The Artificial Intelligence Laboratory
Massachusetts Institute of Technology

Working Paper 263 December 1984

The Role of Intensional and Extensional Representations in Simulation

Daniel Brotsky

Abstract

I review three systems which do simulation in different domains. I observe the following commonality in the representations underlying the simulations:

- The representations used for individuals tend to be domain-dependent. These representations are highly structured, concentrating in one place all the information concerning any particular individual. I call these representations *intensional* because two such representations are considered equal if their forms are identical.
- With important exceptions, the representations used for classes of individuals tend to be domain-independent. These representations are unstructured sets of predications involving the characteristics of class members. I call these representations *extensional* because two such representations are considered equal if the classes they specify are identical.

I draw out various ramifications of this dichotomy, and speculate as to its cause. In conclusion, I suggest research into the process of debugging extensional class representations and the development of intensional ones.

This paper was prepared as the author's area examination.

The Role of Intensional and Extensional Representations in Simulation

A task central to much of intelligent activity is simulation of real-world processes. We are often forced to internally simulate external processes in order to plan a course of action, predict the ramifications of events, or analyze what went wrong. If we wish computers to perform planning, prediction, or debugging tasks, we must develop algorithms and representations that enable them to do simulation. I consider here some of the representational questions involved.

My comments in this paper arise from the study of three research efforts involving simulation. Davis (1984) simulates digital circuit operation, Simmons (1983) simulates geologic processes, and Weld (1984) simulates enzymatic interactions. All three authors choose their representations for objects based on domain considerations, but the results seem to me to display an interesting commonality. In particular:

- The representations used for individuals tend to be domain-dependent. These representations are highly structured, concentrating in one place all the information concerning any particular individual. I call these representations *intensional* because two such representations are considered equal if their forms are identical.
- With important exceptions, the representations used for classes of individuals tend to be domain-independent. These representations are unstructured sets of predications involving the characteristics of class members. I call these representations *extensional* because two such representations are considered equal if the classes they specify are identical.

This dichotomy in object representation leads to a similar, predictable dichotomy in process representation:

- If a process operates on individuals, its representation tends to be operational: it is specified as a series of additions to, deletions from, and alterations of the representations of its targets.
- If a process operates on classes, its representation tends to be declarative: it is specified as a set of predications about members of the class which are asserted to be true when the operation completes.

I will comment at length a bit later on the probable source and ramifications of this dichotomy in representation. But first I will spend some time making the dichotomy concrete by examining two of the simulations.

Simulation of Geologic Processes

Simmons (1983) addresses the task of recovering the geologic histories of rock formations. He takes a diagram showing a rock formation, such as that in figure 1, and hypothesizes a sequence of parameterized geologic processes that would produce the pictured formation. For example, figure 2 shows a sequence of processes that produce the formation of figure 1.

Simmons's system, only portions of which are implemented, approaches this problem much as a geologist might. The system starts by noticing features of the formation that suggest the occurence of particular processes, such as the igneous "dike" in figure 1. A temporal partial order is imposed on the suggested processes by other features, such as the relative heights of deposited layers. The end result of all the pattern matching is a partially-ordered sequence of geologic processes that together should account for all the formation's features. The system then checks this sequence with a simulation.

The verification process, termed *imagining* by Simmons, has two goals:

- Determine the physical feasibility of the proposed sequence.
- Determine the quantitative parameters of each process.

Simmons achieves these goals by performing two simulations: one qualitative, one quantitative. The function of the qualitative simulation is to determine process feasiblity and the interaction of quantitative parameters among the various processes. This then allows the system to deduce exact quantitative process parameters from measurements in the goal diagram. The function of the quantitative simulation is to determine process feasibility and to verify that the hypothesized processes, when acting with the hypothesized parameters, does indeed produce the goal formation.

Multiple Representations Support Multiple Reasoning Techniques

The representations underlying the two simulations are a major focus of Simmons's work. He uses a variety of distinct representation schemes for different objects, justifying each in terms of the operations it must support. For example:

- Rock formations in the qualitative simulation are represented by frame-like objects whose slots contain time lines of values. The time lines facilitate reasoning about change over time.
- Rock formations in the quantitative simulation are represented by diagram objects which are minimally-connected edge graphs. Each edge

is annotated with the faces it separates. This structure allows easy measurement of diagram parameters and the efficient manipulation of edges representing rock boundaries.

- Qualitative and quantitative values used in the two simulations are compared with each other using a partial-order lattice maintained by specialized interval-arithmetic and truth-maintenance procedures.

Each of these representations is intensional. Simmons makes a careful analysis of the types of reasoning required by the domain, and then constructs representations whose structure facilitates this reasoning. Questions about instances are answered by checking local structure; changes in instance properties are effected with local structural changes.

Unfortunately, Simmons does not explain how he keeps the various representations consistent with each other—a task that could be quite difficult in general. As an aside, I conjecture that the issues involved in this task are quite similar to those involved in modelling classes: one must understand all the paths of interaction between the various facets of instances being represented separately. This question deserves further research.*

Different Representations Underly the Different Simulations

Simmons's quantitative simulation is completely supported by his intensional instance representations. Quantitative processes are represented as sequences of diagram manipulations. For example, figure 3 shows Simmons's quantitative process model for erosion.

Simmons's qualitative simulation is not completely supported by intensional representations. Consider the qualitative process model of erosion shown in figure 4. Portions of the model are specifications of intensional operations; for example, the statement

```
(change = BA.orientation 0.0 I EROSION)
```

sets the orientation of the eroded surface to horizontal and marks EROSION as the resposible process. Much of the model, however, is extensional: statements such as

$$\texttt{(> SURFACE.top.height@I}_{\texttt{start}}\texttt{ SEA-LEVEL)}$$
$$\texttt{(> ELEVEL SEA-LEVEL)}$$
$$\texttt{(}\leq\texttt{ SURFACE.bottom.height@I}_{\texttt{end}}\texttt{ SURFACE.bottom.height@I}_{\texttt{start}}\texttt{)}$$

make no alterations to the intensional representations. They serve instead as predications on the possible rock formations that could be modelled by

*Perhaps some poor soul could do an area exam about it?

the intensional representations; that is, they limit the extensions of those representations.*

Qualitative Rock Formations are Classes of Quantitative Rock Formations

I claim that the distinction between the quantitative and qualitative process models arise because of a class/instance distinction in their target representations. Simmons has perspicuous, intensional representations for each individual rock formation he deals with, but he has no intensional representation for classes of formations. He must specify classes extensionally by predicating appropriate characteristics for its members. This forces his qualitative process representation into a declarative form:

- Each process uses a set of (primarily geometric) preconditions to define a class of instances to which it is applicable.
- Each process uses a set of (primarily geometric) effects to define a class of instances that contains the rock formation resulting from the process.

Much of the geometric information contained in Simmons's diagrammatic instance representation is duplicated in the intensional representations used by the qualitative simulator. This is partially motivated by the qualitative simulator's need to model the temporal changes in various quantities, a need the quantitative simulator does not share. But another motivation comes from his modelling of classes extensionally:

- Many of the predicates that limit the extensions of the qualitative representations refer to geometric information that is in the diagram representations. Thus, this information must be replicated in data structures accessible to the qualitative simulator.

A natural way to avoid this duplication of information would be to use some form of "blurred" diagrammatic representation to intensionally represent classes of instances. The qualitative process models could then be operational. Simmons addresses this issue from two perspectives, albeit indirectly. First, he says "...how a formation will split due to faulting, or what the shape of the formation will be after erosion are both difficult to express qualitatively."† This suggests that a good intensional representation would be hard to find. Second, Simmons says "...qualitative models

*To bring home the unimportance of the form of these assertions, consider that the order of their arguments has no effect on their import; they could all have been written using the complementary predicate.

†Simmons (1983), p. 25

are inherently ambiguous, in that a single qualitative representation maps to many real-world situations."* Since Simmons intends his diagrammatic representation to be geometrically unambiguous, this statement suggests that it might be methodologically incorrect to combine the class and instance representations.

I believe that Simmons is both right and wrong. In particular:

- The development of good intensional representations for classes always follows their development for individuals. Class representation development is guided by observation of instance behavior under controlled circumstances—*i.e.*, simulation—and such simulation can not proceed without good representations for the individuals.
- Whether intensional representations of classes and individuals should be structurally related depends on the operations these representations must support.† In Simmons's case, for example, the fact that both the qualitative and quantitative processes manipulate geometic characteristics of individuals suggests that structurally similar representations might be indicated.

I will return to these issues after an examination of Weld's simulation of enzymatic interaction.

Simulation of Genetic Activity

Weld (1984) simulates the interactions of enzymes and proteins in genetic activity. His system is given a description of the various proteins present in a situation, together with process descriptions that govern their possible interactions, and predicts the stable state—if any—that the protein interactions will lead to.

Weld's system, like Simmons's, performs two types of simulation in order to make its predictions. First, it simulates protein interactions as discrete operations involving a small number of proteins. As it performs this simulation, it monitors the particular protein molecules involved in each reaction. Then, when a cycle of reactions involving the same types of proteins is noticed, the system suspends the discrete simulation. The system transforms the representation of the discovered cycle from a discrete model into a so-called *continuous* model. In this view, the cycle is considered an individual reaction which is continuously active, and Weld's system simulates its continuous behavior in order to determine the limit towards which it tends. This limiting state is then translated back into the

*Ibid, p. 25

†This is a truism, of course, but worth repeating in this context.

discrete view and the first simulation starts again. A succinct example of system behavior is given in Weld (1984), §1.1, page 2.

Discrete vs. Continuous is like Qualitative vs. Quantitative

I focus here on the representations that underly the two simulations.* As with Simmons's work, we find that one of the simulations—the continuous one—rests on a base of entirely intensional representations, while the other—the discrete simulation—manipulates intensionally represented individuals whose possible extensions are restricted by predications:

- Each state of the world in the continuous model consists of a single reaction represented by a list of the amounts of surrounding proteins and the influence (positive or negative) the reaction has on those proteins.
- Each state of the world in the discrete model consists of molecular clusters of various types (represented intensionally) which have history (represented intensionally) and which are constrained extensionally to be in certain relations to each other.

The discrete and continuous process models show the effects of the underlying dichotomy in representation, just as they did in Simmons's work:

- A continuous process† is represented as an algorithm that alters the representation of a continuous reaction according to the contents of that representation and pre-computed limit points.
- A discrete process is represented as a set of extensional preconditions and a set of extensional effects, together with some intensional actions on the underlying molecular representations. (The intensional actions are encoded as demons which are attached to predicate names.) A discrete process representation is shown in figure 5.

---

*Note that Weld's focus is neither the discrete nor the continuous simulations in and of themselves, but the aggregation process that recognizes discrete-process cycles and transforms them into continuous-process individuals. Thus, he does not carefully justify his simulation representations to the extent that Simmons does. In his literature review, he attributes much of his discrete-simulation representation to work on situation-action rules (Fikes 71), and his continuous-process simulation to work on Qualitative Process theory (Forbus 1983).

†There is only one continuous process—the one that takes a reaction and computes its limit. It appears from Weld (1984) that this one is hard coded in LISP.

Continuous Individuals Represent Classes of Discrete Individuals

I return now to the issue of using intentional class representations that I raised at the end of my review of Simmons's work. Weld's work provides an interesting perspective once we notice that his continuous-process individuals are in fact representations of classes of discrete-process cycles. For example, consider the two sequences of reactions outlined in figure 6. In the left-hand reaction, A causes E to fold so that it can catalyze the reaction $C \mapsto D$; In the right-hand column it is B that does so. Since the cycles have no net influence on either A or B, their continuous images are identical.

- The continuous process individuals function as *class representatives* in that they encapsulate in an intensional representation all the interesting characteristics of a class of discrete process cycles. The perspicuous nature of this representation allows the efficient computation of relevant characteristics of the class, in Weld's case its limit point.
- The function of Weld's aggregator is to identify process cycles and, having done so, to construct the class representative for that cycle's class.

This perspective suggests a research direction that Weld does not address, but which speaks directly to the extensional/intensional dichotomy:

- Are there classes of non-cyclic discrete process chains that are of interest? Is there a perspicuous intensional representation for the class representatives of such classes? If the discrete process simulator were built on such a representation, would the aggregation task become easier?

Causes of the Intensional/Extensional Dichotomy

I have argued that both Simmons and Weld use intensional representations for individuals and extensional ones for classes. As one might guess from the questions I have raised, I would prefer to see only intensional representations used. My view is that:

- We use extensional representations for classes because we do not know how to characterize them completely. The extensional representations allow us to assert what we know about the class without specifying it completely.
- Once we know of a complete characterization for classes, we use it to construct intensional representations. Even when we only have characterizations for some classes of interest, we construct intensional representations for those classes.

A good example of the use of intensional representation for distinguished classes and extensional representation for others is provided by Davis (1984), which I now examine.

Class Representatives in Circuit Simulation

Davis (1984) considers a variety of questions pertaining to the debugging of digital circuits.* He presents a simulation technique that predicts the behavior of an individual circuit using a highly intentional representation built from the circuit's schematic. The circuit representation models each component of the circuit as a small constraint network and then connects these networks according to the schematic to produce a large constraint network representing the circuit as a whole.†

Note that this constraint-network interconnection technique produces an intensional representation for a single circuit, that is, an individual. Thus, its utility is limited. For example, consider the circuit shown in figure 7, and suppose that we suspect that component `Add-1` is the single element at fault. We could try to test this conjecture by enumerating likely faults, constructing intensional representations for each resulting circuit, and then simulating them to see if they produce the observed behavior. But we would be better off using a simulation technique that operates on classes of circuits at once.

A class simulator could be constructed using the sorts of extensional techniques described above. But Davis presents a technique called *constraint suspension* that models classes of circuits using an intensional, constraint-network representation like that used for individual circuits. This technique takes the constraint-network normally constructed for the circuit of figure 7 and locally modifies it to produce a network representing the class of circuits with the same schematic whose `Add-1` component is faulty. The resulting network can then be used in a simple simulation that determines whether there could be some member of the represented class which displays the observed behavior.

Note that Davis does not seem to use intensional representations for all classes of faulty circuits. For example, when speaking of identifying bridge faults, he uses the following highly extensional language:

> It is plausible to hypothesize a bridge fault between two modules A and B from two different tests if: in test 1, A produced an erroneous 0

*I will not consider the large portions of Davis (1984) that address issues that I am not concerned with in this paper.

†Davis makes a distinction between "forward" and "backward" constraint propagation that does not concern us here.

> and B produced a valid 0, and in test 2, A produced a valid 0 while B produced an erroneous 0.*

But the point here is that Davis gets a tremendous amount of leverage on the simulation problem just by using intensional representations for well-understood classes of interest: in this case circuits with a single faulty component.

Summary and Conclusions

I have reviewed the use of representations in a variety of simulations. I have noticed the following commonality:

- Individuals tend to have domain-dependent, intensional representations. Operations on individuals are represented as algorithms that manipulate the representations of their targets.
- Classes tend to have domain-independent, extensional representations. Operations on classes are represented as declarations in the group representation language that modify the extension of the group.

I have claimed that extensional class representations are used for want of the domain understanding necessary to produce intensional representations, and I have given examples from the reviewed papers of the use and advantages of the occasional intensional class representation. I believe that, to a large extent, representations in simulation systems evolve from extensional to intensional:

- When a domain is first explored, its objects and processes are not well understood. Observed characteristics of objects and processes are listed and used to generate extensional representations for both.
- As further study reveals patterns unifying various observed characteristics, the extensional representations of individual domain objects are supplanted by intensional representations which reveal the structure of the objects they model. This allows the use of operational definitions for processes.
- Good representations for individuals and operations allow exact simulation, which facilitates work on planning and prediction in the domain. These tasks require classification of individuals in a manner related to the domains and ranges of operations. Initial observations about natural classes gives rise to an extensional class representation.
- Extensive experience with simulation reveals the nature of interesting classes of individuals, leading to intensional representations for these classes.

*Davis (1984), p. 28.

The simulations reviewed here are all in domains whose individuals are well understood but whose classes in general are not. In fact, the papers display a range of levels of understanding of classes: circuit classes are understood most clearly, then enzymatic reaction classes, and finally classes of rock formations. Interestingly, one of the main concerns of both Weld's and Davis's research is how to do effective prediction by making the best use of a single well-understood class. This concern is predictable given the evolutionary outline just presented.

Future Research

I want to close by mentioning a line of research suggested by the perspective I have take here. I claim that extensional representations are used for classes because they allow the use of partial specifications, particularly those based on observations of hitherto unrelated characteristics. But a variety of extensional representations are available, such as the various logics, rule-based expert systems, and collections of heuristics. Which one shall we use?

To answer this question, I believe we need to examine the process by which we debug our extensional definitions. Both Simmons and Weld describe errors in their simulations that arise from incorrect classifications. We would like the representation system we use to give us as much help as possible correcting such errors.

Unfortunately, the nature of this process is not very well understood, nor are the techniques that can be used. Many cases of such debugging are documented; for example, Simmons mentions that instances where his qualitative and quantitative simulations disagree are helpful in correcting his qualitative process definitions, and much literature exists about debugging rule-based expert systems. But so far no reviews have been done that focus on the debugging of extensional *classifications*, especially as it relates to simulation problems. Work in this area is needed before the basis for choosing a representation can be anything other than esthetic.

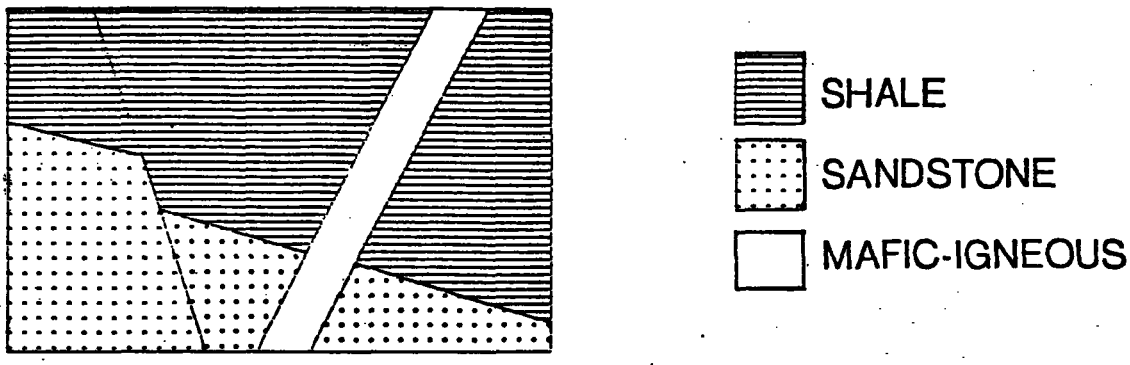


Figure 1. A rock-formation diagram such as that interpreted by Simmon's geology system. The roughly-vertical slice of igneous rock piercing both the shale and sandstone formations suggests an intrusion process. The fact that the sandstone is under the shale indicates that the sandstone was deposited first, the shale second. [Illustration from Simmons (1983).]

1. Deposit `sandstone`.
2. Deposit `shale`.
3. Uplift.
4. Intrude `mafic-igneous` through `sandstone` and `shale`.
5. Tilt.
6. Fault.
7. Erode `shale` and `mafic-igneous`.

Figure 2. A sequence of geologic processes which would produce the formation of figure 1. This sequence does not give the quantitative parameters, such as width of intrusion and angle of tilt, which are deduced by Simmons's system. [Illustration adapted from Simmons (1983).]

`Erosion (Elevel, Eboundary)`

1. Draw a horizonal line at `Elevel`.
2. Erase all parts of the line which do not cut across a face corresponding to a rock-unit.
3. Erase all faces above the horizontal line.
4. The edges corresponding to `SURFACE`—the surface of the Earth—are all the old `SURFACE` edges which were not erased, plus the remaining edges of the horizontal line.
5. The edges corresponding to `Eboundary`—the erosional boundary—are the remaining edges of the horizontal line.

Figure 3. Simmons's quantitative process definition of erosion. The intensionality of the underlying diagram representation allows this definition to be completely operational: it works by altering the structure of the underlying object representation. [Illustration adapted from Simmons (1983).]

```
EROSION
INTERVAL      I : temporal-interval
PRECONDITIONS (> SURFACE.top.height@I_start SEA-LEVEL)
PARAMETERS    ELEVEL : real
AFFECTED      C-PART : (set-of rock-unit), C-ALL : (set-of rock-unit),
              SURFACE
CREATED       BA : boundary
EFFECTS       (change = BA.side-1 {THE-AIR} I EROSION)
              (change = BA.side-2 C-PART I EROSION)
              (change = BA.orientation 0.0 I EROSION)
              (change = SURFACE.orientation
                     (efn3 ELEVEL SURFACE@I_start) I EROSION)
              (change function SURFACE.top (efn2 ELEVEL) I EROSION)
              (change function SURFACE.bottom (efn2 ELEVEL) I EROSION)
              (change function SURFACE.location (efn2 ELEVEL) I EROSION)
              (for-all c1 ∈ C-PART
                  (and (change function c1.top (efn2 ELEVEL) I EROSION)
                       (change function c1.location (efn2 ELEVEL) I EROSION)
                       (change - c1.thickness
                          (* (efn1 c1 ELEVEL) (- c1.top.height@I_start ELEVEL))
                          I EROSION))
              (for-all c2 ∈ C-ALL (change = c2 ⊥ I EROSION))
RELATIONS     (> ELEVEL SEA-LEVEL)
              (= C-PART {r : rock-unit |
                           (and (exists-at r I_start)
                                (≥ r.top.height@I_start ELEVEL)
                                (< r.bottom.height@I_start ELEVEL))})
              (= C-ALL {r : rock-unit |
                         (and (exists-at r I_start)
                              (≥ r.bottom.height@I_start ELEVEL))})
              (⇒ (≥ SURFACE.bottom.height@I_start ELEVEL)
                 (= SURFACE.bottom.height@I_end SURFACE.top.height@I_end))
              (= SURFACE.top.height@I_end ELEVEL)
              (≤ SURFACE.bottom.height@I_end SURFACE.bottom.height@I_start)
              (for-all c1 ∈ C-PART
                 (and (= c1.top.height@I_end ELEVEL)
                      (≥ (efn1 c1 ELEVEL) 0.0)
                      (≤ (efn1 c1 ELEVEL) 1.0)))
```

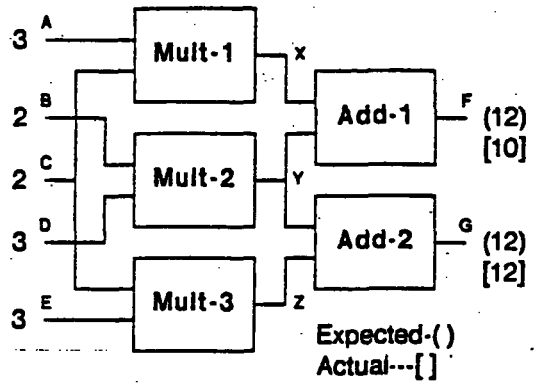
Figure 4. Simmons's qualitative process definition of erosion. The EFFECTS forms cause changes to the intensional rock-formation representations, while the RELATIONS forms are predications on those representations. Intuitively, the predications define the set of rock formations which could result from the erosion process; that is, they limit the possible extensions of the intensional representation.

```
(BIND ?binder>?site ?bindee>?handle)

(and (gt? (number-of ?binder) zero)
     (gt? (number-of ?bindee) zero)
     (not (bound? ?binder>?site ?molecule1))
     (not (bound? ?molecule2>?site2 ?bindee>?handle))
     (complementary? ?binder>?site ?bindee>?handle))

(bound? ?binder>?site ?bindee>?handle)
```

Figure 5. A discrete process representation from Weld (1984). The first form is a pattern used to bind the variables—those symbols starting with a question mark. The first predicate is a precondition on the state of the discrete world. The second predicate is an effect which is asserted when the operation completes. The assertion of the effect has intensional side effects in addition to its extensional effect.

Reaction 1	**Reaction 2**
(BIND E A)	(BIND E B)
(FOLD E)	(FOLD E)
(BIND E C)	(BIND E C)
(REACT C D)	(REACT C D)
(DROP E D)	(DROP E D)
(DROP E A)	(DROP E A)
(FOLD E)	(FOLD E)
repeat	*repeat*

Figure 6. Two discrete cycles which are aggregated to the same continuous individual. In the left-hand column, A causes E to fold so that it can catalyze the reaction C $\mapsto$ D; in the right-hand column it is B which does so. Since the cycles have no net influence on either A or B, their continuous images are identical.



A digital circuit. The expected output of Add-1 differs from the actual output, suggesting that Add-1 is bad. Davis (1984) uses *constraint suspension* to construct an intensional representation of the class of circuits in which Add-1 has a fault, and then does simulation using this representation.