The Mystery Machine: A Classroom Investigation of the Cognitive Processes Underlying Instructionless Learning

Jeff Shrager

"The Homework Machine, Oh, the Homework Machine, Most perfect contraption that's ever been seen..."
— Shel Silverstein

Abstract

Instructionless Learning is the process of figuring out an unfamiliar setting without being told. It is the more commonplace cousin of exploratory learning and scientific discovery, and is one of the most familiar and efficient ways people learn in everyday settings. I describe a classroom activity that enables students to observe and analyze instructionless learning at the cognitive level. Learners must figure out how to operate a slightly puzzling device, called the Mystery Machine, through observation, experiment, and reasoning. By observing a learner struggle to figure out the Mystery Machine, students learn about the cognitive operators that guide Instructionless Learning, and about the cognitive operators learners deploy in such settings, and about the gradual construction of "mental models" – the knowledge that enable us to reason effectively in complex settings.

1 Introduction

All of us are familiar with having had to figure out a new gadget or setting, for example, an new app, a rental car, a new checkout system at the store, how to order in a new restaurant, a new social setting, and so on. We are remarkably good at this kind of spontaneous "figuring out" – a cognitive skill that is more technically called "Instructionless Learning"Shrager and Klahr (1986).

Instructionless learning is neither mere guessing, nor is it as precise as scientific reasoning. Learners efficiently build and refine their knowledge of how the setting works and how to interact with it, through repeated cycles of observation, hypothesis formation, experimentation, explanation, and evolution of their knowledge about the setting.[1] This article describes an in-class experiment where learners are observed figuring out a puzzling device called the Mystery Machine. The activity was developed for Symbolic Systems 245, a senior applied cognition seminar at Stanford University. Over more than two decades hundreds of Stanford students have participated in it. The Mystery Machine is designed to be a little bit difficult and a little bit confusing, so that learners need to work a bit at figuring it out, but it is not so complex as to be frustrating. This balance enables students to analyze how the learners deploy Instructionless Learning.

2 The Mystery Machine Exercise

The present exercise adapts classic work by David Klahr and colleaguesKlahr (2000). The Mystery Machine itself (which we shall hereafter call just "MM", Figure 1, **?**) is browser-based. The exercise can be run in about an hour, including discussion. Students will directly observe the cognitive operators people use when figuring out novel systems, understand how mental models form and evolve during learning, develop skill in observing complex cognitive

[1] In this article, I will use the vague terms "model" (sometimes "mental model"), "understanding", and "knowledge" interchangeably.

activity in themselves and others, recognize the opportunistic, and learn about important cognitive processes including explanation and mental model evolution.

3 Procedure

3.1 Phase 1: Learning the MM Basics

Students form small groups. Each group opens one copy of the MM on one of their browsers. It comes up in "Phase 1: Learn the Mystery Machine." In the interest of space, and getting to the intresting second phase of the exercise I don't explain the MM here. I assume that whoever is guiding the exercise will have studied it before the session, and can explain it clearn to the students. I suggest guiding the class through this Phase so that within about ten minutes, they are proficient with the basic command entry, and what to expect. I also suggest using the displayed keypad explclusiovely, for most of it, but that at the end showing that they can do the same thing using their keypad. They should also experience pressing [Save History] at least once times so that they know how to save the history, and know where it is being saved.[2]

Note that the MM is *intentionally* designed to be somewhat confusing. The importance of this will become clearer when we get to the discussion. Here are some of the confusions that might arise: **Plus and minus are misleading:** Despite appearances, the MM is *NOT* a calculator. The "+" and "−" are just characters like any other. "5 + 1" produces "+++++" (five pluses), NOT "6" or "+ + + + ++" (six pluses). **Multiple pluses and minuses are hard to distinguish:** Here is the same program as just above, but using minuses instead of pluses: "5 − 1" or "− − − − −−". Could you tell that I put 6 minuses instead of 5? **Only the last digit matters:** The MM has a single-digit "register" that overwrites with each new digit. If you enter "12343247+" the output is "+ + + + + + + ++"

[2]Except for the very end of the exercise, this is mostly done automatically, but it's worth trying in this phase so that when it happens automatically the file-saving pop-up isn't a surprise, and because they will need to do it manually at the very end of the exercise.

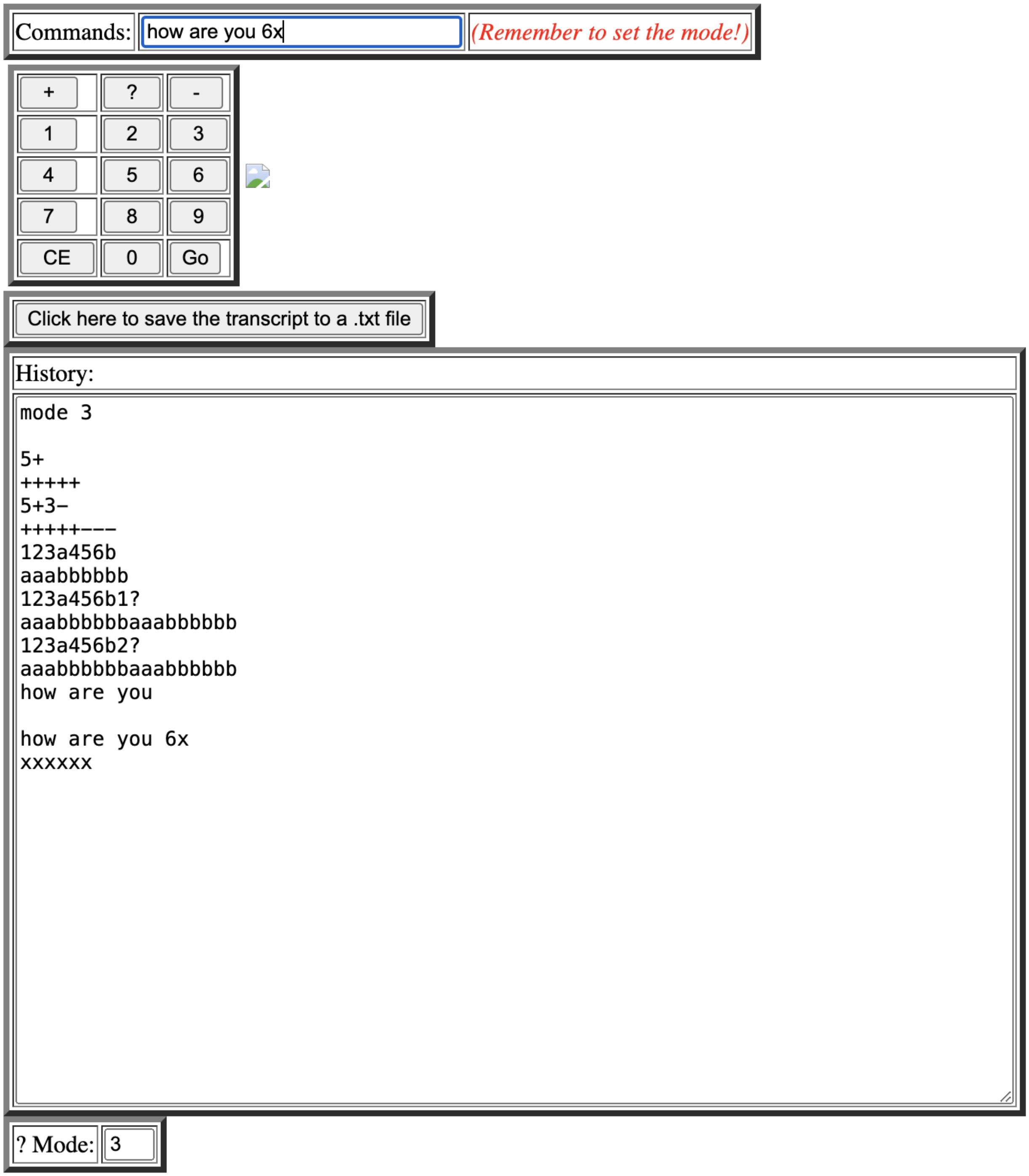

Figure 1: MM Interface Screenshot

(seven plus signs), because only the final "7" is retained. **No error checking:** Malformed programs still execute, often producing confusing results. For example, "$3 - +$" produces "$- - - + ++$" because the 3 applies to both the minus and the plus sign.

A path to less confusion: Alternative characters work equally well: "3x4y" produces "xxxyyyy" just as well as "3+4-". Once explorers discover this, they can avoid the confusing plus/minus characters, though many don't discover it on their own. I've found that it is useful to mention this in passing, but not emphasize it; explorers in phase 2 who are being confused by the minuses may recall that this is possible and use it spontaneously, but it's useful for them to experience some of the above-described difficulties before learning that they can avoid some of them by using x's and y's (or other characters of their choice).

Phase 2: Figuring Out the ? Key

Each team now chooses an initial "explorer". The task of the explorer is to try to figure out what the ? key does, as explained in more detail below. To begin, click the button called [Move to Phase 2]. This button will disappear and a mode will be displayed. The mode is initially set to 3.

The explorer can type anything they like using either the MM displayed keypad, or the computer keyboard (except, of course, to change the mode or to otherwise ruin the experiment). The task of the rest of the team in this phase is to keep the explorer talking about what they are doing – ask them what they are thinking. I find it useful to give each explorer only 5 minutes, and then, if the current explorer hasn't figured out the current mode, rotat the explorer role. Also rotate the explorer role once the team (that is, the current explorer) figures out the current mode (or the team gives up).

Once the explorer has either discovered the ? function, or given up so that the function is revealed by the facilitator, the group changes mode. I recommend this sequence: Mode 3 → Mode 4 → Mode 6 → Mode 2 → Mode 1 → Mode 5 (if you dare!). Note that the MM will try to save the input output trace to a local file; This will be useful in discussion, so agree in advance as to where these will be saved and named.

Generally I let this go on for about 20 minutes, leaving about 20 minutes in a one hour class session for discussion.

3.2 What the ? Key Actually Does

Here is what the ? key does in each mode. “#” refers to the digit before the ? (or anyway whatever the most recent digit was).

- **Mode 3** (simplest): Repeats the entire program once, regardless of the numerical prefix.
- **Mode 4**: Restarts at the #th *step* (digit-character pair).
- **Mode 6**: Restarts # *steps* back from the ?
- **Mode 2**: Restarts at the #th *character* position (where the first character is #1).
- **Mode 1**: Restarts # *characters* back from the ?
- **Mode 5**: Restarts at a *random character position* (non-deterministic!)

Aside from Mode 3, which is fairly simple, these can be extremely confusing. Modes 4 and 6, for example, refer to ”steps”, meaning digit-character pairs, where the first pair is step #1, and so on. To do this, the MM skips two characters at a time through the program to the desired point. For example, if the program is ”3x2y1z”, and you add 3? in mode 4, that is: ”3x2y1z3?” the result will be ”xxxyyzz” because the program skipped to location 5 – the character at which the 3rd step begins. This is find, if the program is in ”standard form” – that is, a series of digit-character pairs, as above. But recall that the MM does not check that the entry is in standard syntax, and it is quite common for explorers to be careless about this, making these modes *very* confusing.[3] So, for exmaple, if one were to put space between one’s steps, as: ”3x 2y 1z 3?”, the result would be: ”xxx yy z yyy z”. I leave it as an exercise for the reader to figure out why this output is correct for mode 4 (hint, it starts at the same *character position* as in the correct example, above).

[3] It may be useful for the facilitator to remind the explorer to use standard syntax, if they are not already, and are becoming confused as a result.

Modes 2 and 1 are even more confusing because they interpret the argument as an instruction start at a particular character position in the previous body of the program, not on a particular step. Therefore, the results of the previous examples in mode 2 are: "3x2y1z?3" -¿ "xxxyyzxyyz", and "3x 2y 1z 3?" -¿ "xxx yy z yy z". Again, we leave it to the reader to figure out why.

Another thing that the facilitator might mention if the explorer is becoming confused is to encourage them to think about what the number before the ? might be referring to.

This set of examples provides an opportunity to observe another interesting, and somewhat annoying, "feature" of the MM. Note the subtle difference between the two second sets of outputs: "xxx yy z yyy z" v. "xxx yy z yy z" Even though you may have noticed that there is an extra 'y' in the first output, you probably didn't notice that there is also an extra space before the final 'z'. The same goes for the plus (+), and especially for the minus (-) characters. This seemingly trivial misfeature can lead to interesting learning phenomena, as we'll see below.

4 Discussion Part 1: The Shape of Instructionless Learning

4.1 The 8E's: Core Cognitive Operators in Instructionless Learning

Instructionless Learning is a problem solving activity. Human problem solving is defined by a goal and operators. Here the goal is a correct understanding of the ? key for a particular mode. The operators are cognitive or physical actions that the person uses to change the problem state, eventually (hopefully) reaching their goalNewell and Simon (1972). I describe eight operators – what I call "The 8 E's" – that explorers use to reach the goal in the MM exercise, and, indeed, in any sort of instructionless learning setting.[4]

Before studying these operators, it is important to understand that human problem solving is fundamentally *opportunistic*. It doesn't follow a rigid sequence. No two explorers will reach the goal in the same way, and the specific order in which operators are used by a given explorer in a given setting depends on countless factors, such as what the explorer notices, what they happen to try, what confusions they encounter, and what knowledge they bring to the setting. Although we will present these 8 operators in a particular order, as there seems to be a natural order to some of them, there is no hard-and-fast rule that says what order these must appear in during the course of figuring out what the MM's ? key does. So the teams will almost certainly observe that their reaching (or anyway reaching for) the MM goal will take a path that differs from the one described herebelow. Eliciting stories if ways in which various groups differed in the shape of their explorers' problem solving can be a good kicking off point for discussion.

The 8 E's are:

- **Exploration** - trying various inputs without specific expectations about what the MM will do in response,
- **Evidence** - observing what the MM does in response to a particular input (either for each input, or from the History, or from one's memory),
- **Explanation** - making sense of the evidence based on one's current model,
- **Experimentation** - testing a model (or partial model) by trying an input with an expectation (or partial expectation) of how the MM will respond,
- **Expectation** - predicting what should happen in response to particular inputs,
- **Evaluation** - assessing whether the evidence matches one's expectations,
- **Evolution** - updating one's mental model based on explanation,

[4] The silly name helps me remember whether I've got them all when teaching this!

- **Exercise** - playfully practicing once a correct (or presumed correct) model is achieved by inputting commands with a ”sure” expectation, that is where the learner assumes that these will work (v. an experiment where they have an ”uncertain” expectation – they are *not* sure whether it will work or not).

4.2 The Typical Arc of Instructionless Learning

When first confronted with the ? key, some explorers may simply guess at its function. They may even guess correctly, although they still need to run **experiments** to confirm that they are correct. More commonly, explorers begin with **explorations**—trying several programs containing ? with only vague **expectations** of what might happen. They gather **evidence** (the outputs), then engage in **explanation**, usually forming (**evolving**) a preliminary mental model.

To validate this model, they run **experiments** with more specific **Expectations**. They again gather **evidence**, but now they **evaluate** it against their predictions. If the **evaluation** succeeds – the **evidence** matches (or apparently matches) their **expectation** – they may declare victory and move to **exercise**—using their model playfully.

If **Evaluation** fails, explorers face a choice. They might try to **Explain** the mismatch between their **expectation** and the **evidence**; they might discard their **expectation** and return to try to **explain** the **evidence**d behavior independently of the failed **expectations**; They might return to **exploration**, seeking new **evidence**; or they might reject the **experiment** as poorly designed. Of course, at any point the explorer might simply give up! (Perhaps this should be the 9th ”E”: Exit! :-)

When **explanation** leads to mental model **evolution**, the cycle begins again with new **experiments** based on the revised model, until the leaner decides that they are satisfied that they have a correct model, at which point (as above) they fall into **exercises**.

4.3 The 8 E's in More Detail

4.3.1 Exploration: Poking Around Without a Plan

Students will observe that subjects often begin by just trying things. An Exploration differs from an Experiment in stance and expectation. In Exploration, the goal is simply to gather **Evidence** without a specific prediction of what will happen.

Example from a protocol:

> "I'm just going to try something... okay, let's see... 3x2?" [Types 3x2?, clicks GO] Okay, that gave me... xxxxxxxx. Huh. So it did something."

Notice the explorer isn't testing a specific hypothesis; they're fishing for data (evidence) that might suggest a beginning model.

4.3.2 Evidence: Gathering Information

Every interaction with the MM produces **evidence**—the output displayed on screen – even no output is potentially useful evidence. However, students will observe that evidence gathering isn't as simple as just reading what appears. Subjects often miscount (as described above) the output (especially with + and -); misremember what they typed; fail to notice important patterns; and see what they expect to see, rather than what's actually there (called "expectation bias"). Note that the problem of expectation bias is exacerbated by the difficulty of correctly counting the characters (especially the minuses).

4.3.3 Expectation: What Should Happen?

Expectations come in two varieties: **Vague expectations** (typical of Exploration): "Something will happen and I'll learn from it." **Specific expectations** (typical of Experiments): "This program should produce exactly this output." Students may observe that the specificity of expectations dramatically affects how explorers respond to results. A vague expectation means almost any output is "interesting," while a specific expectation creates a clear success/failure criterion.

4.3.4 Evaluation: Did It Match?

After gathering **evidence**, explorers **evaluate** whether it matched their **expectation**. Evaluation seems trivial but during **exploration**, evaluation is loose – subjects are satisfied if they learn anything at all. By contrast, during **experiments**, evaluation should be rigorous—but often isn't as a result of expectation bias, as desccribed above, which can cause subjects to "see" what they expected even when it's not there. As a result, near misses are often accepted as successes.

An Example of failed evaluation: "So if I do 4-2? [enters 4-2?, resulting in: "——"] that should give me... wait, how many dashes is that? Let me count... okay, that's eight. So it doubled it! I think it repeats the command." [This subject miscounted—it was actually six dashes, not eight – they were in mode 1. They've formed an incorrect theory based on miscounting.]

4.3.5 Explanation: Making Sense of It All

Explanation is where explorers try to explain what the ? key does based on their observations. In doing this they draw onthe **Evidence** they've gathered, their current mental model of the MM, and their background knowledge (of calculators, programming, math, etc.)

Explanations come in many forms. Here is an example of an early explanation, immediately after the explorer's intitial exploration: "I think it does something with the number before it... maybe it repeats?" This is an example of a mid-stream explanation, after a failed experiment: "Okay, so it's not just doubling... maybe it's adding the number to the command? No, that doesn't make sense either..." Note that the explorer tests this explanation internally (calling upon evaluation, probably based on evidence in the history).

Explanation is deeply tied to a process I'll call "interpretation" or "view application," which we'll explore in some depth below, in part 2 of the discussion.

4.3.6 Experimentation: Testing Specific Ideas

Once explorers have a model (from **explanation**), they often design **experiments** to test it. An **experiment** differs from **exploration** in having a specific hypothesis, that is, a specific **expectation** of the outcome, and a clear criterion for **evaluation**.

However, explorers are often quite poor at designing good experiments. Common problems include creating non-discriminating experiments, that is experiments that don't distinguish between alternative models. For example, if the explorer thinks "?" repeats the whole program, they try "2+1?" expecting "++++". But this result is also consistent with "? doubles the last command," "? adds 2 to everything," and many other theories.

Another common problem is creating overly complex experiments, usually resulting evidence that is too complicated to explain. For example, "Let me try 12x34y56z78?" [Enters that. In mode 3 this results in "xxyyyyzzzzzzxxyyyyzzzzzz"] "Okay, that gave me... wait, what did I even expect this to do?"

Explorers will also commonly create experiments where multiple things change at the same time, leading to confusion.

Despite these limitations, subjects usually succeed eventually, at figuring out the correct model for the ? key in most modes. We'll return to why this is the case in the discussion.

4.3.7 Exercises: Playing With the Model

Once subjects believe they understand the ? key, their stance shifts from testing to using (or playing). **Exercises** often have **expectations**, but subjects are less rigorous about **evaluation**. They're confident enough that small discrepancies get ignored. This is why subjects often declare success even with partially incorrect models. ==============================

4.3.8 Evolution: Changing the Mental Model

When **evaluation** reveals that the gathered **evidence** doesn't match **expectation**, explorers must decide what to do. The most productive response is **evolution**—changing their mental model to account for the new data. But first, the explorer will usually undertake **explanation**-an attempt to find a view that can account for the **evidence**. Once a view has been discovered by the explanation process, that view is merged into the mental model, resulting in a new model. And then the explorer will return to experimentation, testing this new model.[5]

This process of explanation finding a view, and then the view being merged into the existing mental model (possibly replacing it entirely), is the among the most important cognitive processes of all. Unforunately, it is also among the least well understood cognitive processes because it is "cognitively impenetrable"-it can't be easily closed down and broken apart for careful study by psychologists. In this way, model evolution is like a perctual process, which is why the terms "view application", or "commonsense perception" are appropriate. Indeed, view discovery (the result of an explanation) is often accompanied by exclamations that imply a perctual prcoess, like "Oh! *I see*! It's not counting characters, it's counting the PAIRS!" ================

5 Satisficing: Why Imperfect Problem-Solving Succeeds

At this point, students might notice a paradox: **Subjects are simultaneously very good and very bad at figuring things out.** They're **good** at eventually reaching correct (or mostly correct) understanding. They're **bad** at designing optimal experiments, carefully evaluating evidence, avoiding expectation bias, and systematically forming and testing

[5] Klahr and Dunbar (1988) described instructionless learning as "dual-space search" where explorers simultaneously search a **space of experiments** and a **space of theories**.

hypothesis.

So why does instructionless learning work at all?

5.1 Explorers Satisfice Efficiently

Herbert Simon, the Nobel laureate who is usually credited with having founded cognitive science, introduced the concept of **satisficing**—accepting a solution "that will permit satisfaction at some specified level of all of [the explorer's] needs." rather than searching for the optimal one. This is a consequence of **bounded rationality**: humans have limited time, knowledge, and cognitive capabilities (Simon (1956), p.136).

Students will observe that subjects don't design perfect experiments—they try something reasonable; don't exhaustively test hypotheses—they run a few trials and move on; don't carefully analyze all past evidence—they focus on recent observations; and don't eliminate all alternative theories—they accept "good enough" *This is actually smart*, not stupid! The stakes are low, data is cheap and fast to collect, and subjects aren't professional scientists. The MM environment rewards trying a lot of simple things rapidly, rather than than careful systematic investigation.

There are many settings where satisficing results in much more thoughtful and careful behavior. Consider scientists trying to create cancer drugs. Medical xperiments cost millions of dollars, results take years to obtain, effect sizes are tiny, and human lives depend on getting it right. Drug scientists spend years on preliminary work before undertaking human trials, and they must design optimal experiments to the best of their ability because the cost of poor experiments is measured not just in dollars, but also in human suffering. The MM is at the opposite extreme: experiments are free, instantaneous, and low-stakes; trying lots of simple experiments and not worrying about designing them carefully, nor about interpreting the results perfectly, is a good strategy here.

6 Discussion Part 2: How Mental Models Form and Evolve

Having explored **what** explorers do (the 8E's), we now examine **how** understanding emerges. This requires delving into mental models, interpretation, and a ubiquitous cognitive process called "view application."

6.1 Understanding and Mental Models

Throughout this discussion, we've said explorers "understand" the MM or "figure out" what the ? key does. But what does "understand" actually mean? In each case, "understanding" means roughly: *having whatever knowledge, skills, and problem-solving ability you need to achieve your goals in this domain.*[6]

Cognitive scientist often use the term "mental model" to describe a person's understanding of a particular setting. A mental model is whatever internal representation enables you to reach your goals within the setting, including operating within it, reasoning about it, explaining it to yourself or others, predicting its behavior.

6.2 The Three E's of Understanding: Evaluation, Explanation, Evolution

Five of the 8E's (Exploration, Experimentation, Expectation, Evidence, Exercise) describe the **shape** of instructionless learning—what explorers do and when. But three E's—**Evaluation, Explanation, and Evolution**—describe the **content** of learning: how mental models actually form and change. These three are deeply intertwined with a fundamental cognitive process called **interpretation**.

[6] Being correct isn't required for understanding; An incorrect understanding is still an understanding.

6.3 Interpretation: The Foundation of Understanding

Students will observe that subjects often have moments of sudden insight:

These moments signal mental model evolution. But what actually happens in these moments?

The answer involves a cognitive process that operates largely below conscious awareness. I call this process **"view application"**. Understanding this process is crucial because it doesn't just operate in the MM—it operates constantly, in every domain of human activity.

6.4 View Application: A Ubiquitous but Mysterious Process

View application (VA) is so ubiquitous that it's been discovered and redescribed many times under different names "framing"**?**, "script matching"**?**, "conceptual blending"**?**, "conceptual combination"**?**, and VA is relate to analogyGentner (2002). All these researchers discovered approximately the same cognitive process operating in different contexts. VA updates your overall cognitive state—including your mental models—in light of new information. The phrase "I see what you mean" captures this perfectly—understanding feels like seeing, even though nothing visual is happening.

View Application is ubiquitous. MORE HERE?

6.5 VA in the Mystery Machine

Students will likely have observed VA operating throughout their protocols. When the explorer says: "Oh, it's like a calculator!", they are applying a "calculator" view. Commonly they will notice in mode 3 that "Oh! It's repeating the whole program!". At this point a "loop" view may be applied. And so on throughout the exploration. Each of these moments involves VA—the subject taking an abstract concept (calculator, loop, ...) and applying it to reformulate their understanding of the MM.

6.6 Evolution Through Interpretation

So how do mental models actually **evolve**? Through repeated cycles of view application. They may recognize an explorer saying something like: "I thought it was doubling ... but this result doesn't fit ... oh! Maybe it's counting back from the question mark ... let me try... yes!"

6.7 Practical Implications for Students

Despite not fully understanding VA, students should appreciate that **Understanding emerges through interpretation**, not through accumulation of facts. **Different people may apply different views**, leading to different (but functional) understandings. **The views available depend on culture and experience**. **Teaching is partly about providing useful views** to apply, not just facts to memorize. **Design is about enabling correct interpretations** from the start.

7 View Application v. Analogy

The question often arises as to the difference between analogy and view application. Analogy is a process that transfers specific content from one domain to another (the MM is like a calculator, thus transfering, say, the addition and subtraction knowledge). Analogy looks very similar in some cases to View application, but whereas analogy adds knowldge, view application reformulates it using abstract perspectives. It is often hard to tell the difference between these because in order to make analogical carry-over work, the learner often has to undertake a view application-based reformulation of their mental mode, so analogy often – indeed almost always – becomes view application in practice.

8 Topics for Advanced Discussion

The "child as scientist" and "lay" science: Children naturally engage in hypothesis testing and exploration. How does their instructionless learning compare to adults'? Is science just formalized instructionless learning?

Cross-cultural differences in mental models: Different cultures provide different "views" to apply. In what sorts of setting might an analytical mindset vs. a holistic mindset lead to different models?

Interpretive drift and cultural inculcation: Tanya Luhrmann's concept of "interpretive drift" describes how beliefs gradually shift through small reinterpretations.Luhrmann (1989) How does this relate to view application? Could it explain how children acquire culture?

Individual differences: What factors beyond culture affect instructionless learning? Prior knowledge, cognitive abilities, personality, motivation? What role does randomness play—just happening to notice a crucial cue?

The limits of satisficing: In what domains is satisficing inappropriate? How do professional scientists, engineers, and doctors overcome satisficing tendencies? What training or tools support more systematic investigation?

9 The Ubiquity of Instructionless Learning

Instructionless Learning—operates constantly throughout our lives, from infancy through adulthood, as we navigate unfamiliar buildings, learn new technologies, understand other people, and make sense of novel situations.

The Mystery Machine experiment reveals that "figuring things out" is not a simple process. It's not just "practice makes perfect" or a straightforward march up a learning curve. **Microgenetic analysis** shows that learning involves a complex orchestration of cognitive operators—the 8E's—deployed opportunistically in response to the specific situation and the learner's specific experiences.

Moreover, understanding—the goal of instructionless learning—doesn't emerge simply from accumulating facts or observations. It emerges through **interpretation**, a fundamental cognitive process that operates largely below conscious awareness. When subjects suddenly say "Oh, I get it!", they're experiencing view application—the reformulation of their mental model in light of new conceptual perspectives.

The Ubiquity of View Application: Students should leave this exercise recognizing that interpretation, by whatever name, operates constantly. Every sentence you read requires interpreting words in context. Every conversation requires interpreting meaning and intent. Every new situation requires applying familiar frameworks to novel circumstances. Every "aha!" moment involves restructuring your understanding. The MM experiment creates a context where this usually-invisible process becomes observable. The moments when subjects say "Oh!" or "Wait..." or "I get it!" are windows into interpretation in action.

Broader Implications: The Mystery Machine experiment, while simple, exposes some of the most important and powerful aspects of human cognition. Educators might recognize that students are constantly trying to figure things out. Educators can provide useful views to apply, structure learning paths to make productive discoveries likely, and encourage thoughtful exploration. Good teaching is largely about offering useful views at the right moments. Coaches don't just provide views; they **structure the learning path** to make useful discoveries more likely by sequencing practice to build capabilities incrementally. Designers might recognize that easy-to-learn interfaces facilitate efficient and correct interpretation, and, if necessary, support safe exploration. Understanding instructionless learning is essential for anyone designing systems that people must figure out.

Humans are remarkable. Even thought we have poor memories, design poor experiments, and make many mistakes, we are excellent at figuring things out.