THE DECOMPOSITION OF A LARGE DOMAIN: REASONING ABOUT MACHINES

Craig Stanfill
Department of Computer Science
University of Maryland
College Park, Maryland 20742

ABSTRACT

The world of machines is divided into a hierarchy of seven sub-worlds, ranging from algebra to causality. Separate representations and experts are constructed for each sub-world; these experts are then integrated into an expert system. The result is Mack, a system which produces qualitative models of simple machines from purely geometric representations.

1. Introduction

Machine World is a universe consisting of simple mechanical devices. These devices operate according to the ideal gas laws plus Newton's laws of motion. Thus, we will be studying a universe in which the pressure of gasses causes things to move, and in which motion causes pressures to change.

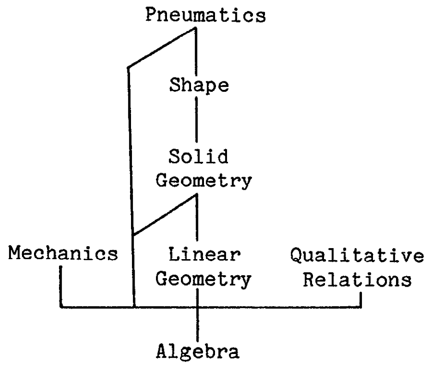
We will examine the process of understanding what a machine does. Specifically, we will start with a geometric description of the parts of a machine, and produce a qualitative description of how it works.

In this work, we will pay special attention to the division of Machine World into smaller sub-worlds as a means of controlling the complexity of the system.

2. Decomposition of Machine World

Mack's representations use 53 different primitives, support 201 different queries, and require 750 different rules. In order to control the complexity of the knowledge base, we decompose it into seven disjoint domains: Algebra, Linear Geometry, Solid Geometry, Shape, Mechanics, Pneumatics, and Qualitative Relations. Each domain is defined by a set of primitives and a set of queries, and is captured by a set of rules for answering these queries. These domains are related by a semantic hierarchy in which objects from high-level domains are defined in terms of objects from low-level domains. The experts for the high-level domains access the low-level domains through queries.

This research was funded by NASA's Goddard Spaceflight Center under contract NAS5-25764 and by NSF under grant NSFD-MC5-80-18294. Their support is gratefully acknowledged.



The Semantic Hierarchy

1. Algebra World represents numbers, algebraic expressions, and inequalities. The expert for this world is essentially a micro-MACSYMA [Bogen et al 75].
2. Linear Geometry World is a world of lines, points, and vectors represented by 3-tuples of algebraic expressions.
3. Solid Geometry World is a world of primitive solids and surfaces defined in terms of points, vectors, and algebraic expressions. For example, a cylinder is defined by two points and a radius.
4. Shape World represents shapes as the sum, difference, and intersection of primitive shapes. For example, a cylinder with a hole in it is represented as the difference of two cylinders.
5. Mechanics World represents accelerations, forces, and motions in terms of vectors and expressions. Several researchers ([Hayes 79], [Novak 76], [de Kleer 75]) have studied the representations and reasoning techniques which go into this domain.
6. Pneumatics World represents chambers (areas which contain gas) in terms of shapes and pressures.
7. Qualitative Relations World represents cause and effect as qualitative relations between variables. For example, the tendency of the

gas in a chamber to cause a piston to move is represented by a relation between the pressure of the gas and the acceleration of the piston. The formulation which is used by Mack is derived from Ken Forbus's Qualitative Process theory [Forbus 82] and other qualitative reasoning schemes ([Kuipers 82] [de Kleer 75]).

It is worth noting that many of the domains mentioned above have already been studied. These earlier systems are superior to Mack within their own fields; the problem which Mack solved was integrating these representations. This ability to build on existing representations is one advantage of Mack's domain oriented architecture.

Another advantage of this architecture is that it provides a good guide for the construction of the experts. For example, when we decide to create an Algebra expert, we have a good idea of what needs to go into it: we will need primitives for representing expressions, procedures for simplifying them, and some means of comparing them. Thus, we can determine what primitives we will need, what queries we will need to support, and we know when we have written enough rules.

Another advantage of this architecture is that it allows us to experiment with representations. There are many cases where, in the course of system development, we find that our representations are inadequate. Because the internal structure of the representations are hidden from the other domains, we need only modify the faulty domain.

Finally, by decomposing a domain we learn about its structure. For example, we decomposed Machine World into these seven sub-domains, and found that a simple relation - the semantic hierarchy - described its structure. We have also examined (in work not fully reported here) the 'fine structure' of some of these domains. For example, Mathematics is composed of a series of 'layers': Arithmetic, Algebra, Trigonometry, and so forth; each of these layers is an extension of the one that lies under it. We have also probed the role of foundational knowledge, such as knowledge about logic, by examining how such knowledge is integrated into the system. These topics are fully developed in [Stanfill 83].

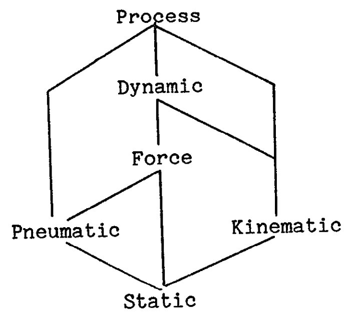
3. Comprehension

In order to comprehend a machine, Mack constructs a sequence of progressively more abstract descriptions of the machine. Specifically, Mack starts with a description of the shapes of the parts of the machine, and produces an abstract, qualitative model of the machine. To do this, it creates models describing motions, forces, chambers, and accelerations.

We implement this reasoning strategy as a set of model experts, each of which knows how to create a specific model. Model creation proceeds in four steps. First, the model expert obtains any low-level models on which it depends. Next, it asks representation experts to extract features from these lower level models. Next, the expert examines these features and constructs a set of objects for its model. Finally, the representation expert for these new objects simplifies the model.

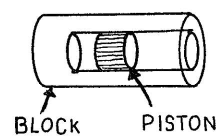
This architecture separates knowledge about how to create models from knowledge about representation. This is important for the conduct of experiments: we tend to modify the techniques used to create the models as we learn more about understanding machines, but the representations which we use and the queries which we ask about those representations are relatively stable.

We will now consider the actual sequence of models which Mack produces in the course of understanding a machine. This sequence of models is related by the following <u>epistemological hierarchy</u>, in which high-level models are inferred from low-level models.

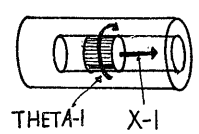


The Epistemological Hierarchy

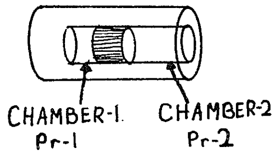
Mack begins with a description of shape of the parts of the machine. This is the <u>Static Model</u> of the machine.



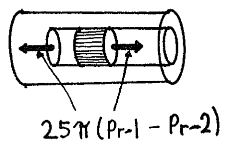
From this, it constructs a model of how the parts of the machine can move. These motions are derived from the manner in which these parts touch. Mack discovers that the piston touches the block in a cylindrical surface, which allows the it to slide left to right and to rotate about its axis. Mack then creates variables, X-1 and THETA-1, to parameterize these motions. The result is the <u>Kinematic Model</u> of the machine.



Mack next models the behavior of gasses in the Machine. Any area which is not occupied by a part of the machine is assumed to contain gas. Thus, Mack discovers two disjoint areas which contain gas, and models each of these as a separate chamber. The result is the Pneumatic Model of the machine.



Mack now constructs a model of forces due to the pressures of gasses. It does this by finding the surface where each part in the Static Model touches a chamber in the Pneumatic Model. For each such surface, it infers a force, the direction of which is determined by the orientation of the surface and the magnitude of which is determined by the area of the surface times the pressure in the chamber. Finally, the mechanics expert adds forces acting on the same object. The result is the Force Model.



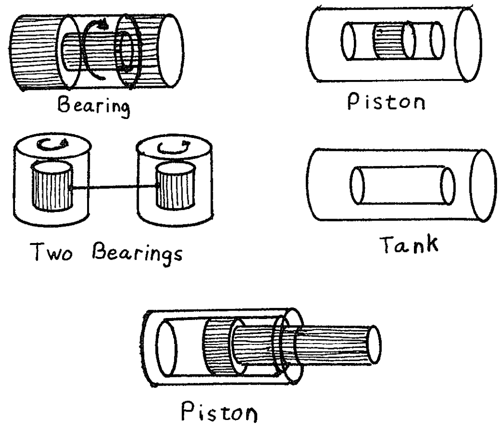
Mack now looks for accelerations. For each force in the Force Model, it discovers the motions (in the Kinematic Model) which it causes. In this case, Mack finds that the rightward-directed force on the piston causes the rightward motion of the piston, measured by X-1. Mack then models these accelerations. The result is the Acceleration Model.

$$X\text{-}1'' \sim (Pr\text{-}1 - Pr\text{-}2)\cdot 25\pi$$

Mack's final step is to extract qualitative relations from the Acceleration Model and the Pneumatic Model. It examines the Acceleration Model, and discovers that the rightward acceleration of the piston (X-1'') is positively influenced by the pressure to the left of the piston (Pr-1) and negatively influenced by the pressure of the Earth's atmosphere (Pr-2). Mack next examines the Kinematic and Pneumatic models and discovers how motions affect the volumes of chambers. It finds that, when the piston moves right (X-1 increases), the volume of the chamber to the left of the piston (Chamber-1) increases; this causes the pressure of the gas (Pr-1) to fall. Finally, it notes that the volume of the Earth's atmosphere (Chamber-2) is infinite, so its pressure (Pr-2) is constant. The result is the Process Model.

```
(Q+ Pr-1 X-1'')     (Q- X-1 Pr-1)
(Q- Pr-2 X-1'')     (Constant Pr-2)
```

This example took 5 minutes on a VAX 11/780 with 4 M-bytes of memory, running Franz Lisp under the Berkeley 4.1 Unix (tm) operating system. The following mechanisms have also been understood by Mack:



4. ACKNOWLEDGEMENTS

Thanks to Randy Trigg, Liz Allen, John Bane, and Mark Weiser for reading drafts of this paper, to the Maryland AI Group for general support, and especially to Chuck Rieger for advising me in this work.

APPENDIX

In the interest of readability, we have generally ommitted the actual representations which Mack used, and explained Mack's reasoning with pictures where practical. For those who want to see the actual representations, this appendix presents the actual input and output of Mack, unretouched except for the breaking up of long lines of output and similar textual adjustments.

Mack's input consists of a command to create a model. This command contains the name of the model, the name of the root of an expert hierarchy, and a set of symbols to be defined in the various sub-domains. The comand below defined the piston explained in the main part of the paper.

```
(model S1 master
        (lin3 p1 (POINT (0 0 0)))
        (lin3 p2 (POINT (10 0 0)))
        (lin3 p3 (POINT (20 0 0)))
        (lin3 p4 (POINT (30 0 0)))
        (lin3 p5 (POINT (70 0 0)))

        (geo cyl-a (CYLINDER {p1 p5} 15))
        (geo cyl-b (CYLINDER {p2 p5} 5))
        (geo cyl-c (CYLINDER {p3 p4} 5))

        (shape A (S-MINUS cyl-a cyl-b))
        (shape B cyl-c)

        (master machine {A B})
        )
```

Next, we gave the root-level expert the query "(all-models machine)". The following output was produced after 5 minutes of computation. The output consists of two parts. First, there is a list of all the models. Second, some symbols referenced in the first part are defined. As a detailed explanation of these representations is beyond the scope of this paper, the interested reader is referred to [Stanfill 83].

```
(Static
 machine

 Kinematic
 {(MOTIONS
     A
     B
     (SEQ
        ((TRANSLATION (VECTOR (1 0 0))) X-1)
        ((ROTATION
           (RAY (POINT (20 0 0))
                (VECTOR (1 0 0)))) THETA-1)
       ))}

 Pneumatic
 {(CHAMBER Shape-4 Pr-2)
  (CHAMBER
      (CYLINDER {(POINT (10 0 0))
                 (POINT (20 0 0))} 5)
      Pr-1
      )}

 Force
 {(FORCES
     B
     {(FORCE
         (POINT (30 0 0))
         (VECTOR
            ([A-PLUS [A-TIMES 25 PI Pr-2]
                     [A-TIMES -25 PI Pr-1]] 0 0)
            ))})
  (FORCES
     A
     {(FORCE
         (POINT (0 0 0))
         (VECTOR
            ([A-PLUS [A-TIMES -25 PI Pr-2]
                     [A-TIMES 25 PI Pr-1]] 0 0)
            ))})}

 Dynamic
 {(ACCELERATIONS
     A
     {(ACCELERATION
         X-1
         [A-PLUS [A-TIMES -25 PI Pr-2]
                 [A-TIMES 25 PI Pr-1]]
         )})
  (ACCELERATIONS
      B
      {(ACCELERATION
          X-1
          [A-PLUS [A-TIMES -25 PI Pr-2]
                  [A-TIMES 25 PI Pr-1]]
          )})}

 Qualitative
 {(CONST Pr-2)
  (I+ Pr-1 (d2 X-1))
  (I- X-1 Pr-1)})

(machine = {A B})

(B = (CYLINDER {(POINT (20 0 0))
                (POINT (30 0 0))} 5))

(A =
   (S-MINUS
      (CYLINDER {(POINT (0 0 0))
                 (POINT (70 0 0))} 15)
      (CYLINDER {(POINT (10 0 0))
                 (POINT (70 0 0))} 5)
      ))

(Shape-4
   =
   [S-PLUS
      (CYLINDER {(POINT (30 0 0))
                 (POINT (70 0 0))} 5)
      (S-COMPLEMENT
         (CYLINDER {(POINT (0 0 0))
                    (POINT (70 0 0))} 15)
         )])

(Pr-1 = (OPEN 0 pos-inf))
(X-1 = 0)
(Pr-2 = (OPEN 0 pos-inf))
(THETA-1 = 0)
```

REFERENCES

[Bogen et al 75]
Bogen, R., MACSYMA Reference Manual, Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1975.

[Forbus 82]
Forbus, K., Qualitative Process Theory, MIT AI Memo 664, February 1982.

[Hayes 79]
Hayes, Patrick, The Naive Physics Manifesto, in Expert Systems in the Microelectronic Age, ed. D. Michie, Edinburgh University Press, May 1979.

[Kuipers 82]
Kuipers, Benjamin, Getting the Envisionment Right, pp. 209-212 in Proceedings of AAAI-82, August 1982.

[Novak 76]
Novak, G., Computer Understanding of Physics Problems Stated in Natural Language, Technical Report NL-30, University of Texas at Austin, 1976.

[Stanfill 83]
Stanfill, C, PhD Thesis, in preparation, Department of Computer Science, University of Maryland, 1983.

[de Kleer 75]
de Kleer, J., Qualitative and Quantitative Knowledge in Classical Mechanics, M.I.T. AI-TR-352, 1975.