EDISON: AN ENGINEERING DESIGN INVENTION SYSTEM OPERATING NAIVELY

Michael G. Dyer
Margot Flowers
Jack Hodges

April 1986
CSD-860087

To appear in:

First International Conference on Applications of
Artificial Intelligence to Engineering Problems

Southampton, U.K., April 15-18, 1986

EDISON: An Engineering Design Invention System Operating Naively

Michael G. Dyer
Margot Flowers
Jack Hodges

Artificial Intelligence Laboratory
Computer Science Department
University of California at Los Angeles
Los Angeles, CA 90024

The goal of the EDISON project is to design a program capable of creating novel mechanical devices, by using knowledge of naive physical relationships, qualitative reasoning, planning, and discovery/invention heuristics applied to abstract devices organized and indexed in an episodic memory. The EDISON program operates in two modes: brainstorming mode and problem-solving mode. In problem-solving mode, a goal specification is given as input and EDISON attempts to achieve the goal through plan selection and sub-goal satisfaction. A goal specification can be to alter or improve a device. Devices are represented symbolically, and are reasoned upon by EDISON without performing numerical computations. In brainstorming mode, EDISON starts with a device recalled from memory, and attempts to create novel devices through processes of mutation, generalization and analogical reasoning. The devices EDISON manipulates consist of simple, everyday mechanisms, such as mousetraps, nail clippers, can openers and doors. A goal of the EDISON project is to gain computational insight into the processes of naive physical reasoning [Hayes 78] and invention [Lenat 76] which people exhibit. To do so, we must address a number of issues, including: (a) how devices are represented and manipulated without detailed mathematical reasoning, (b) how devices are organized, indexed, and retrieved from a personal, episodic memory of devices and experiences of device use, (c) how new devices are discovered or invented during problem solving and/or brainstorming, and (d) how the resulting inventions are assessed for their novelty and/or ingenuity.

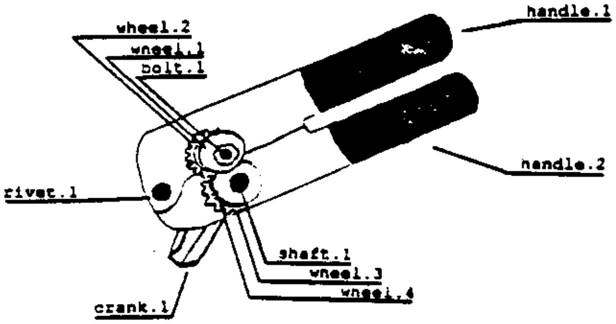


Figure 1: crank can opener

Consider the crank can opener (CCO) in Figure 1. People use this device on a daily basis to open cans, and most non-engineers, including children, will say that they "see" how it works after examining it for a moment. But from a computational point of view, a crank can opener is a complex object, and modeling the naive physical reasoning people employ to understand the structure and function of such a device is an extremely difficult task.

We will use this everyday device to illustrate many of the issues which must be addressed in EDISON. In this paper we discuss what is involved in building programs which can understand such devices in the naive, non-numerical way people do.

The EDISON model is comprised of six major components: device topology, naive physical knowledge, an episodic memory, a problem solver, strategies for invention and invention assessment using qualitative reasoning, and a device simulator.

1. Device Topology

Knowledge of device structure and function requires a symbolic representation of conceptual objects, object relationships, and processes. In Edison, mechanical devices are represented as a composition of *primitive mechanisms* and *simple physical objects* joined through spatial and connective relationships. Levers, springs and screws are examples of primitive mechanisms, while wheels and links are examples of simple physical objects. Mechanical devices are defined in terms of five components: *parts, spatial relationships, connectivity, functionality, and processes.*

1.1. Simple Physical Objects: Device Parts

Simple device parts are considered conceptual primitives in the sense that they need not be decomposed into more primitive elements. Such parts are defined in terms of their physical attributes (features), functions (goals they achieve) and uses, where each use is represented as a process.

Attributes include *physical-type, shape, size, region*, and *material-type*. Some attributes are scaled, such as size. Sizes are defined along dimensions according to shape (e.g. size for a cylindrical part would be defined in terms of its girth and height). In addition, objects have a reference size for comparison on a class basis. For instance, a large wheel on a child's toy car is smaller than a small wheel on an adult's motorcycle. Objects also contain regions.

Objects are organized into generic and specific classes. Generic objects possess a prototypic set of attributes, while specific objects may violate one of more attributes, as well as contain extra features. For example, a generic *wheel* has physical-type = slab, shape = circular, and functions = spin, roll. A specific wheel such as wheel.[1] in Figure 1 for example, includes the attributes: wheel-edge = sharp, and material = metal. Notice here that wheel-edge is a region of a wheel while the attribute of sharp is an abbreviation for the shape of this region both resembling and functioning as a wedge. Regions are represented in terms of locational primitives: *edge, end, side, surface, point*, and *center.*

EDISON currently recognizes seven kinds of parts: *wheel, ball, handle, container, linkage, shaft*, and *wedge*. From these primitive objects other, more complex objects are defined. For example, two *handle* objects, handle.1 and crank.1, are represented in Figure 2.

Both crank.1 and handle.1 are instances of handles, which are viewed as primitive devices. One function of a generic handle is to gain manual control of its associated device through the act of grasping, followed by a motion. Both bar-handles on the crank can opener (CCO) are rotated around a pivot through the action of squeezing, while the center of crank.1 is rotated about a pivot through the act of twisting. Both bar-handle.1 and crank.1 share a similar physical description. They differ in terms of their connectivity with the total CCO device, their subsequent use, and function.

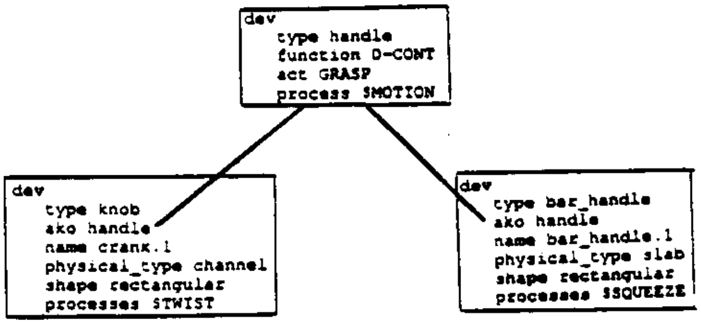


Figure 2: handle representations

EDISON currently recognizes five physical-types: *shaft*, *tube*, *channel*, *slab*, and *wire*. The physical-type attribute places objects in various classes, according to a rough description of their 3-dimensional shape. In EDISON, attributes themselves may be treated as conceptual objects and thus also contain attributes. For instance, physical-types have attributes associated with their regions; their cross-sections may take on shapes: *circular*, *triangular*, *square* and so on.

Materials are defined by attributes of *flexibility*, *strength*, *stiffness*, and *density*. Both materials and their attributes serve as enablement conditions for processes. Naive rules for reasoning over materials include: (1) For a magnet to adhere to an object, that object must have metallic material. (2) For object O1 to cut O2, the material hardness of O1 must be greater than that of O2.

Primitive parts are also defined in terms of function and use. The functions of a part describe the goals its various uses achieve. For instance, a wheel can function to move an object through the process of rolling, where rolling itself can be represented in terms of the process of rotation against the surface of an object.

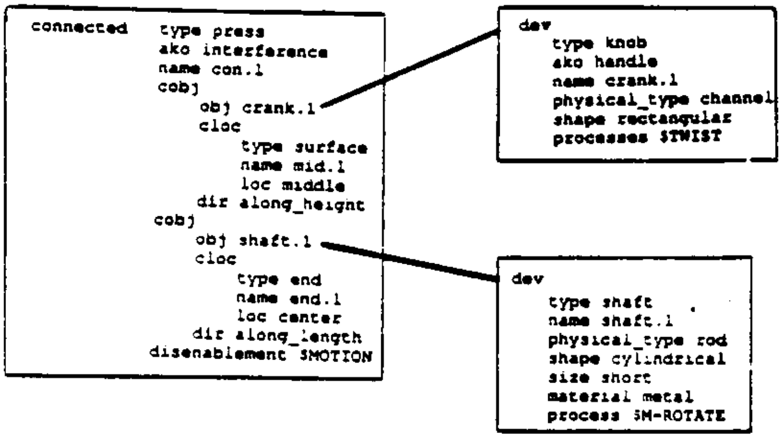
1.2. Device Spatial Relationships

In EDISON there are three spatial relationships between object parts: orientation, placement, and position. *Orientation* is essential for combining parts and devices into usable combinations. EDISON defines spatial orientation in terms of seven descriptors: *coaxial*, *colinear*, *adjacent*, *overlapping*, *offset*, *inside*, and *about*. In Figure 1, for instance, handle.1 is adjacent to handle.2. *Placement* and *position* are two constructs for local and global positioning. Placement defines reference on the object and position defines position in space. Local placement is defined very similarly to size. There is a reference point and a scale value which defines the point. Placement combines the region primitives defined previously with generally spatial locations of *upper*, *lower*, *left*, *right*, *inside*, *outside*. A position scale is used to define placement as well as position in space. The scale is discreet but covers the entire range for a particular reference point. For example, a pair of scissors is comprised of two handle parts, which have an adjacent orientation at the handle-end, and an overlapping orientation on the sharp edges. The placement of the pivot on the handles is midway between the handle end and the pointed end of each handle part. This set of spatial references is used along with conceptual part knowledge to make inferences concerning motion.

1.3. Device Connectivity

Devices are defined by the composition of parts, simpler devices, and primitive mechanisms. Composition can be viewed in either of two ways. On one hand, composition refers to the goal of connecting objects once they have been selected. On the other hand, composition means the state of objects being connected, which defines explicit relationships between objects and allows subsequent inferences to be made, based upon these relationships.

Connectivity defines a state between objects including: *type of connection, specific connector, object placement, connection direction*, and *process enablement*. For example, consider a connection within the crank can-opener (Figure 3).



```
connected   type press
            ako interference
            name con.1
            cobj
                obj crank.1
                cloc
                    type surface
                    name mid.1
                    loc middle
                dir along_height
            cobj
                obj shaft.1
                cloc
                    type end
                    name end.1
                    loc center
                dir along_length
            disenablement $MOTION
```

```
dev
    type knob
    ako handle
    name crank.1
    physical_type channel
    shape rectangular
    processes $TWIST
```

```
dev
    type shaft
    name shaft.1
    physical_type rod
    shape cylindrical
    size short
    material metal
    process $M-ROTATE
```

Figure 3: press connection

The axle is defined as a cylindrical part which is used primarily in rotation. Pressing is then defined as a connection between the crank and the axle, located at the center of the crank and the end of the axle. This example introduces direction as an attribute associated with connection. Directions for the axle are: *along_length, along_radius, along_tangent, about_length, about_radius*, and *about_tangent*. When two objects are connected, the local location and direction for each object must be defined so that inferences concerning transmission of force and motion may be inferred.

2. Naive Physical Reasoning

Naive physical reasoning refers to those symbolic processes which apply qualitative knowledge of device topology and function to achieve design, use, problem solving and invention goals. To account for how devices achieve goals, mechanical objects must also be represented in terms of the physical processes which govern their behavior. For example, a can opener achieves the goal of removing a lid through the process of cutting.

Processes are important for three reasons: (1) they can aid in the recognition of spatial and functional relationships, (2) they generate predictions concerning resulting relationships, and (3) they function in assessing the novelty of an invented device. When object combinations match mechanism topology and/or behavior, the combination should be recognized as an instance of the mechanism. For example, the CCO handles, handle.1 and handle.2, connected by a rivet should be recognized as a simple lever, and allow predictions based upon lever process relationships. In fact, almost any object can be viewed as a lever mechanism, if the process of its use matches that of a lever. One heuristic for determining novelty is that device processes have been altered. For instance, if we move a hinge down one inch on a door, we have technically created a new door. However, it is not novel, since the processes of opening the door remains unchanged. If we move both hinges to the upper edge, then we do have a novel door (i.e. the swinging trap door) since process-motion has changed.

2.1. Motion Processes

Motion processes in EDISON are similar to those described in [Forbus 83], in that all motion types are cases of motion in general. General motion is defined by an object and a direction. As long as the object is unconstrained in the direction, and as long as it has a non-zero speed in that direction, then the object will move. Subsequently, a motion like rolling adds the prerequisite that the object be round and in contact with some surface. These simple definitions require additions to the basic representation.

First, we require some notion of time. In EDISON, this information is kept to a minimum, since we do not believe that people maintain detailed temporal information about events. Temporal relations are inferred from processes attributes. For example, the crank on the CCO can be turned for quite some time with no notion of how long it takes to cut the can. People also have a hard time recalling how many turns of the crank are required to separate the lid from the can. However, they know the cutting process is complete when the point at which they started cutting is again back at the starting point.

Second, we require some notion of force to initiate physical processes. In EDISON there are two types of naive force: *push/pull*, and *friction*. Force is represented in terms of *amount*, *direction*, and *mode*. There are three modes representing how a force can be applied: *impact*, *periodic*, and *continuous*. When mode is specified, it includes a duration. The CCO handle.1, for instance, is pushed downward continuously for a duration to enable a cutting process.

Finally, naive classes of motion are required. EDISON recognizes four basic motion types: *sliding*, *spinning*, *rolling*, and *revolving*. Spinning involves rotation about a single pivot point. Rolling is represented as spinning plus continuous contact along a surface and the outer edge of a wheel. Revolving has no fixed pivot point and represents the motion of objects such as ball bearings.

2.2. Constraining Motion

Motion may be constrained in a number of ways. One way to control motion is through connection of device parts, where connectors are used to join, unite, or fasten objects. When paper is tacked to a bulletin board on a wall, the paper is no longer free to move anywhere, but is constrained to the location of the bulletin board by the connection. This implies that objects which are connected lose some mobility. The paper, being tacked to the board, can still swing in the breeze because the connection only pertains to the direction of contact. Moreover, the paper cannot move away from the wall because the bulletin board is mounted to the wall and so cannot, itself, move in that direction. Thus, mobility constraints are also inherited across connections. The nature of the inheritance depends on the type of connection and the objects involved.

EDISON recognizes three connection types: *interference*, *mechanical*, and *fusion*. The attributes used to classify a connection are its *permanence*, *strength*, and *variability* (or control). Here permanence refers both to how the connection affects the objects involved, as well as its portability. For example, connection via a magnet allows repeated reconnection through the removal of the magnet with a specified amount of pull. Fusion connections involve material changes of various sorts. Examples are welding, gluing, and adhesives. Mechanical connections involve a third object such as nailing, bolting, and clamping. For example, we represent the CCO handle connection as being riveted in the handle-height direction. Interference connections are those where contact between objects prevents motion. One example of interference is an object rolling inside a channel where its motion is contrained to the channel by channel walls. The major difference between fusion and mechanical connections, and interference connections, is that the latter do not inherit constraints, whereas they do transmit constraint and force across the boundary. A hammer, on the other hand, while in contact with a surface, cannot move in the direction of that surface but can otherwise move independently of the surface.

When two objects constrain one another many useful inferences can be made regarding their physical behavior:

> If connected(O1,O2) in direction Z, and O1 moves in direction Z,
> then O2 moves in direction Z.
>
> If connected(O1,O2) in direction Z, and O1 immobile in direction Z,
> then O2 immobile in direction Z.
>
> Overall device constraint is the union of individual constraints.

For example, one subgoal to opening a can is to hold the can while trying to cut it. The CCO achieves this subgoal with an interference connection between the can lip, wheel.1, and wheel.4, both along can length. Wheel.1 constrains can motion along_length (+) while wheel.4 constrains can motion along_length (-). The + and - signs are used here to point along a direction from a reference position.

2.3. Separation and Removal

Constraint and separation are inverse relationships. One is used to add constraints between objects, and the other is used to eliminate them. Separation processes serve the function of modifying objects by decomposing them into their parts. For example, if we need to remove a can's lid from a can we can use two separation process: splitting, following by continual cutting.

In principle there are many ways to separate mechanical objects (e.g. chemically, pneumatically, hydraulically, mechanically). EDISON is a mechanical model so only mechanical separation processes are being addressed. For instance, there are many ways in which people use the process of cutting (e.g. chopping, clipping, mowing, slicing); however, there are fundamentally only three ways to cut material: splitting, tearing, and shearing. Splitting is defined as a single object forcing a material apart via a wedge. Tearing results in motions of material in opposite directions. Shearing is similar to tearing in that two adjacent objects are moved in opposite directions with the material between them.

2.4. Transmission and Primitive Mechanisms

Motion processes describe the affects of pushing on an object, and connection processes alter devices through transmission of forces. There are three attributes of transmission in EDISON: *direction*, *speed*, and *amount* of force. Transmission is effected in all cases by the application of the principle of leverage, which states that there is a tradeoff between the effort applied to a task and the distance through which the task is applied (i.e. all levers transmit along their length). EDISON recognizes five classes of levers: *simple levers*, *gears*, *pulleys*, *shafts*, and *planes* (i.e. wedge-shaped objects). Simple levers transmit through linkages (e.g. crowbar). Shafts transmit from an outer edge to the center radially (e.g. steering wheel). Gears transmit from outer edge of one gear to the outer edge of another. Pulleys transmit by outer edge, and planes transmit along their surfaces. These lever types are called primitive mechanisms, and are comprised of the simple mechanical objects introduced in section 1 (e.g. linkage, wheel, rod, and wedge). By the combination of primitive mechanisms any spatial transfer can be effected (e.g. linear to linear, or linear to circular motions). In the CCO example, the primary function of the gear train (wheel.2 and wheel.3) to transmit rotational motion to wheel.1. Figure 4 depicts two types of transmission.

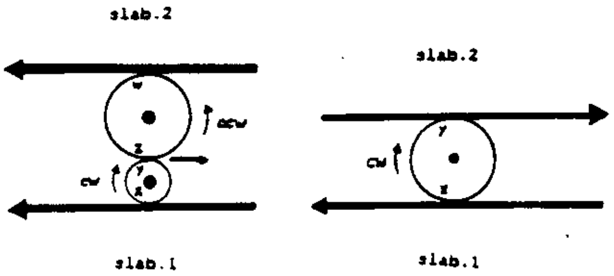


Figure 4: motion and leverage transmission a) single mechanism, b) combined mechanisms

In Figure 4.a motion by slab.1 causes motion at x in the same direction, through friction. Motion is translated into clockwise motion through the following naive physical rule about wheels.

> If O rotates and direction of motion of a location on O is Z
> Then a point on the opposite side of the
> axis of rotation will move in the opposite direction from Z

This results in motion at slab.2 in the opposite direction. Thus Figure 4.a depicts a mechanism for linear-circular transmission. In addition, transmission is transitive. Figure 4.b demonstrates one such transitivity relationship. This figure also shows a leverage mechanism, via wheel diameter ratios.

3. Episodic Memory

An important issue in EDISON is the development of an episodic memory where devices can be stored and retrieved for application to future problems. Key issues which must be addressed are: memory organization, indexing, levels of abstraction, and cross-contextual remindings.

3.1. Memory Organization and Indexing

As we have said, devices are represented in terms of primitive mechanisms, objects, connectivity relations, functions, and associated processes. The organization of memory reflects this decomposition. That is, devices can be retrieved from memory by using these attributes and conceptual objects as indices. When a new device is stored into memory (whether by the user, or through the process of invention), it is important that devices by organized by what they have in common. Devices organized under a generalized configuration of indices are then discriminated by their differences [Kolodner 84]. Consider Figure 5.

Here, both magnet and suction cup are accessible through indices describing contraint permanence since both objects allow repeated connection and separation without permanent change or material destruction. If the problem solver seeks this feature, both mechanisms can be retrieved. However, magnet and suction cup differ by principles of operation, since one uses a magnetic field and the other a vacuum. We do not want to consider a magnet to unclog a toilette but do want to recall a suction cup. In such a case, access via vacuum principle would retrieve the suction cup but not the magnet. That is, we want the memory to support problem solving by retrieving devices appropriate to the problem at hand.

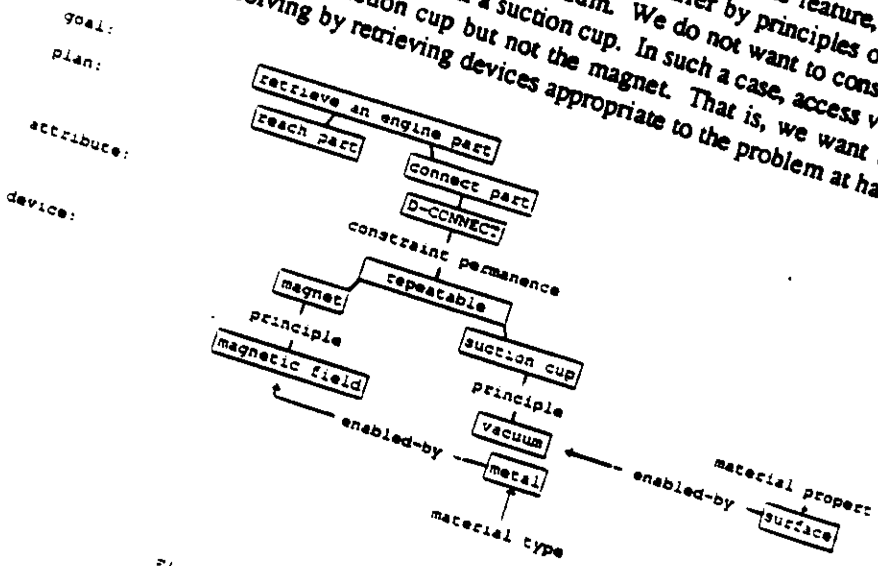


Figure 5: memory organization

In addition to indexing by function and topology, devices are also in
al context. For instance, can openers can be recalled within a
context is defined by the setting, physical constraints, and inv
associated with prototypic device use. Every device pot
contexts. Suppose the setting is a kitchen, the goal is
pose the nutcracker has been selected by the plann
lan fails. In this circumstance, an intelligent p
uld function as a nutcracker. This situation is

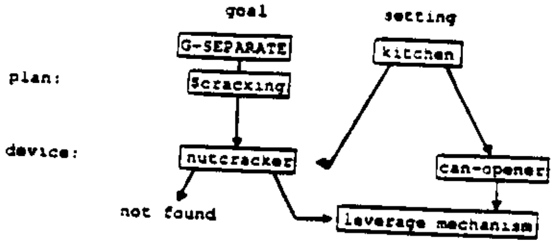


Figure 6: use of situational context

Upon realizing that the standard plan will not succeed due to device unavailability, the problem solver examines the nutcracker to see which physical mechanism is central to its function. The problem solver then realizes that it is the lever mechanism which applies force to crack the nut. Now the prolem solver is faced with a memory search task. There may be many devices in memory which contain levers, so indices must be generated for further discrimination. One important index is situational context. If the context is a kitchen, then kitchen devices will be retrieved and examined to see if they contain levers. In this way the crank can opener can be considered, since the handles function as a lever. In contrast, if the situational context is a tool shed, then a pair of pliers may be recalled first.

3.2. Levels of Abstraction and Cross-Contextual Remindings

Devices stored in memory must be indexed at varying levels of abstraction. For instance, at one level, a door functions to block passage of people. At a more abstract level, a door functions to block passage of any physical object. At this more general level, doors share functionality with umbrellas, which block passage of raindrops. Unless EDISON has both umbrella and door represented at higher levels of abstraction, it becomes impossible to retrieve from memory an umbrella as an object to consider when attempting to invent a novel door. Cross-contextual reminding [Schank 82] is important since it provides access to related and potentially useful memories. Remindings occur when memories are organized by similar indices. Once an umbrella is retrieved, information associated with it becomes available for invention. We can now imagine designing a door which is oriented vertically, and opens/closes in the way that umbrellas do.

In a memory containing many devices, it is important that only relevant devices be considered during problem solving or invention. If we are trying to invent a novel object which achieves the functionality of a door, then recalling a bicycle would do nothing more than increase the number of potential combinations to be examined. If we are trying to invent a novel mechanism to aid in opening or closing a door, then recalling the gear train mechanism of a bicycle might be relevant; so recall must be directed both by feature specifications at various levels of abstraction, and by current goals and context.

4. Planning and Invention

Problem solving in EDISON is achieved through a combination of planning and invention. People rarely begin a problem solving process by trying to be inventive. Rather, invention is usually thrust upon the inventor by the lack or insufficiency of existing devices. Given a goal, then, people first search memory for relevant existing plans/devices. Planning in EDISON is accomplished through standard artificial intelligence (AI) techniques of backward chaining and means-ends analysis. An example in which planning can be utilized directly is depicted in Figure 7, which shows how various can-openers are indexed via can-opening task.

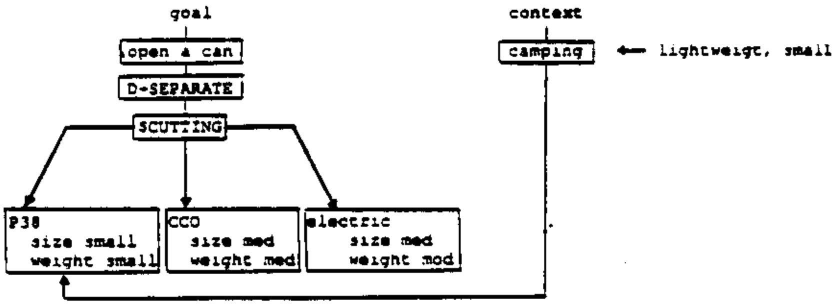


Figure 7: planning in episodic memory

As long as the device needed matches existing memory, the inventor need only recall the device, followed by planning to satisfy enablement conditions as sub-goals. A planner requiring a lightweight can opener, for example, would simply retrieve P38 (i.e. military can opener) rather than to inventing one. When the problem solver comes across a goal which cannot be decomposed to a known plan, then a novel plan must be invented.

4.1. Interference and Inspiration

Memory interference is the inappropriate intrusion of existing knowledge during a problem-solving process. A common example is parking a car. People park their cars so often that it is common to lose them altogether because the recall of previous parking places interferes with the recall of the present one. Interference is common in human memory because our memories are organized to generalize commonly used plans and to access more general knowledge first. This makes sense because more often than not we need generalized knowledge rather than specific memories. During invention, however, interference of commonly used devices is detrimental because the inventor is seeking novel and uncommon device combinations. According to [deBono 84], creative people are not hampered by memory interference; they either use their memories creatively, or ignore memories which are not useful. In EDISON, interference is partially circumvented through the use of heuristic strategies which redirect problem solving along more profitable directions:

> If no devices are accessed across contexts, then consider novel combinations of known mechanisms.

> If only familiar devices are retrieved during invention, then reject these devices

and activate a new goal for generating new retrieval indices

Consider the case where existing devices are recalled during invention. For example, suppose that we want to invent a new screwdriver which is more portable. We are reminded of many screwdrivers, and many ways of making something portable. This can lead to the invention of a screwdriver which keeps different blades in its handle. The second strategy is used to generate new ways of looking at the problem. For example, a new goal could be activated to release a constraint on the device and see if the resulting device gave any useful remindings.

In fact, the human heuristic of taking a walk or performing another activity when stumped on a problem, has a computational explanation. Consider an inventor who is blocked in the task of creating a new kind of door -- one that allows peple through either side simultaneously, but without individuals bumping into each other. The inventor decides to 'quit thinking' about the problem and take a walk. While walking along a river, the inventor sees a water wheel. Suddenly a solution comes to mind: the revolving door. In this case the inventor had an active goal to find a device which allows motion in opposite directions simultaneously. The water wheel could not be accessed directly, but when made available, served to satisfy this specification. At this point, the problem solver must take over, in order to transform the water wheel into a usable door. This includes changing the orientation from horizontal to vertical; scaling the size, and so on. Thus invention calls upon planning and memory, while the planner may call upon strategies of invention when standard plans fail. When the goal itself specifies that a novel solution be found, then strategies for generating novel indices and device combinations must be invoked.

4.2. Device Novelty and Ingenuity

Device assessment is a central issue in invention. EDISON judges devices along five dimensions: *utility*, *simplicity*, *efficiency*, *elegance*, and *novelty*. An ingenious device is one in which many of these dimensions are combined. Newly created devices must be assessed to determine how well a device has satisfied active goals. Utility is measured by goal achievement and constraint satisfaction. Simplicity is measured by topological attributes such as the number of parts, size or weight of a device. Elegance is determined by process size (number of steps) or structural complexity. Efficiency is measured in terms of forces required, while novelty is judged in terms of function or use, as compared to existing devices in memory.

The crank can opener, for example, is an ingenious device. Two novel attributes of the CCO are the use of a rotating cutter and a gear train to move the cutter. The efficiency results because it is easier to propagate a cut with a rotating cutter. Simplicity results because the mechanism providing motion to the gripper wheel is also used to move the cutter.

4.3. Strategies of Invention

Invention in problem solving is used to overcome incomplete or insufficient knowledge of the problem domain. When no prior solution exists invention is always required. Invention strategies also help overcome interference during problem solving by generating new indices for retrieval or recalling potentially relevant devices from other contexts.

EDISON considers three general strategies for creating novel devices: generalization, analogy and mutation, all of which rely on memory organization, indexing and retrieval. Generalization aids in determining the level of definition at which memory application will be most fruitful. Analogy is used to map devices across widely varying contexts, and mutation is used to modify existing devices, through alteration or combination. We have already discussed the importance of generalization and cross-contextual remindings.

Mutation: Device mutation is useful because specific device modifications may result in radically different object relationships. Mutation heuristics to achieve this purpose have been proposed by the authors [Dyer and Flowers 84] to guide the invention process. The types of heuristics proposed consider such strategies as the variation of attribute scale, function, or object number. For example, slicing a door creates two slabs, each covering half of a door frame. This operation results in a problem: the second slab is not connected to the frame. Problem solving results in the free slab being connected either to the hinged slab, or to the opposite side of the door frame. In one case we have invented the swinging barroom door; in the other case, the accordian door.

Analogy: Analogical reasoning consists of creating novel devices by recalling and adapting known devices across context. This is accomplished by finding similarities at high levels of abstraction and then using problem solving techniques to modify the recalled device to conform to the target context. This process was briefly discussed in the hypothetical invention of a revolving door.

5. Process Simulation

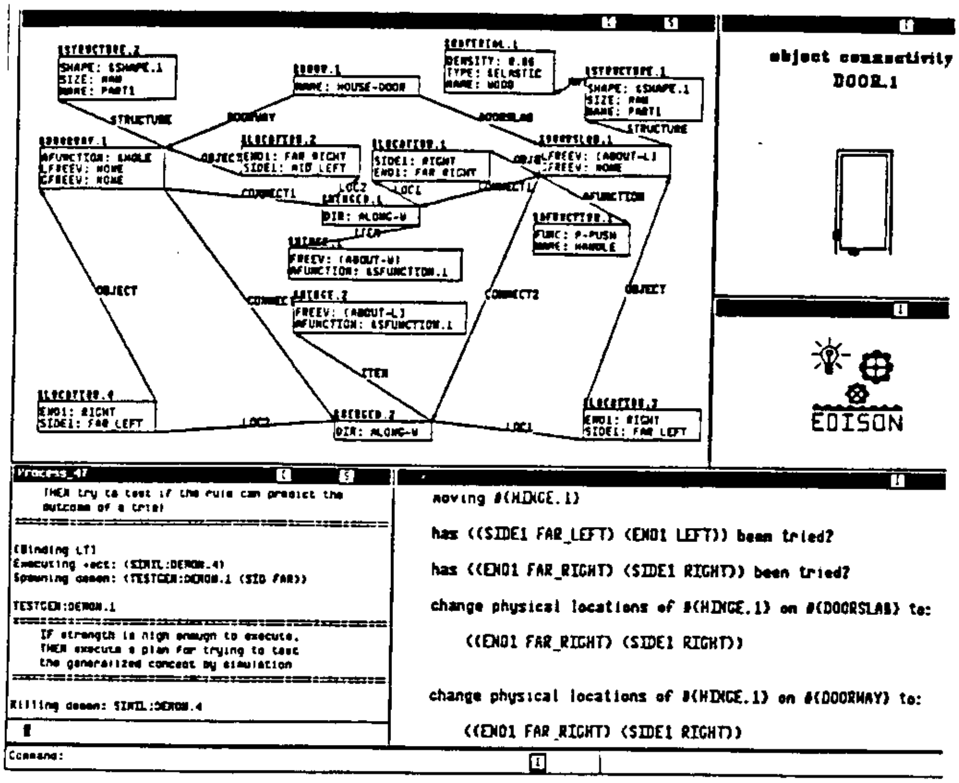
Device simulation is used by the problem solver for verification and experimentation/discovery. EDISON can discover constraints by actually "moving" a device component, and the simulator will reflect how the device actually behaves. For example, if EDISON knows nothing about hinge constraints it can learn by placing hinges both at the top and side of a door and learning that the door will not move by simulating a push.

6. Implementation Status and Future Work

The EDISON program is designed using RHAPSODY, an AI package containing: TLOG [Turner 84], a logic programming language, a demon package, a representation language, and message-passing semantics. RHAPSODY was developed in the UCLA Artificial Intelligence lab. Both EDISON and RHAPSODY are implemented in T, a scheme-based LISP [Rees and Adams 82]. The model runs on the UCLA Apollo Ring Network. RHAPSODY provides a way to declare frame-like representation structures, view and manipulate them graphically, and build episodic memories. Instance objects can be displayed and manipulated graphically via a "mouse".

Currently EDISON applies mutation heuristics to manipulate and create doors. The program is given a goal to experiment with door-hinge locations so as to learn about door-hinge constraints. The large window in Figure 8 shows a simple representation of door characteristics required for the manipulation.

The device (door.1) is decomposed to simple objects (door.1, doorway.1) and devices (hinge.1, hinge.2). Hinged.1 and hinged.2 refer to the connections between door.1 and doorway.1. Hinged connection, in this example, is achieved by a door-hinge, and the locations of connection are noted. Door-hinge location is represented as a reference and local scale value, where local scales are compared using a lookup table. A connectivity generator graphically displays the current device (upper right window in Figure 8).



```
    THEN try to test if the rule can predict the
    outcome of a trial
==================================================

(Binding LT)
Executing +act: (SIMIL:DEMON.4)
Spawning demon: (TESTGEN:DEMON.1 (SID FAR))

TESTGEN:DEMON.1
==================================================
    IF strength is high enough to execute.
    THEN execute a plan for trying to test
    the generalized concept by simulation
==================================================

Killing demon: SIMIL:DEMON.4
```

```
moving #(HINGE.1)

has ((SIDE1 FAR_LEFT) (END1 LEFT)) been tried?

has ((END1 FAR_RIGHT) (SIDE1 RIGHT)) been tried?

change physical locations of #(HINGE.1) on #(DOORSLAB) to:

    ((END1 FAR_RIGHT) (SIDE1 RIGHT))

change physical locations of #(HINGE.1) on #(DOORWAY) to:

    ((END1 FAR_RIGHT) (SIDE1 RIGHT))
```

Figure 8: typical EDISON window

The program has knowledge that door-hinges transmit rotational motion about the axis. In addition, the program knows about inheritance in connectivity: that objects connected to more than one rigid object are themselves rigid. Device dependent 'demons' [Dyer 83] are 'spawned' which check the device against motion preconditions. By comparing device motion capability to door-hinge placement the program can develop a concept of how door-hinge placement is constrained. Figure 8 also shows a door created which EDISON recognizes as being incapable of motion. People have no problem judging that this door cannot move, even without simulating it. We have argued that people use naive physical rules to draw such inferences about motion.

Currently, EDISON only applies mutation heuristics. This results in combinatoric problems and unfocussed invention. We plan to rectify these problems by increasing both the number and types of primitive mechanisms in memory. These will serve as a conceptual base for program device creation and combination. In addition, we intend to expand the problem solving heuristics and implement cross-contextual remindings and simple analogical reasoning strategies.

The graphical interface for displaying invented objects is very primitive at this point. The focus of the research, however, is on organization of memory, representation of devices, qualitative reasoning and invention. Consequently, an improved interface must be treated as a very long-term goal.

7. Comparison with Related Work

The work in EDISON has been inspired by that of Hayes [83], Forbus [83],

DeKleer/Brown [83], deBono [84], Schank [82], Kolodner [84], Lehnert [78], Rieger [75], Wasserman & Lebowitz [83], and Lenat [76] [83].

Hayes, Forbus: The philosophical introduction of the notion of naive physics in 1978 by Hayes has inspired most of the current research in this area. The concept of histories which Hayes proposed has been adopted in EDISON project for representing event process knowledge. Qualitative Process theory [Forbus 83] is an implementation of process histories proposed by Hayes. Like DeKleer and Brown, Forbus has been a proponent of specific device understanding apart from situational context in memory. Process knowledge is used to interpret and simulate specific devices. Although this notion is mandatory for developing accurate naive models, it assumes a supportive and developmental role in EDISON. The inventor bases his inventions on experience, and the relationships between his ideas, his goals, and physical objects. This inventor-based knowledge is central in EDISON. As a result, episodic memory plays an important role in EDISON. Proper episodic memory organization supports learning through experience.

DeKleer, Brown: The concept of envisionment used in [DeKleer and Brown 83] to determine device functionality is indicative of a method which is based entirely on the structure of an object. Envisioning requires the development of device models using what DeKleer calls confluence equations, which are transformed differential equations. These device models are used to develop a set of all possible event states for the device, the dependency relationships of which are determined with a state/space model. Solving confluence equations in this way is only one step removed from solving them quantitatively. People do not have representations for differential equations in memory. They cannot generate a complete set of any relationships, and they cannot remember long process chains. EDISON, on the other hand, reasons entirely symbolically and achieves a notion of functional relevance through relationships in memory. DeKleer and Brown's naive concept of a conduit, reminiscent of object primitives by Lehnert, is a valuable way in which to think of the transmission of force and motion symbolically.

deBono: Research on invention and creativity in children supports the theory of creativity and invention proposed in EDISON. deBono has studied children's understanding of relationships between concept and functionality through a series of design tasks. He has found that children rely heavily on the use of experience. They choose salient features of a problem and apply their preferences, and those of other actors, to the problem solving task. Many domains may be utilized in reminding the child of a direction in which to go, and the conceptual link to functionality occurs on many abstraction levels. deBono has addressed such concepts as abstraction scaling, speed, redundancy, interference, and pure design. He has noted that children utilize heuristic knowledge to a great extent. For example, children use 3 operators: *multiply*, *size*, and *transpose*, which are applied analogically to non-numeric attributes. Finally, deBono notes that child inventors are interested in effectiveness over simplicity, efficiency, or performance to a much greater degree than adults.

Schank, Kolodner: The theory of memory organization and cross-contextual reminding mentioned here is based upon the work of [Schank 32] and [Kolodner 84].

Lehnert, Lebowitz, Rieger: Work on the development of object primitives and relationships for comprehending the use of physical objects in natural language text [Lehnert 78], [Wasserman and Lebowitz 83], [Rieger 75] has been useful in addressing similar issues for representing mechanical devices.

Lenat: AM showed that heuristic models are a valid approach to creativity and discovery

[Lenat 76]. EURISKO [Lenat 83] was a program which created new heuristic knowledge and extended the power of a discovery/learning model.

8. Conclusions

The EDISON project is intended to create a model for experimenting computationally with processes of invention, analogy, and naive mechanical device representation. Studying how people utilize their memories to create and/or adapt devices is valuable in understanding how people design, and how memory is organized and applied to creative problem solving. We have argued for a theory of invention centered on an episodic memory-based understanding of device functionality, memory generalization, analogical reasoning, and symbolic representation.

deBono, E. *Children Solve Problems.* Penguin. New York, NY. 1980.

DeKleer, J. and J. Brown. A Qualitative Physics Based on Confluences. *Artificial Intelligence.* 24, 7-84, 1983.

Dyer, M. *In-Depth Understanding.* MIT Press, Cambridge, Massachusetts. 1983.

Dyer, M. and M. Flowers. Automating Design Invention. *Proceedings of Autofact 6 Conference.* Anaheim, CA. 1984.

Forbus, K. Qualitative Process Theory. *Artificial Intelligence.* 24, 85-168, 1983.

Hayes, P. The Second Naive Physics Manifesto. In Brachman & Levesque (eds). *Readings in Knowledge Representation.* Morgan Kaufmann, Los Altos, CA. 1985.

Kolodner, J. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Lawrence Erlbaum Associates, Hillsdale, NJ. 1984.

Lehnert, W. *The Process of Question Answering.* Lawrence Erlbaum Associates. Hillsdale, NJ. 1978

Lenat, D. *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search.* Stanford University PhD. Palo Alto, CA. 1976.

Lenat, D. EURISKO: A Program That Learns New Heuristics and Domain Concepts. *Artificial Intelligence.* 21(1,2), 61-98, 1983.

Rees, J. and Adams, N. T: A Dialect of LISP, or Lambda: The Ultimate Software Tool. *Proceedings of the 1982 ACM Symposium on LISP and Functional Programming.* August, 1982.

Rieger, C. An Organization of Knowledge for Problem Solving and Language Comprehension. In Brachman & Levesque (eds). *Readings in Knowledge Representation.* Morgan Kaufmann, Los Altos, CA. 1985.

Schank, R. *Dynamic Memory.* Cambridge University Press. Cambridge, England. 1982.

Wasserman, K. and M. Lebowitz. Representing Complex Physical Objects. *Cognition and Brain Theory.* 6(3), 259-285, 1983.