. The purpose of the model is to gather information; display the state of the oil flow system at any one time; and have enough grain to allow questions to be answered which require computations. The model, although intended primarily as a data-structure for storing and retrieving information about oil flow, also is a model of the process in a more rigorous sense. Data is not just "dumped" into records; its entry triggers processes which in symbolic form mimic the state of the world.

One of the features of our representation scheme is the use of procedural hooks of various types. One of these, the "if-added" hook, responds to additions in its domain. We place such an "if-added" hook on the "instance" slot of the generic "petroleum-shipment" frame. It will trigger whenever a new instance of a petroleum-shipment is added by the "shipper," and call an aggregation module which examines the shipment and modifies the sites accordingly. The "if-added" functions as a "Trigger" which informs the aggregator when information relevate to its function has been added. This expresses part of the design philosophy. Different modules will be able to place their own triggers, allowing us to "evolve" the model by adding more modules.

Such procedural attachments on the frames representing our semantic entities provide a great deal of power. Values can be dynamically calculated when needed (i.e., requested), although perhaps never reported, through If-Needed attachments. Of course, this means we must know how to calculate missing information. Automatic deductions can be triggered to fill in missing data, and maintain consistency among related facts, which may be individually reported or modified. (The If-Added or If-Removed attachments.) Default values can be specified and used only if actual values are unavailable.

These procedures are inherited by all instances of the generic frames on which they are placed. Thus, this knowledge can be shared without having to be individually specified for each frame. For instance, all basic knowledge about supply and demand relations for sites is contained in one useful place, the generic site frame. It is used as needed by specific instances of sites. In a heritage tree, an instance may inherit these processes from many superior nodes. This demonstrates the layering of procedure specifications so that each layer is generalizable to appropriate daughter frames.

In our system we use caveat attachments which are keyed to actual retrieval demands for particular facts. These provide warnings on the reliability of data, and on critical states. For instance, they are used to tell us when reserves at a site fall too low. The requirement attachment allows automatic checking of data-additions, important in changing databases, to maintain the integrity of the database. For example, the production slot of the Supply-Petroleum frame might have a requirement that the source of any value must be one of an authorized list of producers. If-added attachments also provide simple noticing power, by warning when values change. For example, if carryover of petroleum for a month ever drops below a threshold value, we can be warned of the pending shortage.

Since FRL is capable of automatically applying requirements, we can cause our database to request confirmation if new values fall outside acceptable ranges. Thus we can dynamically encode many validation techniques into our database. Caveats are used in conjunction with validation. If data can be unreliable, or exceed average boundaries, be reported by unverified sources, or simply be reported in aggregated form rather than directly reported, then we would like users to be warned. Hence we associate caveats of various types with information requests.

Our representation methodology provides a natural way to interleave a procedural semantics with the declarative knowledge. This enables us to build the sort of friendly representation necessary to provide information in our domain.

Acknowledgment

This work was supported by the Division of Institutional Relations of the U. S. Department of Energy under contract No. W-7405-ENG-48.

References

Duda, R. O., Hart, P. E., and Reboh, R., "Rule Based Consultation Program for Mineral Exploration," in Proceedings of the Lawrence Symposium and Decision Sciences, 1977, pp. 306-309.

Goldstein, I. P. and Roberts, R. B., "NUDGE: A Knowledge-Based Scheduling Program," AI Memo 405, MIT, February 1977.

Jeffrey, M., "Representing PLACE in a Frame System," MS Thesis, A. I. Lab, MIT, 1977.

Krishnan, V. V. and Cahn, D. F., "Material-Flow Data Structures as a Basis for Energy Information System Design, ASIS Mid-Year Conference, 1980.

Minsky, M., "A Framework for Representing Knowledge," in The Phychology of Computer Vision, P. H. Winston, ed., McGraw Hill, NY, 1975.

Moon, D. A., "Maclisp Reference Manual," Laboratory for Computer Science, MIT, December 1975.

Newell, A. and Simon, H. A., "Human Problem Solving," Prentice-Hall, NJ, 1972.

Roberts, R. B. and Goldstein, I. P., "The FRL Manual," AI Memo 409, MIT, June 1977.

Rosenberg, S., Reasoning in Incomplete Domains, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 1979.

Rosenberg, S., "Frames-Based Text Processing," AI Memo 431, MIT, 1977.

Stansfield, J., "COMEX: A Support System for a Commodities Expert," AI Memo 423, MIT, 1977.