We Must RE-MEMBER to RE-FORMULATE: The M System

Doug Riecken
IBM Watson Research Center and Rutgers University
riecken@us.ibm.com riecken@home.com

1 Introduction

This paper provides introductory discussion on the M system. M is a study of an architecture that supports integrated multiple reasoning processes and representations. This architecture has been applied and evolved through a series of different domain problems:

1. Wolfgang, a system that learns to compose music (Riecken 1989, 1992a),
2. adaptive user interfaces (Riecken 1991a, 1991b, 1992b), and
3. the M system (Riecken 1994), a software program that acts as an assistant to a user by classifying and managing domain objects in a multimedia conferencing system.

The goal of this work is to develop a theory of mind that enables common sense reasoning to be applied in M. M was designed to observe situations and formulate beliefs about these situations regardless of the truth of the beliefs. It appears that humans observe and believe, and then over time continue to improve their knowledge of their beliefs while many types of computer programs just get stuck!

I take the position that common sense learning is a time variant problem and that learning is a constant series of viewing similar situations from different points of view over time. You can not learn something until you learn about "it" from several points of view. Good common sense reasoning and learning results from the ability to perform reformulation on an idea, concept, or problem. Reformulation requires rich fluid representations, multiple modalities of reasoning, and a range of experiences over time. Minsky's Society of Mind (SOM) Theory (Minsky 1985) is the essence of the study and implementation of M.

M was initially designed and implemented in my work at AT&T Bell Laboratories. M functioned as a software process that recognized and classified objects and actions performed by humans in a multimedia desktop conferencing system we developed at Bell Labs, called the Virtual Meeting Service. In this Computer Supported Cooperative Work (CSCW) service, participants worked in a Virtual Meeting Room (VMR) on a series of tasks. Each participant has a personal M program that watches all legal actions performed on all legal objects in the VMR by all participants. Each personal M attempts to recognize and classify what all the participants are doing and then support its respective user to recall items and actions that might relate to the user's current task and context.

2 VMR as a CSCW environment

To design a prototype model that can perform CSCW classifier functions, a specific CSCW environment was identified. Based on my previous work at Bell Laboratories in developing AT&T's Virtual Meeting Service, I defined the CSCW environment on the idea of a virtual meeting room (VMR) that supports multimedia desktop conferencing.

In a VMR, participants collaborate via computers and shared applications that provide users with documents, whiteboards, markers, erasers, staplers, copy machines, and many other such objects. The actual VMR is a complex set of data structures hosted on a server platform that maintains a consistent state view of a VMR session for all legal participants.

Conceptually, a VMR is a virtual place where one or more persons can work together even though the individuals are physically separated. An example of a VMR is a computer hosted place where individuals physically located in New Jersey and England can meet and work. In a VMR, the individuals share and create information in a variety of media ranging from text to images to drawings.

VMRs also support the functionality of persistence, thus VMRs can exist over arbitrarily long periods of time. A VMR is like a real meeting room where individuals can work, leave at the end of a day while leaving behind all documents and other objects, and then return at a later point in time to continue the work at hand.

3 The M system

The M system is a computer model (program) that performs classification tasks in a VMR. M is a system that applies "common sense" reasoning and knowledge to formulate classifications of VMR domain objects. M's reasoning does not rely on the content contained in VMR objects (e.g., documents), but instead M observes simple contextual cues and features present in typical VMR situations. Simply put, M reasons based on context, not content.

The power of M's "common sense" reasoning results from integrating "simple" facts and rules asserted from different lines of reasoning. M's model is a collection of simple facts and ideas about user collaboration in a VMR.

In order to develop a theory of common sense reasoning, I have studied and designed systems that support multi-reasoning processing. This appears to be essential in that the common sense "things" we understand as humans results from integrating many very simple, sometimes trivial, pieces of information about the world around us. A good theory on common sense reasoning might require that reasoning integrate information based on such distinct views as time, space, and function. To examine such a theory, we first must select a "world" in which to perform common sense reasoning.

The VMR world is a much simpler world than our own physical world. So, in order to better understand how to make use of many very simple facts, some which we use all the time without realizing, I have continued my study via the VMR world. The VMR world is an explicit finite problem space in which a formal representation of the useful information might provide a better understanding on how a system, biological or in silicon, might be able to reason about objects and actions within a VMR.

The M system is a multi-strategy classifier system architecture contains the following:

- semantic net functions
- rule-base system
- scripting system
- multi-ranked blackboard system based on Minsky's Trans-Frames in SOM

The design of M must enable M to function as a useful assistant to a human user. This implies that M's classification and knowledge of users working in a VMR must appear to a user to make sense from the user's point of view. Thus M must reason in a manner consistent with the user.

4 Function of M

In a VMR, each user is supported by a personalized M assistant and the VMR world is composed of domain objects (e.g., electronic documents, electronic ink, images, markers, white boards, copy machines, staplers, etc.) upon which users apply actions. The M assistant(s) attempt to recognize and define relationships between objects based on the actions applied by the users to the VMR world and the resulting new states of that world. For example, in a VMR world, there may exist a set of domain objects – such as several documents. Further, the VMR participants may apply actions to these documents such as annotating over them collectively or joining/appending them together. M attempts to identify all domain objects and classify relationships between subsets of objects based on their physical properties and user applied actions.

5 Simple example

A simple example would be 2 adjacent documents which a user then annotates by drawing a circle to enclose them together. Thus based collectively on (1) spatial reasoning of the nearness of the 2 documents and the circle, (2) structural and functional reasoning of the circle enclosing the 2 documents, and (3) casual reasoning of the semantic action of enclosing objects – M can infer and explain a plausible relationship between the 2 documents.

6 Organizing the VMR workspace

Consider a typical group of designers working in a brainstorming session held within a real physical room. By the end of such a working session, the designers will have created and used many documents, bullet lists, diagrams, notes, post-its, and other such items. Based on the properties of a physical room, the participants could organize themselves and the objects in the room using tables, walls, and whiteboards. Documents and other objects could be spatially organized and located for ease of access by the meeting participants. Typically, the designers would be able to view, engage, review, and reformulate various conceptual relationships over all the physical materials and information generated as the meeting progressed.

When we port the designer's brainstorming session to a VMR, their view of the work environment is significantly constrained to the physical size of their respective computer screens (e.g., ˜ 1000x1000 pixels at best). What if M took on the responsibility to organize the output and interactions of all the participants? In essence, M assists a user to access and manipulate many different materials created and used during a meeting, independent of where the materials are located within a VMR or when the materials were last used or created.

M can generate and present various classifications representing conceptual views of VMR objects created and used by the participants. Thus, each participant can ask, via dialog boxes or direct manipulation techniques, their respective M assistant to present organized views of the various related materials used during a meeting.

Functionally, M observes the actions performed by VMR participants and attempts to reason how the current actions applied to VMR objects relate to other VMR objects and previous actions. As a participant interacts with an object, such as a document, M can provide the user with contextual hyperlinks to related objects, such as documents, drawings, notes, lists, post-its, and pen annotations. One of M's fundamental responsibilities is to assist a user to (RE)formulate relationships between all objects in a VMR.

Specifically, M attempts to maintain simultaneous theories of how objects in a VMR might relate. This enables M to provide participants with multiple views or access of related materials – thus, M and a user can reformulate the relationships between VMR objects.

While M maintains an extensive schema for organizing and representing a VMR, it must also allow the user to (RE)define existing and new relationships and hyperlinks within this schema. This safe-guards that M never takes control away from the user.

A useful idea in building a mind is the application of set theory and partial orderings as clever tricks to think about. Minsky's K-lines in SOM (Minsky 1980, 1985) are extensive sets of partial orderings of the enormous number of "facts and rules" that worked in previous situations and life experiences. The trick in these various learned ASSOCIATIONS is that they are members of various sets representing some learned idea, fact, concept, or process. Marvin has a wonderful play with words to remind us of this powerful idea. In Society of Mind, he writes the word remember as RE-MEMBER. We RE-MEMBER by using some members of a set of members that worked in some previous situation.

7 Design of M

The design of the M system required a formal world representation of a VMR. The world definition contained knowledge about all domain objects and the legal actions which can be applied within the world; the legal set of VMR situations.

The design goal of the M system was to recognize and classify actions and objects in a VMR world based on a "common sense" reasoning approach, instead of relying on "understanding" the content of the VMR objects via some form of natural language processing. In defining the ontology of M's knowledge-base, the following two tasks were required:

- develop a theory of the VMR recognition and classification process
- formulate a representation of the problem domain for all domain objects and actions

AI research has identified problem solving methods for ill-structured problems (Newell 1969, Simon 1973) were a set of heuristic processes generate a solution over a defined problem space. M's "common sense" reasoning relies on heuristics as it observes the world and applies contextual, not contentual, information about the objects and actions relating to a VMR situation. The design theory of M required a multi-strategy reasoning approach.

In a VMR situation, there are many simple and sometimes obvious cues which when combined together formulate a plausible theory of how objects relate. M integrates different reasoning processes which assert very simple facts into shared data structures representing the generation of a classification theory for a VMR situation. Presently, the M system is designed with five modal reasoning processes which collaborate to develop classification theories. The modalities of reasoning are: structural, functional, spatial, temporal, and causal.

8 M's recognition and classification process

M examines a VMR situation via the collaboration of distinct reasoning processes. The design theory for M partitions the problem solving process, the classification of VMR situations, into the following ordered sequence of functional tasks:

- represent a VMR situation consisting of an action, the pre VMR state prior to the action, and the post VMR state resulting from the action
- identify and characterize the object(s) involved in an action – this requires enumerating all known properties of the object(s)
- propagate the constraints relating to the object(s) and action to all reasoning processes responsible to classify the VMR situation
- have the reasoning processes collaborate to develop potential classification theories of the VMR situation
- restrict the range of plausible theories in order to avoid combinatoric growth

9 Ms' architecture

M's architecture consists of the following five key components representing knowledge of domain objects, legal actions, and legal situations: (1) a semantic net system, (2) a rule-based system, (3) a scripting system, (4) five distinct reasoning processes (inference engines) and (5) a blackboard system consisting of an ordered set of blackboards.

10 SEMANTIC NET SYSTEM

The semantic net system is implemented as a spreading activation network over sets of qualifiers (e.g., size, position, color, etc.) which collectively represent domain object characteristics. These qualifiers represent the facts associated with an applied action denoted in an input record. Each qualifier acts as a state machine representing the current legal property value of a VMR object. For example, the color qualifier can enter into a state representing the color of an object or a shape qualifier can enter into a state representing such shapes as square, circle, etc. The basic idea is this – when an object is identified via the I/O system, the corresponding qualifiers within the semantic net collectively become active representing the correct property states of the respective object. As these qualifiers become active in a specific state, they become facts which are asserted to M's rule-based system.

11 RULE-BASED SYSTEM

M's rule-based system performs several important functions. As facts (in the semantic net) are asserted, they in turn satisfy specific pre-conditions expressed in the antecedent of given rules. Thus, as the antecedents of such rules evaluate as true, this enables the consequence of each respective rule to be asserted. This can have the following two results. First, new facts expressed in a rule's consequence are asserted respectively to the semantic net; this then can have an iterative effect over the firing of new rules and the instantiation of other facts. Second, as new rules fire and new facts are instantiated, M's reasoning processes can in turn apply this new information to strengthen or weaken or create or purge the various theories representing a VMR world.

As various facts and rules evaluate as true, this directly influences M's scripting system and reasoning processes as they evaluate and apply various scripts of partial plans provided by the scripting system. In essence, we can view M's rule-based system as a collection of domain conditions that when satisfied are applied to bias the selection of partial plans from M's scripting system by M's reasoning processes to create and "explain" relationships between VMR objects.

12 SCRIPTING SYSTEM

M's scripting system is a corpus of partial plans that have demonstrated frequent success in previous classification problems. In M, a script is a partial ordering of elements in a set; the set represents an interval of time during which a consistent pattern of facts and rules have frequently been applied successfully to predict the state of some object(s) following some action. M's design of a script is based on Schank and Abelson's presentation of scripts (Schank 1977).

An important feature of M's scripting system entails the use of coefficients to weight each script's potential to either initiate or improve upon a theory which attempts to classify and represent some set of actions, objects, and relationships within a VMR. Functionally, these weighted scripts bias the various reasoning processes to dynamically rank all coexisting theories where each theory is formulated on one of the individual blackboards. These weighted scripts serve to minimize combinatoric growth of all possible classification theories. The reasoning processes will select weighted scripts that formulate or improve only the top seven ranked theories.

13 Multi-strategy reasoning

M's architectural design was based on a theory of integrated reasoning processes; sometimes referred to as integrated "agents" or inference engines. This multi-strategy reasoning ability of M allows the system to formulate different points of view while performing recognition and classification tasks.

In the applied domain of the VMR, it was useful and typically necessary that M simultaneously derive and manage several theories representing the actions of VMR participants and the state of all VMR objects (e.g., documents, files, pens, markers, erasers, etc.). This was due to the fact that certain classifications were not immediately obvious – either (1) they emerged over time or (2) given contextual situations enforced reformulation of existing classifications.

In my study, one of the key research issues concerned the management of the different reasoning processes as they collectively formulated multiple theories to recognize and classify a VMR world. This management function required a technique for the processes to "communicate" and leverage key information relative to distinct simultaneous classification theories of a given VMR situation.

In developing a design theory of M as an architecture of integrated reasoners, it was desirable to define a framework in which simultaneous theories of a world could be dynamically generated, ranked, and modified. For the applied problem of the VMR world, five different reasoning processes were required and implemented as distinct inference engines. The five types of reasoning supported in M are:

- structural
- functional
- spatial
- temporal
- causal

The integration and management of these inference engines was achieved via a traditional shared data structure and governing processes known as a blackboard system. In the M system, each reasoning process served as a knowledge source (KS) which inter-worked with other KSs via the blackboard system.

The design and implementation of M's blackboard system resulted in two unique features. First, M consisted of a dynamically ordered set of blackboards. Each blackboard hosted a distinct theory representing M's recognition and classification of a VMR situation. The set of blackboards were ranked based on the strength of each theory's probable correctness. Second, the structure for representing information posted by KSs to a given blackboard was based on Minsky's Society of Mind Transframe.

14 Blackboard systems

Blackboard systems are a means of implementing dynamic, opportunistic behavior among cooperating reasoning processes that share intermediate results of their efforts by means of a global data structure (the blackboard). Penny Nii (Nii 1989) describes the basic structure of a blackboard system in terms of three components:

- The knowledge sources (KSs). The knowledge needed to solve the problem is partitioned into knowledge sources, which are kept as independent processes.
- The blackboard data structure. The problem-solving state data (objects from the solution space) are kept in a global data store, the blackboard. KSs produce changes to the blackboard which lead incrementally to a solution to the problem. Communication and interaction among the KSs take place solely through the blackboard.
- Control. What KS(s) to apply when and to what part of the blackboard are problems addressed in control. Typically, a scheduling process performs the control function.

In addition to the organizational requirements, a particular reasoning (computational) behavior is associated with blackboard systems. The solution to a problem is built incrementally over time. At each control cycle, any reasoning assertion (e.g., data driven, goal driven, forward chaining, backward chaining, etc.) can be used. The part of the emerging solution to be attended to next can also be selected at each control cycle. As a result, the selection and the assertion by KSs are dynamic and opportunistic rather than fixed and preprogrammed.

15 Ranked blackboards

M's blackboard system consists of a dynamic set of ranked blackboards which are allocated and reallocated as needed. The maximum number of blackboards allocated at any given moment is seven. Each blackboard contains an emerging classification theory over some subset of actions and objects. Basically, an emerging theory can be thought of as a hypothesis to be proved by M's reasoners. M's reasoners attempt to develop a strong theory by individually applying axioms to a given theory's hypothesis on a blackboard.

As M observes actions being performed by VMR participants, M's semantic net, rule based system, and scripting system assert new facts, rules, and scripts respectively via the five KSs. The KSs collaborate by applying this information as axioms to the respective blackboard of a given classification theory. Further, as M computes the weighted scripts for each blackboard, the theories with the greatest weighted sum are ranked high to low, thus defining the dynamic ordering of blackboards.

16 Trans-frames

When a KS posts an axiom to a blackboard, this information can be viewed either as some type of modal information reflecting a modality of reasoning (e.g., spatial, temporal, structure, etc.) and/or some set of "conceptual dependency information" representing an action. The fundamental data structure of an individual blackboard is based on Minsky's Trans-frame. The Trans-frame provides a representation of an action, a trajectory between two situations; this information represents the pre and post states of a VMR situation.

The "conceptual dependency information" depicted in a Trans-frame structure includes:

- the actor performing the action
- instrument used by actor to perform action
- the action applied to some object(s)
- the object(s) with pre state properties
- the object(s) with post state properties
- the difference(s) between the pre and post properties
- list of plausible goals addressed by the action
- causal effect of the action

The Trans-frame structure provides a canonical form which enables M to effectively compare:

- different theories or sub-theories posted over the ranked blackboards,
- the various weighted scripts contained within the scripting system with a given theory posted on a blackboard, and
- the pre and post properties of the object(s).

Embedded within a Trans-frame structure are two object property graphs representing the object(s) pre and post state properties. This graph-based structure represents an object's properties based on the different modalities of reasoning. The application of this structure was reported by Winston et al. (Winston 1983) and Mitchell et al. (Mitchell 1986). The object property graph depicts properties based on their functional, structural, spatial, and temporal values and enables inference across different modal reasoning. Like the Trans-frame, the object property graph is a canonical form which enables effective evaluation and comparison of multiple objects.

References

M. Minsky. K-lines: A theory of memory. *Cognitive Science*, 4,:117–133, 1980.

M. Minsky. *The Society of Mind*. Simon and Schuster, New York, 1985.

T.M. Mitchell, R.M. Keller, and S.T. Kedar-Cabelli. Explanation-based generalization: A unifying view. In R. Michalski, editor, *Machine Learning, Volume 1, Number 1,*. Kluwer Academic Publishers, Netherlands., 1986.

A. Newell. Heuristic programming: ill-structured problems. In J. Aronofsky, editor, *Progress in Operations Research, Volume 3,*. Wiley, New York, 1969.

H.P. Nii. Introduction. In V. Jagannathan, R. Dodhiawala, and L. Baum, editors, *Blackboard Architectures and Applications,*. Academic Press, New York, 1989.

D. Riecken. Goal formulation with emotional constraints: Musical composition by emotional computation. In *AAAI Proceedings First Annual Conference on Innovative Applications of Artificial Intelligence, Stanford University,*, Cambridge, Massachusetts, 1989. AAAI/MIT Press.

D. Riecken. Adaptive direct manipulation. In *Proceedings of IEEE International Conference of Systems, Man and Cybernetics*, pages 1115–1120,, Charlottesville, VA., 1991a.

D. Riecken. Auditory adaptation. In *Proceedings of IEEE International Conference of Systems, Man and Cybernetics*, pages 1121–1126, Charlottesville, VA., 1991b.

D. Riecken. Human-machine interaction and perception. In M. Blattner and R. Dannenberg, editors, *Multimedia Interface Design.* ACM Press, New York, 1992a.

D. Riecken. WOLFGANG - A system using emoting potentials to manage musical design. In M. Balaban, K. Ebcioglu, and O. Laske, editors, *Understanding Music with AI: Perspectives on Music Cognition*. AAAI/MIT Press, Cambridge, MA, 1992b.

D. Riecken. M: An architecture of integrated agents. *Communications of the ACM*, 37(7), July 1994.

R.C. Schank and R.P. Abelson. *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

H. Simon. The structure of ill-structured problems. *Artificial Intelligence*, 9, 1973.

P.H. Winston, T.O. Ginford, B. Katz, , and M. Lowry. Learning physical descriptions from functional definitions, examples, and precedents. In *Proceedings National Conference on Artificial Intelligence*, pages 433–439, Cambridge, MA., 1983. AAAI/MIT Press.