A REPRESENTATION FOR COMPLEX PHYSICAL DOMAINS

Sanjaya Addanki
IBM T.J. Watson Research Centre

Ernest Davis
New York University

Abstract

We are exploring a system, called PROMPT, that will be capable of reasoning from first principles and high level knowledge in complex, physical domains. Such problem-solving calls for a representation that will support the different analyses techniques required (e.g. differential, asymptotic, perturbation etc.). Efficiency considerations require that the representation also support heuristic control of reasoning techniques. This paper lays the ground work for our effort by briefly describing the ontology and the representation scheme of PROMPT. Our ontology allows reasoning about multiple pasts and different happenings in the same space-time. The ontology provides important distinctions between materials, objects, bulk and distributed abstractions among physical entities. We organise world knowledge into "prototypes" that are used to focus the reasoning process. Problem-solving involves reasoning with and modifying prototypes.

1 *Introduction*

PROMPT (Physical Reasoning by the Organisation and Modification of ProtoTypes) is a problem-solver operating in the world of devices: power supplies, printers, pens etc. The pilot study is in the world of pens and PROMPT is aimed at producing solutions to tasks such as:

- Design a pen that can write upside down.
- Why is my pen not writing on glass?
- What will happen to a pen at 200deg.F.?

Solving such problems requires reasoning from fundamental knowledge of the world (first principles, deep knowledge, basics etc.). Consider the task of designing a pen that writes upside down (i.e. on the ceiling). The solution to this problem requires fundamental reasoning about the effects of gravity on the ink and about forces that can reverse these effects.

Although pure forward chaining from first principles is an essential part of design verification, it is highly inefficient, as is pure backward chaining in open worlds. An efficient problem solver must heuristically control, and mix, fundamental and higher level reasoning to achieve reasonable performance. A powerful approach to effectively controlling fundamental and high-level reasoning is to provide a common substrate for the two by organising knowledge of the world (from first principles to higher level existing solutions) into hierarchies of structures called "prototypes".

Prototypes in PROMPT are descriptions of classes of elements rather than default structures, and solutions to problems often involve creating new prototypes by merging or modifying existing prototypes. The primary function of the prototype structure is to allow the heuristic control algorithms in PROMPT to *focus* the search and to *index* into the knowledge base. For example, assuming the system has a prototype for a ball-point pen, a heuristic algorithm for design might consist of:

1. Decide that the ball-point pen prototype is a reasonable starting point.
2. Diagnose why the ball-point pen prototype will not work upside down.
3. Solve the sub-goals (the defects noted above) by modifying the ball-point pen prototype.

Each of these steps is non-trivial and the focusing and indexing capabilities of the prototype structure are clearly advantageous. ([Addanki and Davis 85a] contains more details of this algorithm)

The idea of prototypes organising knowledge has a venerable history; the basic notion crops up under various guises: schema, frames (Minsky 75, Winston 78, Roberts and Goldstein 77, Bobrow and Winograd 77, etc.7.], natural kinds [Rosch 78], scripts [Schank and Abelson 77], templates [McCarty and Sridharan 81] etc.. Of these, only a few refer to modifications of prototypes, [Winston 78] in learning and [McCarty and Sridharan 81] in problem solving. The work most closely related to ours is [McCarty and Sridharan 81], in which a *fixed* set of transforms is used to generate modifications of prototypes in legal reasoning. However, expert problem solving often requires creating new modifications of prototypes and PROMPT is capable of *generating* transforms of prototypes from first principles.

Problem-solving in physical domains also requires an ontology that supports the different kinds of analyses required (e.g. differential, asymptotic, perturbation etc.). These analyses are greatly aided by an ontology that highlights the features of the elements appropriate to their roles in the analyses. Our ontology extends current ontologies of time and physical entities to provide such a framework.

The purpose of this paper is to briefly describe the foundations of our work in physical reasoning, namely, the ontology and representation of PROMPT.

2 *The Ontology.*

Our ontology of the world consists of time, entities, relations, and processes. Entities, relations, and processes are represented in prototype form. This section briefly sketches our ontology; a more complete account of our characterisation of time may be found in [Addanki and Davis 85a].

Time

Our temporal logic is based on the idea of mapping the continuous time line, isomorphic to R1, into the states of the world. The approach is similar to [Hayes 78 and McDermott 82]. A "state" is a snapshot of the world at a fixed time, specifying all the extant entities, their properties (including spatial properties), the relations between them, and the processes that affect them. A "scene" is a partial description of a state. A "history" is a function from an interval of time to scenes. In other words, a history specifies some of what is going on during some time period. Defining a history in this manner allows the system to reason about different "happenings" in the same space and time (e.g. current flow and heating in a resistor). A "chronicle" is a function from the real line to states; that is, a chronicle specifies all of what happens throughout time. In problem solving it is necessary to reason about alternative sequences of past and future states (diagnosis and planning). Hence we allow reasoning about multiple chronicles. An important predicate on chronicles is "physical possibility"; a chronicle is physically-possible if it observes known scientific laws.

Physical Entities.
Physical entities are the actual "things" that exist in the world, that take part in relations and processes. Entities are either generic or individual.

Generic entities include the following:

1. **Materials.** Any solid, liquid, or gaseous material which can be considered homogeneous at some scale of material (e.g. copper, water, mud etc.). The knowledge stored includes the physical properties of materials.
2. **Simple Homogeneous Objects.** Simple homogeneous objects are those that are composed entirely of the same material and are typically solid simple homogeneous objects (e.g. a screw, a cam etc.). While it is possible to reason about 10cc. of water as a simple homogeneous object, reasoning about IOcc. of oxygen introduces the necessity to specify the temperature and pressure of the gas, a compexity that we avoid for the present. Knowledge here includes shape, size, cost, etc..
3. **Assemblages.** An assemblage is an entitiy that is composed of solid, liquid, or gaseous simple homogeneous objects. A subclass of Assemblages is the class of Solid Objects [Hayes 78], that consists of those objects that are either uniform in material, or are composed of connected sub-parts that are themselves solid objects. Apart from shape, size, and decompositional information, assemblages point to functions and proofs of the functions served by the assemblage. Assemblages also point to the histories in which they take part. For example, the assemblage PEN points to the statement and proof that PEN's are used in the controlled deposition of ink on a surface. The assemblage also points to the description of the writing history.
4. **Bulk Physical Abstractions.** Types of physical entities that are associated with an assemblage as a whole (e.g. energy, momentum, mass, entropy, etc.). Knowledge here includes equations that help compute necessary parameter values.
5. **Distributed Physical Abstractions.** Physical entities defined at each point in a region (e.g. fields, temperature, density, etc.).

The systems entire knowledge of basic physics is stored under entities of the form 1, 4, or 5. For example, Newton's laws are indexed under FORCE in 4 and the material properties of copper are stored under COPPER in 1. Higher level, more specific knowledge is indexed under 2 and 3.

The categories of individual entities correspond to the kinds of generic entities. They include: (1) infinitesimal pieces of solids, liquids, or gases (e.g. WATER101); (2) individual simple objects (e.g. CAM53); individual assemblages (e.g. the SUN); individual physical abstractioni (e.g. KINETIC ENERGY(PEN53)); individual distributions (TEMPERATURE (INSIDE(BOX02))). Individual entities are individuated by their parameter values and positions as a function of time.

Relations,
Relations are predicates and properties of entities that hold between individual entities in a given state. For example two objects ABUT, a reservoir CONTAINS a liquid and a force APPLIES to an object. All our relations are generic.

Processes.
In principle, given the starting state of a system, it is possible to predict all subsequent behaviour from basic laws of physics. In practice, the need for efficiency in such prediction dictates the need to "precan" many known types of behaviours. Such pre-canned histories have been termed processes (Forbus 84]; examples are rolling, boiling , liquid flow and bouncing. The knowledge associated with these processes includes the entities involved, the dynamics of the history (what the entities do etc.), the conditions needed to start the history, the conditions needed to maintain the history, the effects of the history, and any equations associated with the history. Finally, we store a proof, from basic physics, that if the starting and maintenance conditions are met, the history will take place and have the stated effects.

Our representation of processes extends previous efforts by explicitly specifying the behaviour of the entities involved. In liquid flow, for example, part of this knowledge states that pieces of liquid translate along a channel. Such knowledge is essential in first principles reasoning about interactions such as the one between liquid flow and a turbine or strain-guage pressure sensor. The analysis of impulse and the consequent transfer of momentum is easy if LIQUID-FLOW is thought of as many pieces of liquid, each of which has a mass and velocity, impinging on the turbine or sensor. Allowing the size of the piece of liquid go to zero in such reasoning leads to the powerful techniques of differential calculus and analysis. While such techniques may not be necessary in "naive" reasoning, they are invaluable in the kinds of problem solving PROMPT is meant to do.

3 *The Prototype Representation.*

A prototype is a named chunk of information about an entity, relation, or process. Prototypes are organised into IS.A and PART.OF hierarchies. The knowledge inside a prototype is grouped into categories to allow rapid access to the different types of knowledge (e.g. preconditional, computational, etc.) required by a problem solver. The structure of a prototype consists of a tag that identifies the prototype as that of an entity, relation, or history, the name of the prototype, pointers to proofs and histories, and a selection of categories of knowledge about the element. The prototype of an element need not contain all the information about the element; elements may take part in descriptions that are specified only in prototypes of other elements.

Preconditional Knowledge: Entities, relations, and histories (collectively called elements) either exist at the beginning of the universe or they come into existence during some part of each chronicle. This history is the core of generating the entity, relation, or history and is a part of preconditional knowledge. For example, the precondition of a BPEN is: If "x" is a BPEN at time "t" in a physically-possible-chronicle "chr", then it MUST have been the case that the process PEN-MANUFACTURE took place during an interval in that chronicle and the end of the interval preceeded "t".

Definitions of elements often include constraints that must be satisfied if the element is to exist. For example, if SLIDING-FRICTION is to exist between x and y, it is required that x and y be solid and in contact, and that the relative motion and normal force between x and y be greater than zero.

Preconditions may be necessary (e.g. BPEN) or necessary and sufficient (e.g. SLIDING-FRICTION) or just sufficient.

Given a state of the world, the existence of an element constrains, through its preconditions, possible histories that led to this state. Also, given a state of the world, the existence of sufficient preconditions implies the existence of the element.

Definitional Knowledge: Abstract entities and relations are described by relations on parts of entities For example, STRICTLY-SPACE-CONNECTED is described by the fact that the opening of one of the objects is contained in the opening of the other, that the boundary of one of the openings is flush with the surface of the other, and the objects do not otherwise intersect. RIGID-ATTACHED says that the two objects maintain a fixed spatial relationship to each other and that a force on one object is transmitted to the other and that the two can be treated as one object. (Note the redundancy here. This is common in our prototype representation and desirable in problem solving.)

Processes are defined by the essential change during the process. For example, the essential change in liquid flow is that a piece of liquid changes position along a channel. This is one major extension we make to the representation of [Forbus 84]; the benefits of the extension were discussed earlier.

Physical knowledge: The counterpart of definitional knowledge for assemblages is the description of the physical properties of the objects. Physical descriptions of objects include properties such as size, shape, material, weight, and so on. The prototype of BPEN describes it to be solid, made of many materials, generally cylindrical, within a given mass range and so on.

An important gap in our system is the representation of shape. Any system that reasons about designs of devices must be able to represent and reason about shape, particularly if the devices are in any way mechanical. Shape representations are traditionally a hard problem and we have no easy solution (see [Davis 84] for some of the issues involved).

Physical knowledge is essential in any reasoning about objects. For example, attempting to determine if a component fits into a given space, or determining if a component can withstand a given physical load, requires physical knowledge of the component.

Structure: Entities bear structural relationships to other entities. For example, the reservoir of a cartridge CONTAINS ink, and the reservoir is ATTACHED to the tip. These relations form hierarchies of detail: At the next level of detail, the reservoir and tip-casing are GLUED (along a surface with a given glue). Processes can also bear structural relationships to each other. GAS-FLOW interacts with TURBINE-SPIN in the design of turbo-jets, and the destination of LIQUID-FLOW in the cartridge is the surface of a ball in BALL-ROLL.

Although it is possible to infer the interaction between LIQUID-FLOW and BALL-ROLLS from geometric knowledge, we see no reason to make the system artificially ignorant. Structural knowledge permits rapid deduction of many of the interactions within a device

Functional Knowledge: Components of devices take part in processes; i.e. they are bound to variables in the process definitions. Functional knowledge about a component is this binding information. For example, a piece of paper might be bound to the variable "surface" on which the ball rolls in the process BALL-ROLLS.

Modifying components of a device requires knowledge of the roles of the component in the functioning of the device. Functional knowledge allows the problem solver to identify those existing contraints that must continue to be met by the modified component by highlighting the processes in which the component participates.

Decompositional Knowledge: Elements of each of the primitive classes form PART.OF herarchies and this category of knowledge describes the place of the element in its PART.OF hierarchy.

Hirarchical Knowledge: Elements of the classes form IS.A (specialisation) hierarchies, and the place of an element within its hierachy is described by hierarchical knowledge. Again the examples are straightforward.

Continuation Conditions: Relations and processes continue to exist provided certain conditions are not violated. For example, ATTACHED continues to hold provided the force on the fastener is less than the strength of the fastener. Similarly, HEAT-FLOW continues as long as there is a temparature difference and a path for the flow of heat. Note that while continuation conditions appear similar to preconditions, they are not the fame. It might be a precondition to a wood fire that a match be held to firewood. It is certainly not a continuation condition that a match be held to the wood.

Continuation conditions are very useful in "limit critical point analysis" : A process ceases when one of its continuation conditions is false and the values of the variables involved in the condition being false are "critical points" of the variables with respect to this process. Given that a process is in execution, "critical point analysis" consists of analysing the behaviours of variables to see which "critical points" are reached and when they are reached. (This is an extension of "limit value analysis" of [Forbus 84].)

Computational Knowledge. Computational Knowledge consists of declarative representation of equations and algorithms that are used in various computations with entities. For example, we often use F = Ma and Friction = uN in mechanical computations. More complicated computations are those in fluid flow (Navier-Stokes equations) or Thermodynamics (the laws).

Expert problem solving in physical domains requires numerical solutions of many problems; having a declarative representation of the computations admits the capability for a future system to change its theories of the world!

4 Conclusions.

Initial efforts at implementing an envisioning system (the critiquer in PROMPT [Addanki and Davis 85b]) have clearly demonstrated the efficacy of the ontology and the prototype representation system in the heuristic control of reasoning. Briefly, our envisioning system consists of several weakly coupled analysers operating in parallel, that exchange results and newly-discovered information in order to constrain search. Our prototypes greatly simplify the required tight control of forward chaining and the ontology is very helpful in deciding how some newly discoverd piece of knowledge is to be disseminated.

Comparisons of PROMPT to QPT [Forbus 84] and Envisionment [DeKleer and Brown 83] are unfair to all systems because the purpose of PROMPT is different from that of the others. While QPT and Envisionment seek to reason at the naive level within "closed" worlds, PROMPT seeks to reason at the expert level from first principles in a world where these principles themselves may be changed by PROMPT. Although the current version of PROMPT does not change the basic rules, these rules are themselves prototypes and can be be modified! Worth pointing out however, is the explicit description of the behaviour of entities and the continuation conditions in PROMPT'S processes. These two allow PROMPT to analyse unanticipated interactions and to deduce "limit points" [Forbus 84] on the fly.

The first principles reasoning and the possible "openness" of the world resulted in important extensions to current ontologies. One thrust of the extensions was to provide a substrate for the analyses techniques required (e.g. differential analysis). The ability to reason about different happenings in the same space-time allows the system to reason about interacting/overlapping sub-systems separately, and later consider their interactions. Reasoning about multiple pasts is essential to PROMPT'S critiquing and diagnosing abilities. Further, we believe that the distinctions in the different types of physical entities (materials, assemblages, bulk, and distributed abstractions) will prove crucial to our work in physical reasoning.

Addanki, S. and Davis, E.S., PROMPT: Physical Reasoning from Prototypes, (in prep.).

Addanki, S. and Davis, E.S., Envisioning in Parallel, (in prep.)

Davis, E.S., Shape and Function of Solid Objects: Some Examples., TR xx The Courant Institute, New York University, 1984.

DeKleer, J. and Brown, J.S., A Qualitative Physics Based on Confluences., AI Journal, 24,1-3. 1984.

Bobrow, D.G. and Winograd, T., An Overview of KRL-0, a Knowledge Representation Language. Cognitive Science 1,1,1977.

Forbus, K.D., Qualitative Process Theory. AI Journal, 24,1-3, 1984.

Hayes, P J., The Naive Physics Manifesto. 1978

McDennott, D., A Temporal Logic for Reasoning about Processes and Plans. Cognitive Science, 6, 2, 1982.

Minsky, M., A Framework for Representing Knowledge. McGraw Hill, 1975.

Roberts, R.B. and Goldstein, I.P., The FRL Primer. AI memo 408, MIT,1977.

Roach, E. and Lloyd, B.B., Cognition and Categorisation. Lawrence Erlbaum Associates, 1978.

Schank, R.C. and Abelson, R.P.. Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Associates, 1977.

McCarty, L.T. and Sridharan, N.S., The Representation of an Evolving System of Legal Concepts. IJCAI 1981, vol. 1, pp246-253.

Winston, P.H.. Learning by Creating and Justifying Transfer Frames. Artificial Intelligence, 10, 2, 1978.

Appendix: The Pen World

An ordinary BIC-like pen has about eight parts, yet its function involves interactions between gravity, viscosity, cohesion, adhesion, friction, and simple forces; not to mention constraints on rigid objects and geometry. This property of incorporating complicated phenomena without having to represent a multiplicity of parts makes the pen world particularly attractive for investigations of physical reasoning.

The BIC-like Pen.

We assume that our pen consists of seven parts. The housing consists of three parts and the cartridge of four. The housing consists of a main body, a hollow truncated cone attached at the lower end, and a cap that is a disk with a cylindrical indentation. The cartridge consists of a main body, a shaft of narrower diameter and the tip. The cartridge is held in the housing because the indentation in the cap holds the upper end of the main body of the cartridge and while the shaft fits through the hole in truncated cone of the housing the main body of the cartridge does not (see fig. 1.) The tip of the cartridge consists of a hollow truncated cone that contains the ball. The ball is restrained on the upper end by the restraint.

The functioning of the pen, with little idealisation, is as follows: Gravity pulls the ink in the cartridge down into the tip and into contact with the ball. When the ball is pressed against a rough surface (by a force on the housing) and moved in the horizontal plane the ball rolls because the friction between ball and surface is greater than the friction between ball and tip. (Also the ink tends to provide a certain measure of lubrication.) A layer of ink in contact with the ball sticks to the ball because the adhesion between ball and ink is greater than the cohesion of the ink. This ink is transferred, by the rolling ball, through the gaps between ball and tip, to the paper. (Under normal circumstances the ink does not leak through these gaps because the viscosity of the ink is too high to allow the ink to flow through the small gaps.)

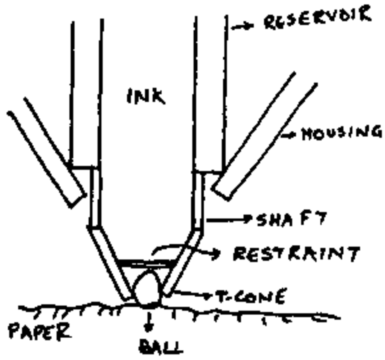

Figure 1. **Details of Ball-Point Pen Tip.**