

UC Irvine

ICS Technical Reports

Title

A Model-Driven Question Answering System for Mixed Initiative CAI

Permalink

<https://escholarship.org/uc/item/4kv4z5t5>

Authors

Brown, John S.
Burton, Richard R.
Zdybel, Frank

Publication Date

1972-09-01

Peer reviewed

A MODEL-DRIVEN
QUESTION ANSWERING SYSTEM
FOR MIXED INITIATIVE CAI

by

JOHN S. BROWN
RICHARD R. BURTON
FRANK ZDYBEL

TECHNICAL REPORT #22 - September, 1972

TABLE OF CONTENTS

Introduction	1
Representation of Processes	3
Problems of Context	7
Question Answering With the Model	8
Representation of Factual Information	14
Static Question Answering	17
Combined Inferencing Techniques	19
Future Plans	20
Conclusion	21
References	23
Appendix 1 - Meteorology Model	24
Appendix 2 - Meteorology Questions	28
Appendix 3 - Sample Dialogue	30
Table 1 - Augmented Transition Table	31
Table 2 - Process Automata	32
Table 3 - External State Vector	33
Table 4 - Inference Tree	34

Introduction

An intriguing goal for a CAI system is the ability to perform as a general purpose question answering system. Several years ago Jaime Carbonell at M.I.T. commenced an ambitious research project to investigate the feasibility of such a system [3,4]. His goal was to create a mixed initiative CAI system which would permit the student to interrupt the tutorial process and pose direct questions. Although Carbonell's system performed quite well with highly structured geographic data, its direct application to less uniformly structured data seemed problematic. The inferencing techniques used were clearly tuned to answering simple factual questions. Nevertheless, the philosophy underlying the mixed initiative system remains impressive.

Our primary goal was to design a question answering system (for a mixed initiative CAI system) which would be able to handle questions concerning causal relationships between events. The notion of causality is central to the characterization of all physical phenomena and it also underlies much of our common sense knowledge of the world.

Because our question answering system is to function as a pivotal element in a CAI system, it must be not only efficient (cost effective) but it must incorporate a

representation of information (inference rules, etc.) that facilitates the addition of new concepts as well as new facts. Although this may seem an easy task, the reader may be well surprised how difficult it is to create "inference rules" for question answering machines which characterize some elements of knowledge.

In view of these goals, we rejected the highly popular approach which invokes resolution type strategies requiring an axiomatic representation of knowledge about the world. (For a general treatment of resolution see Nilsson [5].) Although such systems are extremely attractive because of their generality, this benefit is more than offset by:

- 1) The degree of difficulty in obtaining a complete axiomatic characterization of a body of knowledge complicated enough to be interesting.
- 2) The substantial complications involved in the handling of causality which often require complex considerations of the current "state of the world".
- 3) The inherent inefficiency of uniform proof procedures.

On the other hand, we did not want to construct a system so specialized that it would be applicable to only one area of knowledge or be able to answer only foreseen questions.

Metcorology provides an ideal subject area for studying the problems of representing causal phenomena. Introductory

meteorology is characterized by the study of the processes of evaporation, condensation, cloud formation, precipitation, etc. and their effects upon each other. These processes are amenable to a general representation in a qualitative manner. To explain the processes which cause rain, for example, lengthy logical explanations can be offered which need not be preponderantly quantitative.

The following sections discuss a question answering system that satisfies the basic requirements outlined above. The system is written in UCI LISP (Bobrow et al. [1]) for the PDP-10. The response times of our question answerer are encouraging. Most of the short questions in Appendices 2 and 3 get answered immediately (teletype speed) and none take longer than a few seconds. The system is approximately sixty thousand words, but more than half of this is data.

Representation of Processes

Thorough understanding of any process demands a comprehension of its interaction with other processes. We therefore require rules of inference which characterize not only processes but also their interrelationships. In particular, we seek a representation in which processes can be symbolically executed and in which their interactions can

be made perfectly explicit.

An obvious possibility for encoding such information is to employ a programming language (such as LISP) and symbolize each process as a procedure. The "meaning" or extensional definition of a process could then be realized by evaluating its procedure. Such a representation has advantages of power and generality but provides too little structural framework for specifying interactions among processes (recent development of languages such as MICRO-PLANNER [9] and QA4 [7] may now make this approach more feasible).

Borrowing from automata theory, we decided to represent each process or concept as an augmented finite-state automaton. We supplemented the normal notion of a finite-state automaton by associating predicates with each allowable transition (such as was done by Woods [10]) and by adding to each transition text generative and computational information. As we shall see, the augmented finite-state automaton provides a reasonable structure for describing a process and its interactions with the "world" and other processes. It also has the virtue of permitting simple "generate and search" techniques for inferencing.

***** Insert table 1 *****

A simple automaton, that for the Rate of Evaporation, is detailed in TABLE 1. This automaton has three possible states: (A) decreasing, (B) stable, and (C) increasing. Each row in the transition table concerns a particular transition between two states of the automaton. The first column specifies a particular transition between two given states. The second column, "External Conditions," specifies conditions which must be satisfied over variables or flags external to the model before the transition can occur. The column marked "Internal Conditions" specifies constraints concerning which states other automata must be in before this automaton can exercise the given transition. Both the external and internal constraints take the form of arbitrary LISP predicates which must evaluate to "True" before the transition can occur. Usually these predicates are simple Boolean functions which check the states of other automata. They can be, however, as complicated as is necessary to capture the subtleties of interactions between two given concepts or processes.

The "Computational Column" specifies how a parameter of the automaton is to be computed if the query requires a numeric answer. This entry is either the name of a table or the name of a computational procedure along with information specifying the binding of arguments for this procedure. For

the case in TABLE 1, the numeric value associated with the automaton is the value of the rate of evaporation in millimeters of mercury. That value is determined by interpolation from a stored table (rate of evaporation is a function of temperature) and is altered if there are hygroscopic (water attracting) nuclei present.

The last column, "Text Generation", specifies a fragment of text that can be generated in conjunction with the automaton making this transition. We will later see how these fragments can be synthesized into a paragraph or complex sentence in response to a question.

***** Insert table 2 *****

The cross product of all the automata in the system forms a global automaton which functions as the dynamic model of our meteorology data base. Currently, the model contains 20 automata representing several billion different possible global states (not all of these are accessible from reasonable starting states of the model). TABLE 2 provides a list of the automata and Appendix 1 contains complete descriptions of some of these. One can easily augment this model by adding new automata and specifying via the internal and external conditions (columns) how the new automata

interface with the old.

Problems of Context

Students using this system invariably formulate questions which assume a certain amount of contextual information. Questions are not formulated in a vacuum, and usually involve numerous presuppositions even apart from contextual considerations. For example, if a student asks: "What happens when the temperature drops?", he may be assuming that there is some water vapor present or perhaps some other objects whose attributes might be altered. In certain contexts, it would be clear in what range the temperature was before it "dropped". Or, depending on previous assertions, the student may be assuming that "temperature" means "air temperature" and presupposing that air temperature is always equal to water temperature (of the water droplets in the clouds). Obviously, it would be unreasonable to expect students to make all such assertions explicit. A useful system for a CAI environment must somehow keep track of at least part of the underlying context. In other words, some "model", however simple, of the student's current assumptions must be maintained throughout the session.

Towards this end, we have included an external state vector, or E vector, which reflects various implicit assumptions. The E vector is a sequence of flags and parameters which are altered in accordance with the kind of material the student is currently exploring. At present, these flags have no associated transition tables; their states may be changed only by direct respecification.

***** Insert table 3 *****

TABLE 3 specifies some of the parameters included in the E vector. These parameters are referenced by the predicates in the "External Condition" columns of the transition tables (TABLE 1). In the absence of any information, default conditions are specified which are set to evoke reasonable behavior from the model.

Question Answering With the Model

Question answering with the process model is accomplished via the following procedure. The global automaton and selected components of its E vector are reset to their initial (default) state. The question is parsed and interpreted, and a command is established to "twitch"

the global automaton by specifying changes in the E vector or by forcing a particular transition in one of the automata. Once a change occurs in one of the flags or automata, those predicates in the model which could be affected by the change are evaluated to see if their respective automata can exercise the associated transition. The model is cross referenced by transition conditions so that the relevant set of automata can be quickly determined.

This process is repeated recursively until no new transitions occur. An inference tree is thus developed (see TABLE 4) which reflects all the possible global state changes which are a result of the original "twitch".

***** Insert table 4 *****

An additional piece of information is necessary in order to control the construction of the inference tree. As a casual inspection will reveal, the process model as it is now described would be very prone to infinite looping. No automaton could ever be allowed to return to a previous state unless it could be guaranteed that some condition, which allowed the original transition, had been changed. One way of coping with this would be to selectively reset the automata whenever cyclic behavior is detected.

A closer examination reveals a natural division of the automata into two classes. The first of these consists of automata which represent mathematical variables or relations, e.g. [IV] and [IA] ("Relative Humidity" and "Humidity of Saturation"). These entities exhibit instantaneous change in response to sudden respecification of their determining variables. Secondly, there are those - e.g. [III] ("Process") - that directly depict conditions or events which are perceived in the medium of time and intrinsically possess the property of duration.

We have attached to each automaton a "time flag" that specifies to which class it belongs. Whenever an automaton of the second class exercises a transition, all the automata of the first class (and various flags which represent physical parameters) are reset to their initial values. Further, only automata of the second class are permitted conditional transitions which lead away from non-initial states. Provided that the conditions for these "reset" transitions are properly specified, we are assured that the process model will never exhibit looping behavior. Of course oscillatory behavior could be purposely evoked in order to model some types of processes.

Besides eliminating unwanted cyclic behavior, this timing information assists us in organizing the growth of

the inference tree. Addition of further type two nodes to the tree is delayed until no further class one additions are possible. The expansion of the tree is thus divided into alternating periods of type one and type two growth. The result is that all the changes of state of the global automaton which can be thought of as occurring together in time are completely explored before the next "time period" is entered. The timing information is saved in the tree and is used to mark paragraph boundaries in the text generation phase.

The inference tree thus developed acts as a "deep structure" for the generation of the explicative sentences or paragraphs which form the response to a query. Each node in the tree represents one action in a causal chain of events. The structures taken from the node include the transition and the reasons it occurred. At this stage only the crudest text generator is invoked to transform this inference tree into English. This can easily be seen by examining the sample output (Appendices 2 and 3). Nevertheless one can see that enough information is present to permit a good natural language "rear end" system to do reasonable text generation (the words surrounded by parentheses in these examples are lifted directly from the "text" column in the appropriate transition table).

Suppose that the question pursued in TABLE 4 had instead been phrased:

Does Relative Humidity decrease when the temperature drops?

The tree would be searched for a node (global state) that corresponds to a decrease of the Relative Humidity automaton. Location of such a node causes a "Yes" answer.

Handling "Why" type questions follows directly from the above considerations. Suppose we asked:

"Why does the Rate of Condensation decrease when the water temperature decreases?"

To answer this question we pick up the phrase "when the water temperature decreases" and "twit" the model in accordance with it. By forcing the Water Temperature automaton to decrease, the tree in TABLE 4 is again generated. We now scan the tree to find a node which reflects the appropriate transition of the Rate of Evaporation automaton and then construct a response from the chain which arises from pursuing a path from the root of the tree to that node.

Because the generated tree is preserved until the next question is posed, the following type of dialogue is always possible:

Q: Does x cause y to happen?
R: Yes.
Q: Why?
R: Because ...

In addition to these purely qualitative questions, quantitative queries can be likewise handled. Suppose, instead of asking the qualitative question:

Q: What happens when the Water Temperature drops?

We ask the question:

Q: (What happens when the Water Temperature drops to 22 degrees)

R: THE (AIR TEMPERATURE) (DECREASE) FROM 25 TO 22 DEGREES BECAUSE (WATER TEMPERATURE) (DECREASE). THE (HUMIDITY OF SATURATION) (DECREASE) FROM 24 TO 20 MM HG BECAUSE (AIR TEMPERATURE) (DECREASE)...

(See Appendix 2 for the entire three paragraph response.)

Again, the constructed response is open to further interrogation by the student. For example, he might ask how we arrived at the value for the Humidity of Saturation of 20 mm. Hg. For such a request we would access that column of the appropriate transition table and see what technique is used for that particular computation. If it is a table look-up, then the student can request the table to be printed. If it is a procedure, then the student can ask to

have the formula displayed, to insert his own values into the procedure or to have a graph drawn of the functional values produced by the procedure.

Appendix 2 contains several examples of actual output from our system. For further sample output covering a much wider range of meteorological processes, the reader is referred to Brown et al. [2]

Representation of Factual Information

Of course, not all questions require the above kind of treatment and not all "knowledge" can be conveniently encoded in terms of automata. We therefore need another specialized data structure to represent the kind of information we believe is important to an understanding of meteorology but which is not easily captured by the above methods. In addition, certain information about "processes" (e.g., their definition) is inherently factual, rather than functional, and hence can best be represented externally to the transition tables. In light of these considerations, a semantic/conceptual network, similar to those developed by R. Quillian [6] and R. Simmons [8], was designed.

Our network possesses three basic kinds of nodes, each with slightly different property list structures. Since any

node within a given generic class has the same kind of properties, we can take advantage of this uniformity in writing specialized routines to perform a limited class of inferences. Nevertheless, the combination of this network and its referencing procedures does not afford the flexibility of a general purpose question answering system. In order to expand the range of questions that the system can handle, the user must adjoin new inferencing procedures to the executive which processes this network.

The three kinds of nodes in this semantic/conceptual network are:

- 1) Objects (e.g. Rain, Ice, Snow, ...)
- 2) Concepts (e.g. Temperature, Pressure, Freezing Point, ...)
- 3) Processes (e.g. Evaporation, Precipitation, Saturation, ...)

Objects are interrelated by two basic ordering relationships; e.g., x "is a kind of" y, or x "is an example of" y. These relations induce a partial ordering over the set of objects and function in the standard way by allowing the object x to inherit properties of the object y as specified by the given inference procedures. Each object has the following properties (a property can have sub-properties):

- 1) DEF: an ASCII string giving the object's definition - key words in the string are subject to cross-referencing as described later.
- 2) PATT: Physical attributes, such as shape, size (bounds), color, weight, etc.
- 3) BC: Boundary conditions for the object's
 - a) formation
 - and b) existence.
- 4) SUB: What things are examples of this object.
- 5) EX: What things this object is an example of.
- 6) LOC: Where the objects are usually found or contained.
- 7) WAFF: What things the object affects (which process automata this object affects)
- 8) UF: What things the object is used for (which process automata this object is involved in)
- 9) WAFFT: What affects the object's attributes.
- 10) TXT: A short exposition (coded simply as text) about this object.
- 11) SCH: A schema used by the CAI executive to specify an ordering over the other properties to handle general requests for information such as: "Tell me something about x".
- 12) XREF: Contains cross-reference information
- 13) SYN: Any synonyms that are also contained in the network

The network structures for processes and concepts have many properties in common with that of objects, but contain some other important properties. For example, the list structures for a process contain the property XFORM, which

specifies what objects are transformed by the process as well as their states before and after the transformation. The most important property that all three nodes have in common is "XREF". This property specifies which other nodes in the network contain properties that reference (mention) this node and also which properties these references occurred under. Therefore, when a procedure commences its search from a particular node, it can selectively pursue certain paths determined by how that node is used or referenced by other nodes.

Static Question Answering

Many questions can be answered directly from the semantic/conceptual network. In fact, the network is always consulted first, and only when no responses can be found is the process model used. Limited inferences can be obtained directly from the semantic/conceptual network by a variety of simple routines. For example, when asked:

Q: What happens to ice when it melts?

it is noted that:

- a) Ice is an example of solid water
- and b) To melt means to transform solid water into liquid

water.

Via a simple inferencing procedure which interrelates the SUB relation "is an example of" with the XFORM property list we obtain the response:

R: Liquid water.

A slightly more sophisticated example is:

Q: Is it true that fog is a form of precipitation?

Once parsed, a command is set up to access the semantic/conceptual network at the node representing "Precipitation" and a search is made on its value list attached to the property "SUB" (examples of). If there are some items under "SUB", but "Fog" is not one of them, the "SUB" property of each of these items is checked. When "Fog" is not encountered in any "SUB" chain under precipitation, the cross reference list for Fog is searched for a node which references Fog as a value to the property "EX" (those things of which Fog is an example). If such a cross reference exists, that node is processed accordingly. And in this case our system would respond:

R: No, fog is an example of condensation.

If this cross reference search fails, then a search is made

for any object which references it via an "is a kind of" or "is an example of" relation and so on.

The semantic/conceptual network encodes a great deal of information in a sufficiently systematic way that simple inferencing procedures can operate on the network to answer many of the questions that naturally arise in a CAI environment. Again, we stress that our goal is not to build another "deep" (with respect to its logical capabilities) question answering system, but to explore a pragmatic system carefully tuned to the particular requirements of mixed initiative CAI. As we encode still more information in the network and expand the set of inferencing procedures that operate on it, we expect to discover better ways to structure this network.

Combined Inferencing Techniques

Earlier in our discussion of inferencing with the dynamic model we discussed the handling of "Why" type questions. Our method of answering a question of the form "Why does x when y?" was to find the path from y to x in the generated tree. There is one obvious case where this strategy is unsatisfactory. Suppose the node being sought

is adjacent to the root of the tree. In this case, no information is gained from the chain. When this happens, the semantic/conceptual network is accessed for explanatory information on that particular transition. This information can be a simple explanatory note or in special cases, another automaton. This automaton may be used to generate the explanatory note and may be investigated further by the student. For example, if the student wants to know why the Rate of Evaporation increases when the Water Temperature increases (see TABLE 1 for condition), an automaton model of thermo-dynamics and molecular interactions could be called upon to explain this transition. After explanation of the transition, the molecular model is available if the student wishes to continue his investigation along these lines. In this way, hierarchies of global automata can be created for exploration by the student.

Further Plans

The above described question answering system has been utilized in a prototype CAI system which permits open interrogation of both the semantic/conceptual network and the dynamic model. Thus far we have not refined the host CAI system or the simple key word natural language parser which transforms the student's requests into commands

understandable by our question answerer. We are currently investigating a restricted version of Woods [10] network parser for this purpose. We are also studying a class of questions which require that our dynamic model be "run" backwards. For example, we would like to be able to handle questions of the form: "Under what conditions can "x" occur?"

Conclusion

We feel that we have just begun to realize the potential of representing processes through these highly specialized automata-like structures. This representation provides a natural method for describing processes and allows efficient inferencing about them. Of special interest is the helpfulness of this structure in synthesizing complex natural language responses. We believe that these structures partially provide a semantic representation of "causal" structures that is prerequisite to meaningful discourse. We are especially interested in the combined use of two divergent representations of knowledge (i.e. the semantic/conceptual network and the automata) where each modeling structure is especially well suited for handling certain classes of questions.

We were surprised by the relative ease of adding new information to the system and by the amount of knowledge that we finally got encoded into it. With hindsight, we realize how fortunate the decision was to implement the external and internal conditions for each automaton as LISP predicates instead of restricting ourselves to some Boolean calculus. With the flexibility afforded us by being able to write arbitrarily complex conditions on certain transitions, we could easily add to and modify the automata until they behaved as desired. Of course, the end set of conditions is not as parsimonious as might be hoped, but then we were not after a theory of meteorology but rather a pragmatic question-answering system that could successfully function in a CAI environment.

References

1. Bobrow, Robert J., Richard R. Burton and Daryle Lewis, "UCI LISP Manual", Technical Report No. 21, Department of Information and Computer Science, University of California Irvine, September 1972
2. Brown, John Seely, Richard R. Burton and Frank Zdybel, "A Model Driven Question-Answering System for a CAI Environment," Technical Report No. 13, Department of Information and Computer Science, University of California Irvine, January 1972
3. Carbonell, Jaime, "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction", IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, December 1970, pp. 181-189
4. Carbonell, Jaime, "Mixed-Initiative Man-Computer Instructional Dialogues," Ph.D. dissertation, M.I.T., June 1970
5. Nilsson, Nils J., Problem-Solving Methods in Artificial Intelligence, McGraw-Hill Book Company, 1971
6. Quillian, M. Ross, "Semantic Memory," in Minsky (ed.) Semantic Information Processing, The M.I.T. Press, 1968, pp. 216-270
7. Rulifson, J. F., R. J. Waldinger, and J. A. Dirksen, "QA4, A Language for Writing Problem-Solving Programs," Proceedings of the IFIP Congress, North-Holland Publ. Co., Amsterdam, 1971, pp. 111-115
8. Simmons, Robert, "Natural Language Question-Answering Systems: 1969," Communications of the ACM, Vol. 13, January 1970, pp. 15-30
9. Winograd, Terry, "Computer Programs For Understanding Natural Language," MAC TR-84, Project MAC, M.I.T., February 1971
10. Woods, William, "Transition Network Grammars for Natural Language Analysis," Communications of the ACM, Vol. 13, October 1970, pp. 591-606

APPENDIX 1

PARTIAL DESCRIPTION OF THE AUTOMATON MODEL

The automata described here are those used in answering the example questions in the text. The model also includes cloud formation, the ice crystal process, hail formation, etc. A description of the Coalescence process automaton is included as an example of a more complex automaton. For further information about the model, see Brown et al. [2]

Following the description of each automaton is a list of its states and the possible transitions which may occur between these states. The computation and default value are given for those automata which have quantitative parts associated with them.

Notational notes: Interpret square brackets ([]) to mean "state of"; thus, read [IA] as "the state of the Humidity of Saturation". Interpret opposed double angle brackets (<<>>) to mean "the value of"; thus read <<A.T.>> as "the value of the Air Temperature", meaning the numerical value of this variable rather than the state of the automaton which represents changes in this value. Terms or disjuncts of transition conditions which are flagged with an "*" are to be suppressed from output generated by the model in order to eliminate redundant or implicit requirements from reaching the student.

CENTRAL AUTOMATA

Central automata are those which are employed in the solution of container problems and in the preliminary considerations of the formation of airborne particles (clouds).

HUMIDITY OF SATURATION - The absolute humidity at which the rate of condensation should just equal the rate of evaporation, assuming the air temperature to be equal to the water temperature and the absence of hygroscopic nuclei or other mechanisms which would interfere with the rate of evaporation. This value is a dependent variable of the air temperature.

[IA] (HUMIDITY OF SATURATION)

A = "DECREASES"

B = "STABLE"

C = "INCREASES"

B=>A ([A.T.]=A)

B=>C ([A.T.]=C)

COMPUTATIONAL NOTE:

<<IA>> = TABLE(<<A.T.>>)

RATE OF EVAPORATION - The rate at which liquid water is being converted into water vapor, measured at the boundary between the liquid water and the air, expressed as a pressure. This quantity is essentially the same as the vapor pressure of water measured over the same boundary.

[IB] (RATE OF EVAPORATION)

A = "DECREASES"

B = "STABLE"

C = "INCREASES"

B=>A ([W.T.]=A) OR ([N] X=>B IF X NOT EQUAL TO B)

B=>C ([W.T.]=C) OR ([N] B=>X IF X NOT EQUAL TO B)

COMPUTATIONAL NOTE:

<<Q>>=1 IF ([N]=B)

<<Q>>=0 IF ([N] NOT EQUAL TO B)

<<IB>> = TABLE(<<W.T.>>)-<<Q>>*1.5*TABLE(<<W.T.>>)

ABSOLUTE HUMIDITY - The partial pressure of water vapor in the atmosphere at the site of the described process.

[IIA] (ABSOLUTE HUMIDITY)

A = "DECREASES"

B = "STABLE"

C = "INCREASES"

B=>A ([B.P.]=A) OR (([III]=C) AND ([P]=C)*)

B=>C ([B.P.]=C) OR (([III]=A) AND ([P]=C)*)

COMPUTATIONAL NOTE:

<<IIA>> = <<IB>> IF [IIA] TRIGGERED BY [III]

<<IIA>>' = (<<B.P.>>' / <<B.P.>>)*<<IIA>>

IF [IIA] TRIGGERED BY [B.P.]

(' MEANS NEW VALUE)

RATE OF CONDENSATION - The rate at which water vapor is returning to the liquid state measured as a pressure upon the air-water boundary. The value is essentially equal to the partial pressure of water vapor in the air.

[IIB] (RATE OF CONDENSATION)

A = "DECREASES"

B = "STABLE"

C = "INCREASES"

B=>A ([IIA]=A)

B=>C ([IIA]=C)

COMPUTATIONAL NOTE:

<<IIB>> = <<IIA>>

PROCESS - This graph describes the imbalance between the rates of evaporation and condensation as net evaporation, net condensation or equilibrium.

[III] (PROCESS)

A = "NET EVAPORATION"

B = "EQUILIBRIUM"

C = "NET CONDENSATION"

B=>A ([IB]=C) OR ([IIB]=A)

B=>C ((([IB]=A) OR ([IIB]=C)) AND [N]≠B)

A=>B ([IIB]=C)

C=>B ([IIB]=A)

RELATIVE HUMIDITY - The absolute humidity divided by the humidity of saturation, the quotient being expressed as a percentage.

[IV] (RELATIVE HUMIDITY)

A = "DECREASES"

B = "STABLE"

C = "INCREASES"

B=>A ((([IA]=C) OR ([IIA]=A)) AND [G]=B)

B=>C ((([IA]=A) OR ([IIA]=C)) AND [G]=B)

COMPUTATIONAL NOTE:

<<IV>> = (<<IIA>>/<<IA>>)*100

AIR BECOMES - This graph describes the air as being either saturated, unsaturated or super-saturated, depending on how the rates of condensation and evaporation compare.

[V] (AIR BECOMES)

A = "UNSATURATED"
 B = "SATURATED"
 C = "SUPER-SATURATED"

B=>A (([IB]=C) OR ([IIB]=A)) AND ([G]=B)
 B=>C (([IB]=A) OR ([IIB]=C)) AND ([G]=B)
 A=>B ([III]=A)* AND ([IIB]=C)
 C=>B ([III]=C)* AND ([IIB]=A)

COALESCENCE PROCESS AUTOMATON

COPROG - This graph describes the various conceptual stages of the coalescence process. Various states in this process chain are individually devoted to examinations of the physical forces which drive the Coalescence Process and of certain requirements which must be satisfied for this process to occur.

[COPROG] (PROGRESS OF COALESCENCE PROCESS)

A = "PROCESS DOES NOT OCCUR"
 B = "LARGE AND SMALL WATER DROPLETS FORM"
 C = "LARGE AND SMALL WATER DROPLETS FALL AT DIFFERENT SPEEDS"
 D = "COLLISIONS OCCUR BETWEEN DIFFERENT SIZE WATER DROPLETS"
 E = "COALESCENCE OCCURS"
 F = "PRECIPITATING DROPLETS RESULT"
 G = "FRACTIONATION MAY OCCUR"

A=>B ([NWD]=T)* AND ([G]=B) AND ([E]=T)* AND ([C]=D)*
 B=>C (FORCED)*: "THE LARGER THE DROPLET, THE GREATER THE EFFECT OF GRAVITY RELATIVE TO THE EFFECT OF AERODYNAMIC DRAG."
 C=>D ([COPROG]=C)
 D=>E ([E]=T)
 E=>F ([C]=D)
 F=>G (FORCED ON FIRST ENCOUNTER OF STATE "F")*:
 "THE DROPLETS MAY GROW BEYOND 7 MM. IN DIAMETER, AT WHICH POINT AERODYNAMIC DRAG MAY PULL THEM APART."

APPENDIX 2

CONTAINER MODEL QUESTIONS

Sample output from questions asked of the system while dealing with the processes of meteorology in a closed environment. (Note the first two questions are quantitative while the last is purely qualitative.)

Q: (WHAT HAPPENS WHEN THE WATER TEMPERATURE DROPS TO 22 DEGREES)

THE (AIR TEMPERATURE) (DECREASE) FROM 25 TO 22 DEGREES BECAUSE (WATER TEMPERATURE) (DECREASE). THE (HUMIDITY OF SATURATION) (DECREASE) FROM 24 TO 20 MM HG BECAUSE (AIR TEMPERATURE) (DECREASE). THE (RELATIVE HUMIDITY) (INCREASE) FROM 100 TO 120 % BECAUSE (HUMIDITY OF SATURATION) (DECREASE) AND (GRADIENT OF CHANGE) (SUDDEN). THE (RATE OF EVAPORATION) (DECREASE) FROM 24 TO 20 MM HG BECAUSE (WATER TEMPERATURE) (DECREASE).

THEN THE (AIR CONDITION) (SUPERSATURATED) BECAUSE (RATE OF EVAPORATION) (DECREASE) AND (GRADIENT OF CHANGE) (SUDDEN). THE (PROCESS) (NET CONDENSATION) BECAUSE (RATE OF EVAPORATION) (DECREASE) AND (NUCLEI) (HYGROSCOPIC NUCLEI ARE PRESENT). THE (ABSOLUTE HUMIDITY) (DECREASE) FROM 24 TO 20 MM HG BECAUSE (PROCESS) (NET CONDENSATION). THE (RELATIVE HUMIDITY) (DECREASE) FROM 120 TO 100 % BECAUSE (ABSOLUTE HUMIDITY) (DECREASE) AND (GRADIENT OF CHANGE) (SUDDEN). THE (RATE OF CONDENSATION) (DECREASE) FROM 24 TO 20 MM HG BECAUSE (ABSOLUTE HUMIDITY) (DECREASE). THE (AIR CONDITION) (SATURATED) BECAUSE (RATE OF CONDENSATION) (DECREASE). -

THEN THE (PROCESS) (EQUILIBRIUM) BECAUSE (RATE OF CONDENSATION) (DECREASE).

Q: (WHAT HAPPENS IF THE AIR TEMPERATURE DECREASES TO 20 DEGREES AND THERE ARE NO CONDENSATION NUCLEI)

THE (HUMIDITY OF SATURATION) (DECREASE) FROM 24 TO 17 MM HG BECAUSE (AIR TEMPERATURE) (DECREASE). THE (RELATIVE HUMIDITY) (INCREASE) FROM 100 TO 180 % BECAUSE (HUMIDITY OF SATURATION) (DECREASE) AND (GRADIENT OF CHANGE) (SUDDEN). THE (WATER TEMPERATURE) (DECREASE) FROM 25 TO 20 DEGREES BECAUSE (AIR TEMPERATURE) (DECREASE). THE (RATE OF EVAPORATION) (DECREASE) FROM 24 TO 17 MM HG BECAUSE (WATER

TEMPERATURE) (DECREASE).

THEN THE (AIR CONDITION) (SUPERSATURATED) BECAUSE (RATE OF EVAPORATION) (DECREASE) AND (GRADIENT OF CHANGE) (SUDDEN).

Q: (SUPPOSE THE PRESSURE DECREASES GRADUALLY)

THE (ABSOLUTE HUMIDITY) (DECREASE) BECAUSE (BAROMETRIC PRESSURE) (DECREASE). THE (RATE OF CONDENSATION) (DECREASE) BECAUSE (ABSOLUTE HUMIDITY) (DECREASE).

THEN THE (PROCESS) (NET EVAPORATION) BECAUSE (RATE OF CONDENSATION) (DECREASE). THE (ABSOLUTE HUMIDITY) (INCREASE) BECAUSE (PROCESS) (NET EVAPORATION). THE (RATE OF CONDENSATION) (INCREASE) BECAUSE (ABSOLUTE HUMIDITY) (INCREASE).

THEN THE (PROCESS) (EQUILIBRIUM) BECAUSE (RATE OF CONDENSATION) (INCREASE).

APPENDIX 3

SAMPLE OUTPUT DIALOGUE

Q: BEGIN LESSON ONE

In order to understand precipitation, we must first understand about water vapor. A small fraction (about 4 %) of the atmosphere is water vapor. Under normal conditions, tiny increases in this small percentage can result in precipitation. Water molecules can assume three different physical states: gaseous, liquid and solid.

Q: WHAT IS PRECIPITATION

PRECIPITATION IS (WATER IN LIQUID OR SOLID STATE WHICH FALLS FROM THE SKY).

Q: WHAT ARE SOME EXAMPLES OF PRECIPITATION

RAIN, SLEET, HAIL AND SNOW

Q: IS FOG A FORM OF PRECIPITATION

NO

Q: WHAT IS FOG

FOG IS (CLOUDS WHICH ARE FOUND AT VERY LOW ALTITUDES; USUALLY AT LESS THAN A FEW HUNDRED FEET ABOVE GROUND LEVEL).

Q: TELL ME ABOUT WATER VAPOR

WATER VAPOR IS (WATER WHICH IS IN THE GASEOUS OR VAPOR STATE).

TABLE 1
AUGMENTED TRANSITION TABLE
FOR
RATE OF EVAPORATION AUTOMATON

TRANSITION	EXTERNAL * CONDITIONS	INTERNAL * CONDITIONS	* COMPUTATION	TEXT GENERATION
B => A	[N]=A,C,D => [N]=B	[WT]=A	IF NB="TRUE" THEN (IB=.85 * TABLE(WT)) ELSE IB=TABLE(WT)	DECREASE
B => C	[N]≠B => [N]=A,C,D	[WT]=C	- -	INCREASE

The second line is read: "The Rate of Evaporation decreases if the Water Temperature decreases". The other symbols used are: [N] for the state of Nuclei (see TABLE 3) and [WT] for the Water Temperature automaton (see Appendix 1).

* These columns contain arbitrary LISP predicates.

TABLE 2

A List of the Process Automaton in the Complete Meteorology Model

[IA]	Humidity of Saturation
[IB]	Rate of Evaporation
[IIA]	Absolute Humidity
[IIB]	Rate of Condensation
[III]	Air-Water Process
[IV]	Relative Humidity
[V]	Air Condition
[ICPROG]	Ice Crystal Process
[HFPROG]	Hail Formation Process
[FRPROG]	Cloud Freezing Process
[COPROG]	Coalescence Process
[PRECIP]	Form of Precipitation
[NET]	Processes During Ice Crystal Process
[SSC]	Size of Super-cooled Water Droplets
[SIC]	Size of Ice Crystals
[NSC]	Number of Super-cooled Water Droplets
[NIC]	Number of Ice Crystals
[SPI]	Number of Small Droplets
[LPI]	Number of Large Droplets
[LPII]	Size of Large Droplets

TABLE 3

SAMPLE OF EXTERNAL STATE VECTOR

(COMPLETE STATE VECTOR HAS 16 COMPONENTS)

[S] (WATER TEMPERATURE ALWAYS EQUALS AIR TEMPERATURE)

* T = "TRUE"(ALWAYS TRUE WHEN DEALING
WITH WATER IN A CLOUD)

F = "FALSE"

[N] (NUCLEI)

A = "NON-HYGROSCOPIC NUCLEI PRESENT"

* B = "HYGROSCOPIC NUCLEI PRESENT"

C = "NO SURFACES OR NUCLEI PRESENT"

D = "SURFACES PRESENT"

[G] (GRADIENT OF CHANGE)

A = "GRADUAL"

* B = "SUDDEN"

[C] (CLOUD DEPTH)

* D = "DEEP"

S = "SHALLOW"

[P] (PROCESS TYPE)

* C = "CONTAINER"

M = "METEOROLOGICAL"

PHYSICAL PARAMETER DEFAULT VALUES

AIR TEMPERATURE = 25 DEGREES CENTIGRADE

WATER TEMPERATURE = 25 DEGREES CENTIGRADE

BAROMETRIC PRESSURE = 760 MM OF MERCURY

(* INDICATES DEFAULT VALUE)

TABLE 4

E Vector =

[G]=B
'Gradient of Change is Sudden'

 ,

[N]=B
'Hygroscopic Nuclei are Present'

 ,

[P]=C *
'Process is Occurring in a Container'

 ,

[S]=T *
'Water Temp. Equals Air Temperature'

 , ...

Inference Tree

