Title
COMMON SENSE REASONING ABOUT PETROLEUM FLOW

Author
Rosenberg, Steven

Publication Date
1981-02-01

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

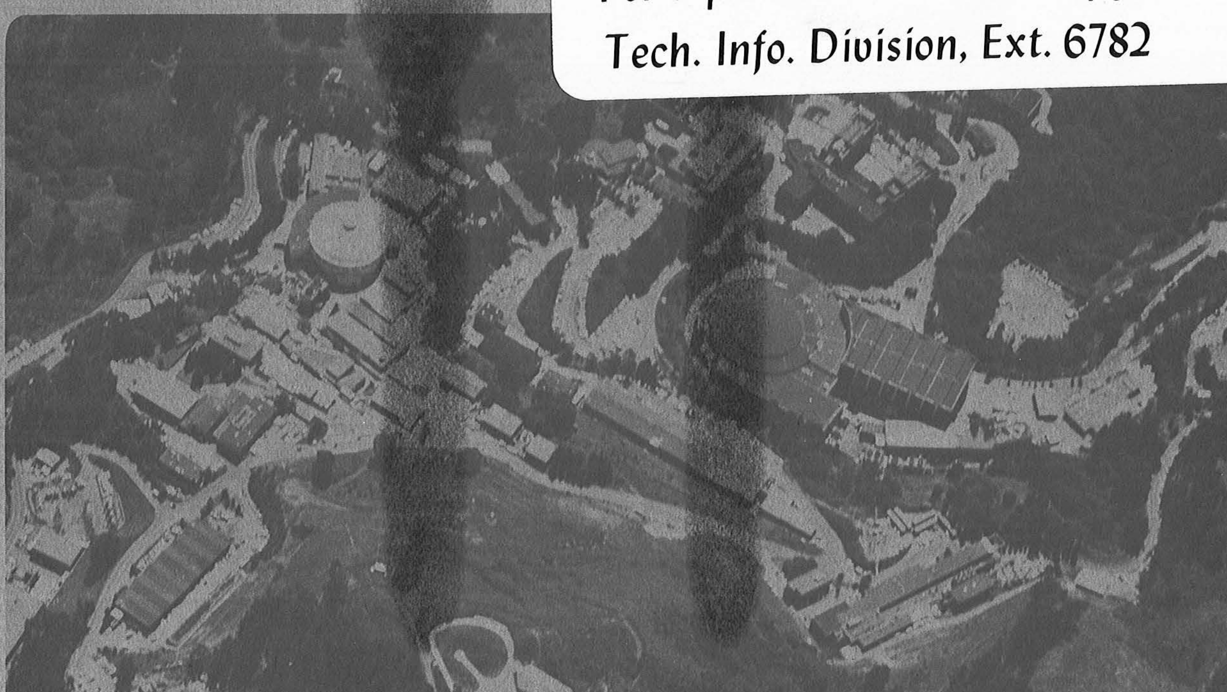
Physics, Computer Science & Mathematics Division

Submitted to the International Joint Conference on Artificial Intelligence, Vancouver, Canada, August 24-28, 1981

COMMON SENSE REASONING ABOUT PETROLEUM FLOW

Steven Rosenberg

February 1981



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

Common Sense Reasoning about Petroleum Flow

By

Steven Rosenberg
Computer Science and Mathematics Department
Lawrence Berkeley Laboratory
University of California
Berkeley, California

Abstract

This paper describes an expert system for understanding and reasoning in a petroleum resources domain. A basic model is implemented in FRL (Frame Representation Language). Expertise is encoded as rule frames. The model consists of a set of episodic contexts which are sequentially generated over time. Reasoning occurs in separate reasoning contexts consisting of a buffer frame and packets of rules. These function similar to small production systems. Reasoning is linked to the model through an interface of Sentinels (instance driven demons) which notice anomalous conditions. Heuristics and metaknowledge are used through the creation of further reasoning contexts which overlay the simpler ones.

This work was supported by the Office of Energy Research of the U.S. Department of Energy under contract W-7405-ENG-48.

LBL-12325

INTRODUCTION: The Computer Sciences and Mathematics Department (CSAM) of the Lawrence Berkeley Laboratory maintains many large databases. We are currently exploring the usefulness of expert systems in providing advanced capabilities for use with these databases. This research has focused primarily on understanding and using information concerning petroleum resources in the U.S. Our goal is to model the flow of oil and understand "what if" questions. (e.g. what if Iran stops exporting oil to New York?)

MODEL: A small testbed model serves as the focus for development. The model organizes information around the notion of transactions between Sites. Sites represent physical loci at which oil is handled, such as a port, refinery, oil field, or tanker farm. Sites are in turn owned by companies and located in areas. The basic transactional unit is a shipment. A transaction consists of some transfer of oil between sites. There are several possible types of transaction, such as importing, exporting, producing, consuming, or transhiping oil.

The basic semantic system is constructed using FRL (Roberts and Goldstein, 1977). FRL is a Frame Representation Language based on Minsky's (1975) notion of frames. It provides a hierarchically organized, frames-based semantics with inheritance and procedural attachments among other features.

CONTEXTS: Groups of transactions are organized into episodic contexts. Each context represents all transactions for a particular month. At the beginning of the month, a new context is created with initial values generated from the previous month's context. At this point, transactions can occur which will drive the model, causing changes in various petroleum supplies as oil is shipped, consumed, produced, etc. during the course of that month. Old contexts are saved.

Contexts are linked through the set of initial values which one context gets from the previous context. Suppose a report on a transaction arrives late, after that month's context has been "closed", and new contexts created for subsequent months. If we update that context, by adding the transaction, then all subsequent contexts will have false information since their initial values were derived from the now preliminary data in the changed context. The solution we have adopted is that of a simple form of truth maintenance (Doyle, 1978). Changes in one context which affect knowledge in another context will propagate to the new context. Sentinels (Rosenberg, 1979) monitor these changes and update the consequences.

As transactions occur, the episodic contexts will show the cumulative results for that month. Other contexts can overlay these monthly ones. Currently, we are interested in chronological aggregation over larger time spans, and over other semantic entities. (e.g. how much oil was consumed in New Jersey last winter?) Information in this aggregation context is derived from the episodic contexts.

RULES: Rules consist of condition/action statements (i.e. productions (Newell & Simon, 1972)). Since all knowledge is represented as frames, rules are expressed as frames with condition and action slots. A rule frame is interpreted as a production with the slot values controlling the interpretation. A condition slot causes a condition to be tested and the action slot specifies the action to be performed. Rules operate in special contexts called reasoning contexts. A reasoning context consists of a single buffer frame, and a set of rule frames organized into packets. The buffer frame contains slots for the various pieces of information which may be required or generated by the rules, and a single goal slot which functions as a queue.

A reasoning context functions as a pseudo-production system. Each active rule places triggers in the buffer frame. The network of triggers embody a discrimination net of rule conditions. The order of application of rules can be controlled through the activation of rule packets, principally by heuristic rules, and by priorities for individual rules generated by metaknowledge. A reasoning context is triggered by the assertion of information. The reasoning context for shipping oil contains rule packets for importing oil, consuming oil, and so on. The result of reasoning is either the assertion of information back into the database, the evocation of another reasoning context through the assertion of information into that context, or the instigation of some action in the database, such as shipping oil.

SHIPPING OIL: The model works through the creation of transactions which ship oil. Shipments are created either by the user through a menu interface, or internally generated whenever the local semantics indicate it. For instance, whenever Dallas has more than a million barrels of oil in storage, it ships to its customers. A typical transaction might be to ship 200,000 barrels of oil from Dallas to Newark in January.

SENTINELS: The link between the model and reasoning is through the mechanism of Sentinels (Rosenberg, 1979). Sentinels monitor local conditions, and when warrented, assert goals. For instance, when the oil stocks of an exporter are above some level, the goal to ship this oil to importers is created. Sentinels also notice constraint violations and assert goals to correct them. For example, if Iran ceases to export oil, importers must find alternate sources. We also want to monitor more complex developments. Small reductions in supply by various producers, together with changes in demand at several sites can result in a severe shortage at one particular site. However the change at any one other site is not significant in itself. Such dynamic noticing is also done by Sentinels.

REASONING ABOUT OIL SHORTAGES: The identification of a problem directly evokes a small reasoning context specialized for such problems. For example, the Oil-supply-context is the context where reasoning about shortages and surpluses of oil occur. Information about significant changes in demand or supply at different sites is recorded here in the buffer frame. Goals to respond to these changes are also asserted here. A simple example of the reasoning that might occur when an oil shortage at Newark is noticed, and the goal of eliminating this shortage is asserted, is as follows.

```
Rule1: IF    (oil shortage at X?) and (goal ==> Increase supply)
       THEN  (or (Goal ==> Buy oil on spot market)
                (Goal ==> Share oil shortage among U.S. sites))

Rule2: IF    (Goal ==> Buy oil on spot market)
       THEN  (IF (Price < Limit) ==> (assert (choose supplier)))

Rule3: IF    (Oil shortage at X?) and (Oil supplier = Y?)
       THEN  (Ship oil from Y? to X?)
```

The first sub-goal in Rule1 is an attempt to reduce the shortage by buying sufficient oil on the spot market. If the current price of oil is less than the current limit we are willing to pay for oil on the spot market, Rule2 will choose a supplier from the set of oil exporting countries. The addition of a supplier triggers another rule, Rule3. Rule3 is triggered when there is both a buyer and a supplier of oil. Its action consists of creating a shipment, Shipment1, using the information available through the buffer frame, and then asserting the goal (goal ==> ship Shipment1). This goal will cause a transaction to be generated.

This approach provides only a first order solution. For instance, buying oil on the spot market can drive its price up. At some point this becomes less desirable than reducing demand by restricting gas station hours. Similarly, it has been a general policy to try and share gas shortages equably among the states. Such interactions require access to heuristic and metaknowledge that can be used to select among the rules.

Heuristic knowledge is encoded as rules which activate and deactivate other packets of rules. Thus heuristic rules control the selection of other groups of rules. Metaknowledge is encoded as metarules. Metarules take goals as their conditions, and produce rules which may achieve the goal and also satisfy the metaknowledge. Thus the piece of metaknoweldge "shortages can be avoided by buying more oil" would take (Goal ==> Oil shortage at X?) as input, and produce Rule2 as output.

Before entering a rule context, the metarules can be collected and evaluated. The result is a list of rules suggested by the metaknowledge. This provides a preferential ordering to use when more than one rule is applicable. In the ideal case, using metaknowledge will produce unique advice. However, there may be goal conflicts where two different rules are prefered by separate pieces of metaknowledge. In this case, it is necessary to have a goal resolution strategy. Our solution is to give each piece of metaknowledge a numerical priority. This allows us to order rule selection so that more important metaknowledge dominates less important knowledge. When goal conflicts result, the rule with the highest priority (derived from the metarules that suggest it) is tried first. The result is a small plan that says "first try this, and if it fails, then try that."

Sometimes, there will be a goal conflict which cannot be resolved by the priority of metaknowledge, since two rules can have the same priority. Our resolution principal is to look for a piece of metaknowledge which dominates the two conflicting metarules (Wilensky, 1980). Thus metarules can be treated as rules which themselves are subordinate to further metaknowledge.

References

Doyle, Jon, "Truth Maintenance Systems for Problem Solving," AI TR-419, January 1978.

Minsky, M. "A Framework for Representing Knowledge," in P. H. Winston (Ed.) The Psychology of Computer Vision, McGraw-Hill, N.Y., 1975.

Newell, A. and Simon, H.A. "Human Problem Solving," Prentice-Hall, N.J., 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.

Wilensky, R. "Meta-planning," Proceedings of the First Annual National Conference on Artifical Intelligence, 1980.