

MOPs and Learning

Roger C. Schank
Yale University
Computer Science Department
PO Box 2158
New Haven, CT 06520

This paper is an attempt to sketch out some of what MOPs are about. It is taken from Schank (in press).

A MOP is an orderer of scenes.

A scene is a memory structure that groups together actions with a common goal, a common time, and some other common thread.

It provides a sequence of very general actions. Specific memories are stored in scenes, indexed with respect to how they differ from the general action in the scene.

Scenes actually point to specific memories. MOPs do not. MOPs merely point to scenes. Scripts are particularly common instantiations of scenes. Thus, a scene consists of a generally-defined sequence of actions, while a script groups together particular realizations of the generalizations in a scene. Scripts package together particular realizations of scenes that have been known to frequently recur in a given context. Specific memories can be organized in terms of scripts also. This follows from the above, since a script is no more than a scene that has been colored (particularly instantiated) in a given way.

MOP's Defined

Since memories are to be found in scenes, a very important part of memory organization is our ability to travel from scene to scene. A MOP is an organizer of scenes. Finding the appropriate MOP, in memory search, enables one to answer the question 'What would come next?', where the answer is another scene. That is, MOPs provide information about how various scenes are connected to one another.

A MOP consists of a set of scenes directed towards the achievement of a low level goal. A MOP always has one major scene that is the essence or purpose of the MOP.

There is a natural progression in terms of generality of structures that suggests itself:

- meta MOPs
- MOPs
- scenes
- scripts

Meta MOPs describe ordered progressions of scenes at their most abstract levels. As such they provide the stuff out of which MOPs are made. They do not actually contain memories. MOPs are less general descriptions of such progressions. The scenes they contain actually contain specific memories.

There are three kinds of scenes, physical, societal, and personal. Physical scenes represent a kind of "snapshot" of one's surroundings at a given time. Memories grouped in physical scenes provide information about what happened and how things looked.

Some MOPs refer to societal things rather than physical ones. M-CONTRACT is a MOP that organizes scenes that are not physically bounded. Thus, entities such as AGREE, or DELIVER, while behaving very much like scenes in a physical MOP, have no physical instantiation. They can happen anywhere and can take a great many different physical forms. Thus, a delivery of agreed upon services that fails to come about will be indexed under the DELIVER scene in M-CONTRACT. In this way, a failure of a department store to deliver a package that was paid for might remind one of a restaurant that required pre-payment and then failed to serve the desired food. Such reminding can only be accounted for by a memory organization that has scenes that are not exclusively physically bounded. DELIVER is an example of a societal scene, that is, one that may have many possible physical realizations.

Personal scenes are responsible for idiosyncratic behavior that is personally-defined. A personal scene is a scene whose common thread is a particular goal that belongs to the person whose scene it is. Any private plan to achieve one's own ends that is liable to repeat itself frequently is a possible personal scene.

This division in scenes is parroted by a similar division in MOPs. Physical MOPs can contain scenes that seem societal in nature, but what is actually happening is that one event is being governed by two scenes. Thus, for example, both M-CONTRACT which is a Societal MOP, and M-AIRPLANE, which is physical share a PAY scene. But each relates to different aspects of that event. In other words, "paying" can be seen as both a physical event and as a social event. Different MOPs provide expectations in each case. These expectations will coalesce to some degree, providing uniform expectations. Events confirming those expectations will be remembered in terms of both of the scenes that were active.

What is the difference between '\$AIRPLANE (that is, our prior view of a script) and M-AIRPLANE (that is, our current view of a MOP)?

The Difference Between MOPs and Scripts

A MOP is an ordered set of scenes.
A script (1977 version) is an ordered set of scenes

BUT---

The definition of scene is different in each case.

For a MOP, a scene is a structure that can be shared by a great many other MOPs.

For a script (1977 version) a scene was particular to a given script and was not accessible without using that script.

A script (new version) is scene-specific. No script transcends the boundaries of a scene.

Now, to make this specific, let us actually look at M-AIRPLANE and '\$AIRPLANE. Recall that '\$AIRPLANE was more or less a list of an entire airplane trip. It included making the reservation, getting to the airport, checking in, riding in the plane, eating the meal and so on. In SAM and FRUMP these things were all stored in a complex structure, complete with optional tracks, under the name '\$AIRPLANE.

But, now we wish to be able to make generalizations, get reminded across contexts and within contexts, and in general bring whatever relevant information from memory that we can find to help us in processing an input. To do this, we need structures that are far more general than a detailed complex list of events. For example, getting someplace by car, and making reservations by telephone are two scenes that were part of M-AIRPLANE that could not possibly be part of M-AIRPLANE. The reason for this is that one could easily confuse one trip in a car to visit a friend who lives near the airport, with a trip to the airport that was intended to enable one to fly someplace. Similarly, one could easily confuse a phone conversation making airline reservations with one making hotel reservations. In fact they might well be the same conversation.

The problem with our old conception of scripts was that much too much that could have been defined generally, and that is likely to be stored in a general fashion in memory, was defined specifically as a part of a particular script. When one takes away everything that could have been defined generally from M-AIRPLANE one is left with the things specific to M-AIRPLANE, namely getting on the plane, being seated, being served a meal and so on. The above entities are the scripts that we now believe in. That is, M-AIRPLANE is a structure that, like any MOP, organizes a set of scenes. One of these scenes is SITTING IN THE PLANE (SITP). This scene has in it a number of scripts specific to that scene. These include M-SITP-EATING, M-SITP-MOVIE, and so on. Experiences that occurred within them, that is while those scripts were directing processing, that did not coincide with the expectations generated by that script, would be encoded as failures and indexed within that script.

M-AIRPLANE fills one strand of the meta MOP-TRIP. It consists of the following scenes:

M-AIRPLANE's scenes

CHECK-IN + WAITING AREA + BOARDING +
SIT-IN-THE-PLANE + DEPLANE + COLLECT-BAGS

Each of the scenes used by M-AIRPLANE is constructed as generally as possible. We should point out that it is people who are doing the construction of these scenes. One of the scenes of M-AIRPLANE is something called WAITING AREA. Now it is reasonable to ask, is this the same as the scene as WAITING ROOM in M-PROF-OFFICE-VISIT? Clearly such answers depend upon the experiences a memory has had and the decisions about what is like what (its generalizations) that it has made. It is perfectly plausible that a memory that had been to a doctor's and a lawyer's office and had constructed a scene WAITING ROOM, might upon its first encounter with an airport, see the waiting area as a version of WAITING ROOM. And, of course it might not.

Our point is that the possibility for such generalizations, for interpreting a new experience in terms of what it believes to be its most relevant old one, must exist for a memory. In order to do this, scenes must be memory structures in their own right, disassociated from the structures they are used with in processing. Thus MOPs as we have outlined them must be the kinds of memory structures we need. Scripts, in the old version of them, were too restrictive in this regard. This does not mean that scripts do not exist of course. Some of the experimental work on scripts relates to MOPs as we have now defined them and some of it relates to our new, more restricted definition of them.

Higher level learning and generalization takes place by indexing a given expectation failure in terms of the MOPs and scenes that were active at each of the three levels of analysis whenever the expectation failure occurred.

One key problem that a theory of memory must explain is what to do when an expectation fails. Consider again the Legal Seafood case (first discussed in Schank, 1980). After processing an episode at Legal Seafood, we would want to have detected a MOP-based expectation failure and have so indexed it. Why is this a MOP-based failure and how does a system know what structure to alter? The MOP M-RESTAURANT indicates the order of occurrence of scenes in a sequence. One way that a MOP can fail is by having the ordering of scenes that it predicts turn out to be wrong. In Legal Seafood, the PAYING scene comes immediately after the ordering scene. Thus M-RESTAURANT would be marked, at least initially, with an index after ORDER that PAY came next in this particular instance. But, just simply marking M-RESTAURANT is not enough.

The main question that is generated by any expectation failure is: What alteration of the structure that generated that expectation must be made? There are three possibilities, alteration, reorganization, and the construction of a new structure.

Consider our visitor to Burger King and MacDonald's. A first encounter with Burger King, for a person whose knowledge structures contain only the standard M-RESTAURANT, would produce an expectation failure in the order of ORDER, SEATING, PAY. When multiple failures occur, it is a good bet that it is because the MOP being used was of little value. Thus, in a situation of multiple failure, a new MOP must be constructed. This construction is complex since it involves reworking the existing MOP to create the new one. This is done by altering the MOP first, and the scenes second, as follows.

As in the Legal Seafood example, in Burger King PAY goes right after ORDER. In fact, we might expect a reminding here if the Legal Seafood episode came first. We have an additional problem with respect to M-RESTAURANT in that the SEATING scene follows PAY and ORDER. Further there are some script expectation failures too. For example, M-RESTAURANT-ORDER is not usually done while standing.

The first thing that must be done then is to construct a new MOP. To construct a new MOP, we start with the scenes of the old MOP and reorder them according to the new episode. This is easy in the case of what we will temporarily call M-BURGER KING. The problem is that while the scenes may be the same, the scripts are different. A scene describes what takes place in general. And, in general, what takes place in a regular restaurant and a fast-food restaurant is the same. But the specifics are different. We do not want to use the scripts associated with M-RESTAURANT therefore. The problem then is to construct new scripts. Actually, this is hardly a problem at all. The new script is identical to the first Burger King episode. The real problem is to alter the scenes.

At this point we have a new MOP, M-BURGER KING, that contains the scenes ENTER + ORDER + PAY + SEATING with very specific scripts attached to each scene. Two problems remain. First we must encode the scripts correctly in the scene. Second we must generalize M-BURGER KING to the MOP that is more likely to be the one of greatest use, namely M-FAST FOOD. These two problems are related.

allows every MOP that uses ORDER to have that fix incorporated in it without doing a thing. The new, altered, ORDER is simply used by any MOP that previously used the old ORDER. In other words, this hypothetical person should now know to ask his wife to cook him things in a polite way and so on.

Now let's consider the Burger King example again. The problem in constructing M-BURGER KING is to take each scene that that MOP uses and treat each action that occurs within it in terms of its deviation from the baseline scene. Thus, M-BURGER-KING-ORDER is built by noting how the actions observed in the first experience with Burger King differ from the ORDER scene.

The problem is, of course, that we want this MOP to be M-FAST FOOD. To get this MOP to be built, it is necessary to index M-BURGER KING in terms of M-RESTAURANT. The reason for this is as follows: Consider a patron entering MacDonalds. We want this patron to get reminded of Burger King. To put this another way, we want the patron to know to use M-BURGER-KING and not M-RESTAURANT. How can this be accomplished? One way is to index M-RESTAURANT at the point of its failed expectation relevant to Burger King, in this case noting that the scene ordering was different in a particular way. Thus, M-RESTAURANT must now have in it a marker recalling the past expectation failure and directing the processor where to go for help in further processing.

After this rerouting of processing has occurred a few times in the same way, the reminding ceases to occur. At that point M-BURGER KING has been transformed into a MOP with entry conditions of its own, that is, one that can be called in for use without even seeing it as a type of restaurant. To put this more generally, a new MOP is grown at the point where its conditions for use have been detected so that it can be called up independently from the MOP in which it originated as an expectation failure. Thus, after a few trials, M-RESTAURANT and M-FAST FOOD are independent MOPs.

In general then, expectation failures that are MOP-based, will initially just produce markers valuable for reminding. However, if the failure is radical enough, a new MOP must be constructed immediately.

REFERENCES

- Schank, R. C. (1980) Language and Memory. COGNITIVE SCIENCE, Vol 4 no. 3, 243-284.
- Schank, R. C. (In Press) DYNAMIC MEMORY: A Theory of Learning in Computers and People.