Reasoning About Fluids Via Molecular Collections

John W. Collins
Qualitative Reasoning Group
Department of Computer Science
University of Illinois
1304 W. Springfield Avenue
Urbana, Illinois, 61801
Internet: Jcollins@uiuc. ARPA
Voice: (217) 333-6680

1 Introduction

Sometimes two distinct but interrelated views of an object or system are needed to reason about the physical world. For example, sometimes an engineer must think of "the liquid in the container" as an object (the *contained-liquid* ontology) while also reasoning about a hypothetical collection of molecules traveling together through the system as an object (the *piece-of-stuff* ontology).

As Hayes [4,5] notes, neither ontology alone suffices to explain commonsense reasoning about liquids. It is easy to reason about "the pressure at a portal" in the contained-liquid ontology, but impossible to explain the details of a thermodynamic cycle without following a "piece of stuff" through the system. The *piece-of-stuff* ontology, as we shall show, makes explicit the notions of continuity of space and conservation of matter, but provides no mechanism for reasoning about the overall behavior of the system.

This paper presents a technique for generating and reasoning with descriptions of fluids as "pieces of stuff". We introduce the *molecular collection* (MC) ontology as a specialization of Hayes' piece-of-stuff ontology. We claim that the MC ontology is *parasitic* on the Contained-Stuff ontology, in that a description of a system in terms of contained stuffs is a prerequisite to computing its description in MC terms. We show implemented rules for performing this computation, and illustrate their use with several examples. We argue that this representation provides a basis for more complex inferences, and discuss some open problems.

2 The theory of molecular collections

We begin by reviewing the original Hayes ontologies:

Contained-Liquid: Consider the liquid in a container as a single object. If the container is open then it is possible for liquid to leave the container and for new liquid to enter. Contained liquids have a continuous quantity `Amount-of` which may be influenced by various processes (flow, evaporation, condensation). They may disappear and reappear.

as when a cup of coffee is emptied and refilled. In this ontology the two cups of coffee are viewed as the same object.

Piece-of-Stuff: Consider a particular collection of molecules as a unit traveling around inside a system. The collection of molecules will have a fixed mass and a continuous position in space, which is influenced by its velocity, which in turn is influenced by various forces acting upon the object. A piece of stuff is never created or destroyed (assuming conservation of mass), so there are fewer problems of changing existence from this ontology.

It is straightforward to generalize the Contained-Liquid ontology into a *Contained-Stuff* ontology that describes gasses and allows multiple substances as well [1]. Qualitative Process theory [1,2,3] can be used to generate descriptions of contained stuffs, and we build on those descriptions.

In [5] no restriction is made as to the size of a piece of stuff. We obtain the *molecular collection* (MC) ontology by stipulating that the collection be so small that we can assume it is never distributed over more than one place (we return to this later). This tiny piece of stuff is viewed as a collection of molecules — as opposed to a single molecule — so that it may possess such macroscopic properties as temperature and pressure. Call the arbitrary collection of molecules to be considered as a unit MC.

Any ontology must divide the world into individuals: For reasoning it is important that the number of individuals be few. The Contained-Stuff ontology partitions a fluid system into a few discrete objects using the natural boundaries provided by containment. But the Contained-Stuff ontology fails to preserve molecular identity. Considering individual molecules would be prohibitive and unnecessary, since all the billions of them act more or less alike. By considering the possible behaviors of an anonymous collection of molecules, we constrain the possibilities for the whole by considering only one individual.

We claim that the molecular collection ontology is parasitic on the Contained-Stuff ontology. No one has succeeded in saying anything coherent about establishing the conditions for reasoning with the molecular collection ontology. We believe the reason for this failure is that the MC ontology alone is insufficient. Global information is required to identify what MC is doing. In classical physics the notion of gradient provides a local method for determining such motion. But establishing the gradient requires a global view of the physical system. The Contained-Stuff ontology provides this viewpoint for the MC ontology by establishing paths and conditions for flows and state changes. The reasoning based on molecular collections *starts* with the results of the Contained-Stuff description, and consequently does not have to re-derive those conclusions.

Consider the system shown in Figure 2. Figuring out how MC moves requires knowing the mass properties of the fluid, viewed with respect to the components of the system. Looking solely at MC, there is no way to establish the pressure differences between system components that imply the direction of flow. Although MC must play a role in the solution of the problem, the molecular collection ontology is inadequate. To determine facts like flow direction, the Contained-Stuff ontology must be used. Given a Contained-Stuff description, we can talk about pressure as a function of location rather than trying to find the pressure on an arbitrary collection of molecules. Here, a pump establishes a pressure gradient, causing a liquid flow into the boiler.

Figure 1: Sample rules for generating MC movements

These rules, associated with particular processes, describe how the MC's place and state change as a consequence of that process acting. Space limitations preclude showing the entire rule set.

```
If Flow(source, destination, path)
  then if Location(MC, source)
          Transition(MC, PLACE, path)
       if Location(MC, path)
          Transition(MC, PLACE, destination)

If Boiling(substance, container)
  then Transition(MC, STATE, GAS)

If Condensation(substance, container)
  then Transition(MC, STATE, LIQUID)
```

Our goal is to construct a history for MC, describing the sequence of places it is in and what is happening to it in those places. For our purposes, MC is uniquely defined by the place it is in, the substance of which it is composed,[1] and its current phase (i.e., solid, liquid or gas). The place is the container or fluid path in which MC resides.[2]

Constructing the MC history occurs in five steps. The first step is to feed the domain knowledge and the specific example through the Qualitative Process Engine (QPE) to generate the total envisionment for the given configuration. The total envisionment consists of all consistent situations connected by the possible transitions between them.[3]

In the second step, a single situation is selected for which the MC history is desired. In order for the history to be meaningful and interesting, the situation should involve some active processes and should last for an interval of time.

The third step finds the possible locations and states of MC and establishes how these properties can change. The critical observation is that each active process specifies a fragment of MC's history. Thus, the algorithm is linear in the number of active process instances, making it quite fast.[4] We can associate rules with each process to describe what its activity implies about changes in location or phase of MC, when located whithin the realm of influence of the process. For example, the rule associated with `liquid-flow` implies that when MC is in liquid form in the source, it can move into the path of the flow, and end up in the destination of the flow without changing state (see Figure 1). The rule associated with boiling implies that MC will undergo a liquid to gas phase transition within the same location. By combining these

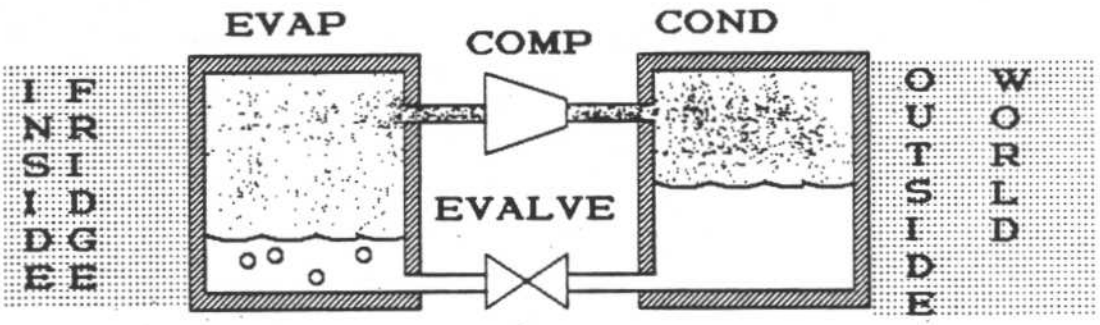
[1] In this paper only single substance systems are considered.

[2] Potentially, this could be refined through the use of some coordinate system such as submerged-depth or the length along a path.

[3] Each situation represents a unique set of active views and processes taken together with the signs of derivatives for all quantities.

[4] Running the rules over a total envisionment takes roughly one minute; constructing an MC history for any situation afterwards takes 5-10 seconds.

Figure 2: A refrigerator



partial histories, we can compute the full spatial extent of MC's travels and its associated phase transitions (if any).

In the fourth step, the Ds values for MC's quantities are computed.[5] By assumption, Ds[Amount-of(MC)] = 0. Pressure is simply inherited from the surrounding contained stuff. Changes in Heat, Temperature, Volume and Height are determined by rules associated with processes. For example, if the temperature of the destination of a liquid flow is greater than the temperature of the source, then both Heat and Temperature of MC will be increasing when MC reaches the destination. During boiling Heat is increasing and during condensation it is decreasing.

Finally, the fifth step constructs the graph defined by the relevant places and the possible movements between them. From this graph it is easy to recognize such phenomena as branching or cycles of flow. When branching occurs, the choice of which path to take will depend on the goal of the reasoning. Sometimes it is the properties of a specific path which are of interest, while in other cases all paths must be considered.

The relationship between the episodes in the MC history and the states of the envisionment is slightly complicated. One state in the envisionment can give rise to a number of episodes in the MC history. For example, the steady flow of working fluid in a refrigeration system would typically be described as a single state in the envisionment using the Contained-Stuff ontology. But viewed from the MC level, it will give rise to episodes involving heating, liquid-gas phase transition, compression, gas-liquid phase transition, etc.

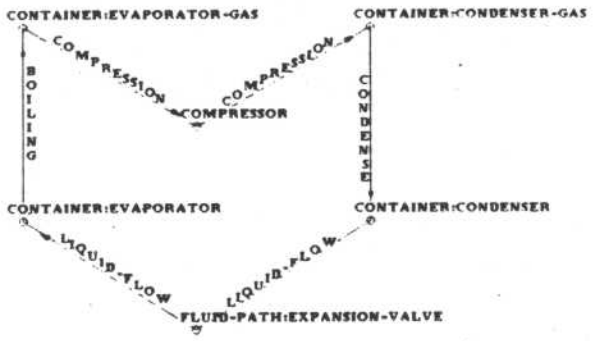
3 An Example

The MC-history generation algorithm has been tested on a number of examples of varying complexity. For concreteness we describe a representative example below.

3.1 A Refrigerator

[5] The Ds value of a quantity is the sign of its derivative.

Figure 3: The refrigerator MC history



Location	Evap	Evap	Comp	Cond	Cond	EValve
State	Liquid	Gas	Gas	Gas	Liquid	Liquid
Ds[Heat]	1	0	1	-1	0	-1
Ds[Temperature]	-1	0	1	1	0	-1
Ds[Pressure]	0	0	1	0	0	-1
Ds[Volume]	1	0	-1	-1	0	0
Ds[Height]	1	1	0	-1	-1	0

One of the motivations for looking at the MC ontology was to allow reasoning about complex thermodynamic cycles such as that used in a refrigerator. Figure 2 shows a simple refrigerator involving six seperate processes: two heat flows, two state changes (boiling and condensation), a compressor flow and a liquid flow. The situation selected for the MC history is the steady state, where all flows have equalized.

Figure 3 shows the MC history. MC boils in the evaporator and then is pumped through the compressor to the condenser, where it returns to the liquid phase and is finally forced through the expansion valve back into the evaporator. This representation provides the foundation for an important class of engineering conclusions. Since MC gains heat during boiling and loses it during condensation, it must be moving more heat through the compresser than returns via the expansion valve, so there is a net heat flow from the evaporator to the condenser. Thus the refrigerator is pumping heat *uphill* to a higher temperature.

4 Discussion

The ability to reason with multiple views of a situation provides significant advantages over using a single ontology. The Contained-Stuff ontology provides the conditions to determine which processes are active, and thereby determines the overall behavior of the system. The MC ontology provides the complementary ability to reason about where a piece of stuff came from and where it might go. We demonstrated that MC histories can be easily computed from QP models of fluids organized around Contained-Stuffs, and argued that this representation

provides the basis for several important engineering inferences (i.e., closed-cycles, recognition of heat pumps and differential analysis).

It is unclear whether or not growing an MC history across transitions between situations in the Contained-Stuff ontology is a good idea. If one is considering a liquid system that oscillates, for instance, then this could be necessary. However, most questions that arise in engineering concerning the MC history are about steady-state behavior, i.e., a single situation in the Contained-Stuff ontology.

It may be possible to generalize the MC ontology and history generation techniques to spatially extended pieces of stuff. This generalization would provide the ability to, for example, identify the spread of a contaminate through a fluid system.

We have only begun to explore the reasoning potential of the MC ontology. Currently we are implementing rules to calculate quantity space information involving MC parameters. Furthermore, we plan to augment the MC history by associating equations with each movement. These equations will be combined to yield quantitative descriptions of relevant system parameters, such as efficiency or work output per pound of working fluid. A differential qualitative (DQ) analysis could then be performed to identify how these parameters could be optimized, or in general how a change in one quantity will affect the behavior of the system.

5 Acknowledgements

Brian Falkenhainer, John Hogge, and Gordon Skorstad provided valuable comments, both in developing the theory and in writing the paper. John Hogge's ZGRAPH program has been invaluable for displaying the MC histories and total envisionment graphs. This research was supported by the National Aeronautics and Space Administration, Contract No. NASA NAG 9137, and by an IBM Faculty Development award.

References

[1] Forbus, K. "Qualitative Process Theory" *Artificial Intelligence*, **24**, 1984

[2] Forbus, K. "Qualitative Process Theory" MIT AI Lab Technical report No. 789, July, 1984.

[3] Forbus, K. "The Problem of Existence", Proceedings of the Cognitive Science Society, 1985.

[4] Hayes, P. "The Naive Physics Manifesto" in *Expert systems in the Micro-Electronic Age*, D. Michie (Ed.), Edinburgh University Press, 1979

[5] Hayes, P. "Naive Physics 1: Ontology for Liquids" in Hobbs, J. and Moore, B. (Eds.), *Formal Theories of the Commonsense World*, Ablex Publishing Corporation, 1985.