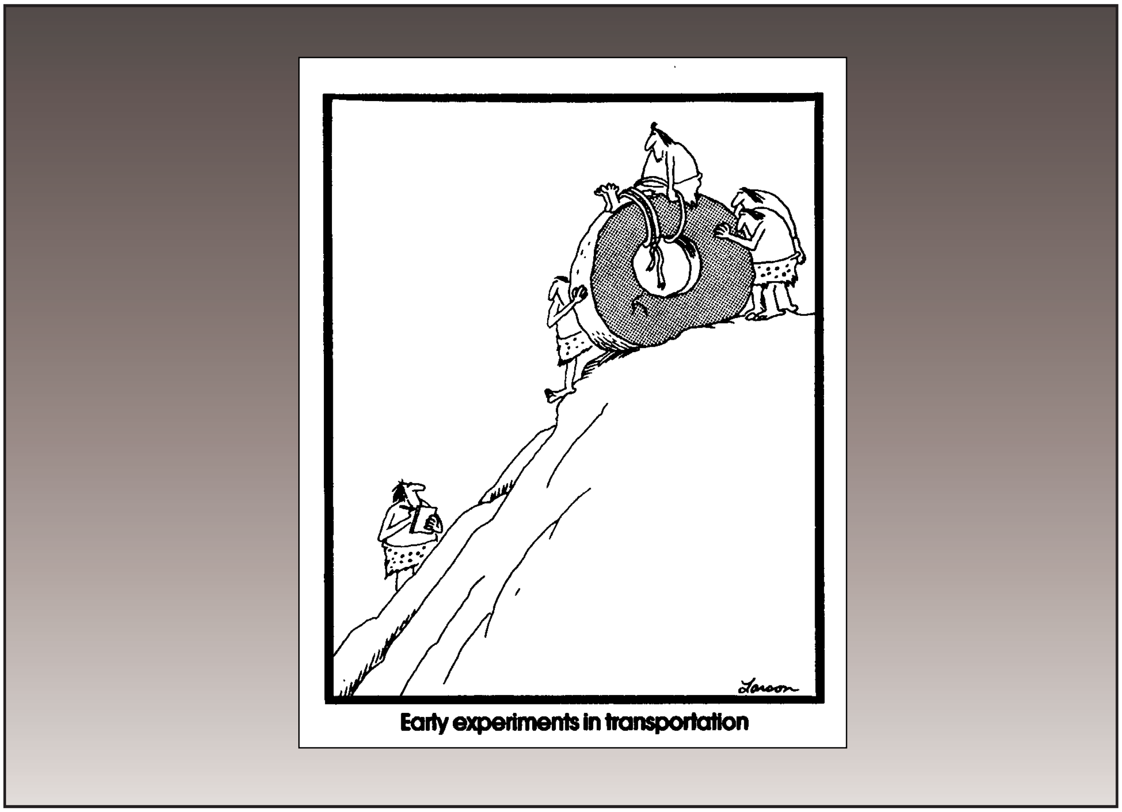


Early experiments in transportation

FAR SIDE copyright FARWORKS, INC. Reprinted with permission of UNIVERSAL PRESS SYNDICATE. All rights reserved.

○
○
○
○

Chapter 4

High-Level Dynamics and Device Function

In this chapter, the representation constructs which are used in FONM to describe high-level device dynamics are presented. Every device and device function can be described in terms of a finite set of objects which have associated with them a specific function. These *machine primitives* comprise a set of functional building blocks for representing complex devices. The functions of these primitives are represented with behavioral sequences, so the discussion begins by defining a structure for function which is based on device behavior, and which is based on the physical requirements of objects which can instantiate them. The discussion ends with the presentation of dynamic representations for six devices of increasing complexity. Three topics are addressed in this chapter:

(1) Device function
(2) Machine primitives (MPs)
(3) MP combinations and mechanical device representation

4.1 Device Function

A *function* describes *what* a device does and is invoked by perturbing a device. For example, two boards can be connected with screws using a screwdriver. The function of the screwdriver is to provide the necessary force needed to drive the screw into the boards. The driving function is invoked by turning the screwdriver when it is in contact with a screw, and results in a magnification of the rotational force applied at the handle to the screw head. A device may instantiate different functions, depending on what force is applied, to what region of the device the force is applied, and the device statics. The screwdriver can be used to punch holes in oil cans, or to pry staples from documents. Driving, punching, and prying are each invoked by applying a different type of force to the same object. Each function makes use of different aspects of the device statics, and each produces a different result. In each case, the overall function of the device can be represented as a combination of the functions of its components, and they, in turn, can be represented as BPP sequences.

The shaded portions of Figs. 3.31 - 3.33 represent the underlying behavioral sequence of the screwdriver components. If the shaded boxes are darkened, so that the details of the component behavior are not visible, as in Fig. 4.1, then the boxes represent the individual functions of the screwdriver components in the overall driving function. The only information entering the boxes are the enabling states from the applied force and from the boxes on the right, which represent the connectivity relations between the components (device statics). The only information leaving the boxes are the resulting states. This is called a *black box* description because it describes what happens when the device is perturbed, but not how it happens.

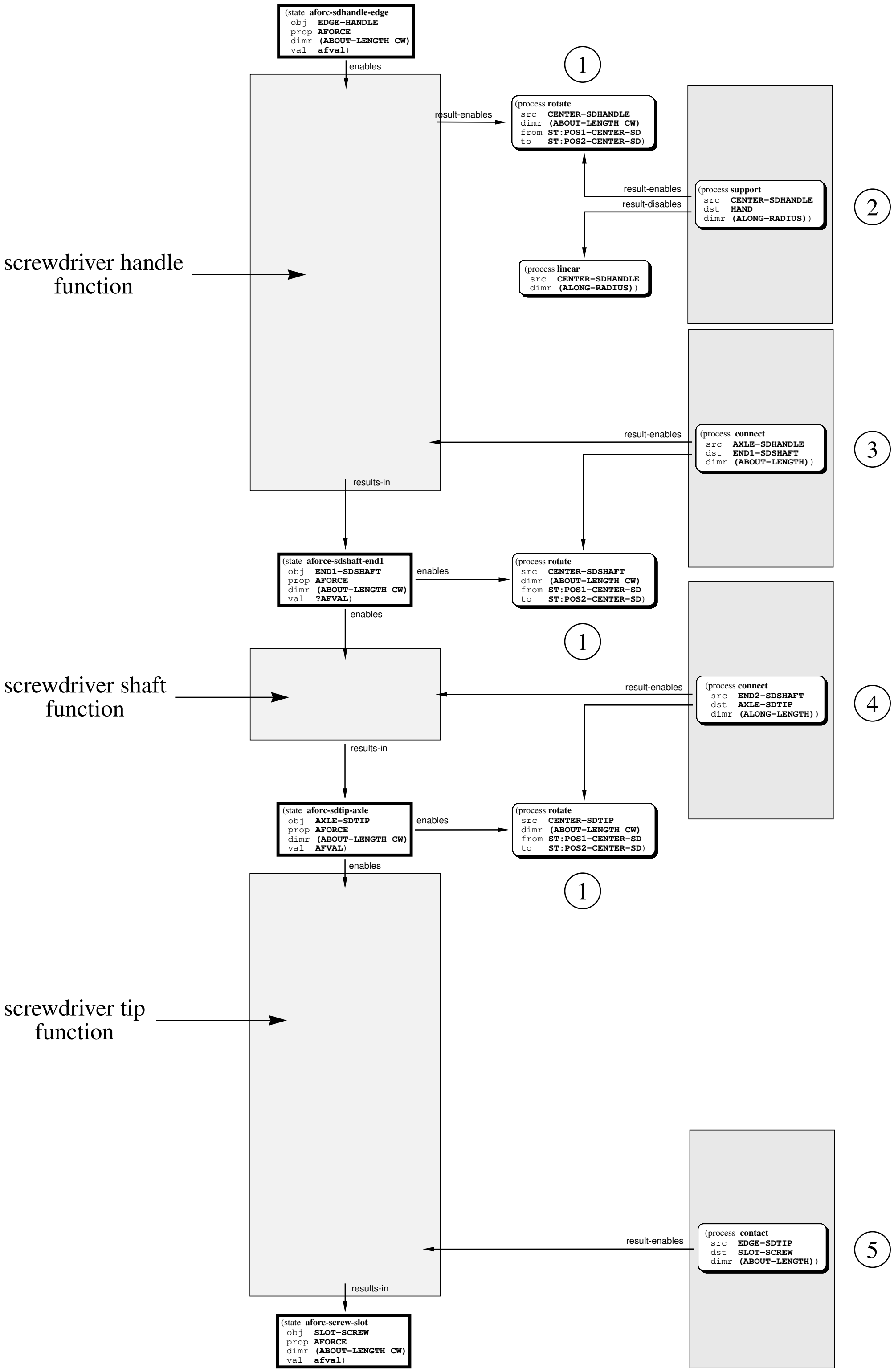


Figure 4.1 Screwdriver components in overall driving function, shown as black boxes.

The boxes representing the components functions describe *intra-object dynamics*, behavior which may not be observable. The motions at label (1) in Fig. 4.1 describe *inter-object dynamics*, behavior which is observable. Since each of the screwdriver components is rigidly connected (3 and 4) to the others, the enablement of rotation for the handle (2) enables rotation for the other components. The rotations depicted at (1) all represent rotational motion of the device and are illustrated to show that each component enables motion of the screwdriver.

In FONM, device function is represented as a black box, with a frame that describes a BPP sequence by the associated device and by the states which enable and result from the BPP sequence. In Fig. 4.2, the function knowledge structure is depicted on the right (at 1) and the associated BPP sequence is depicted on the left (2).

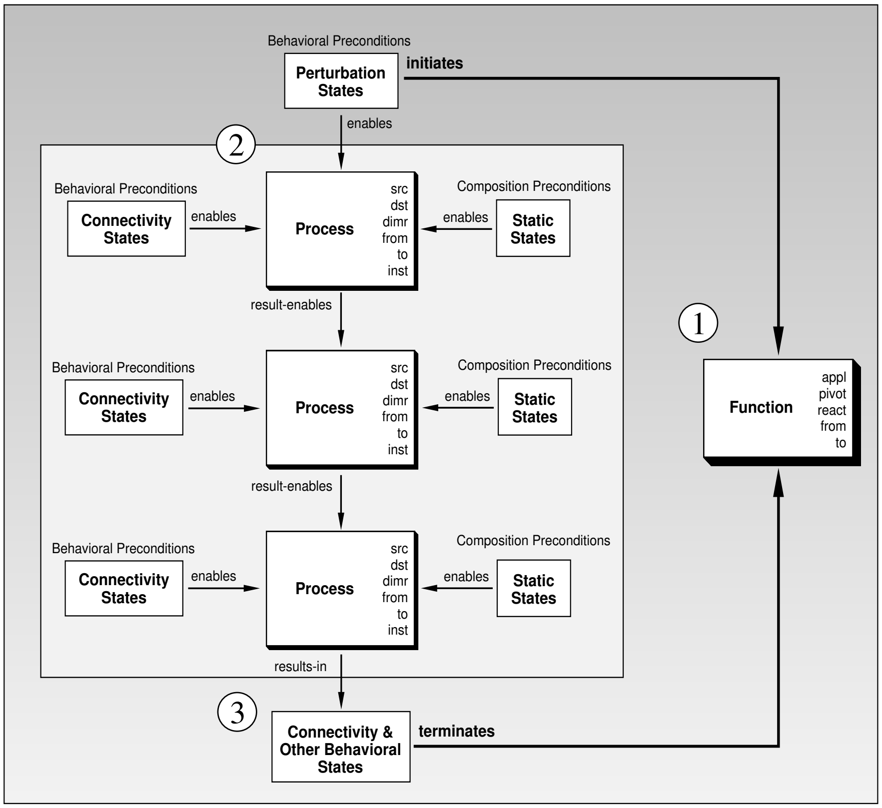
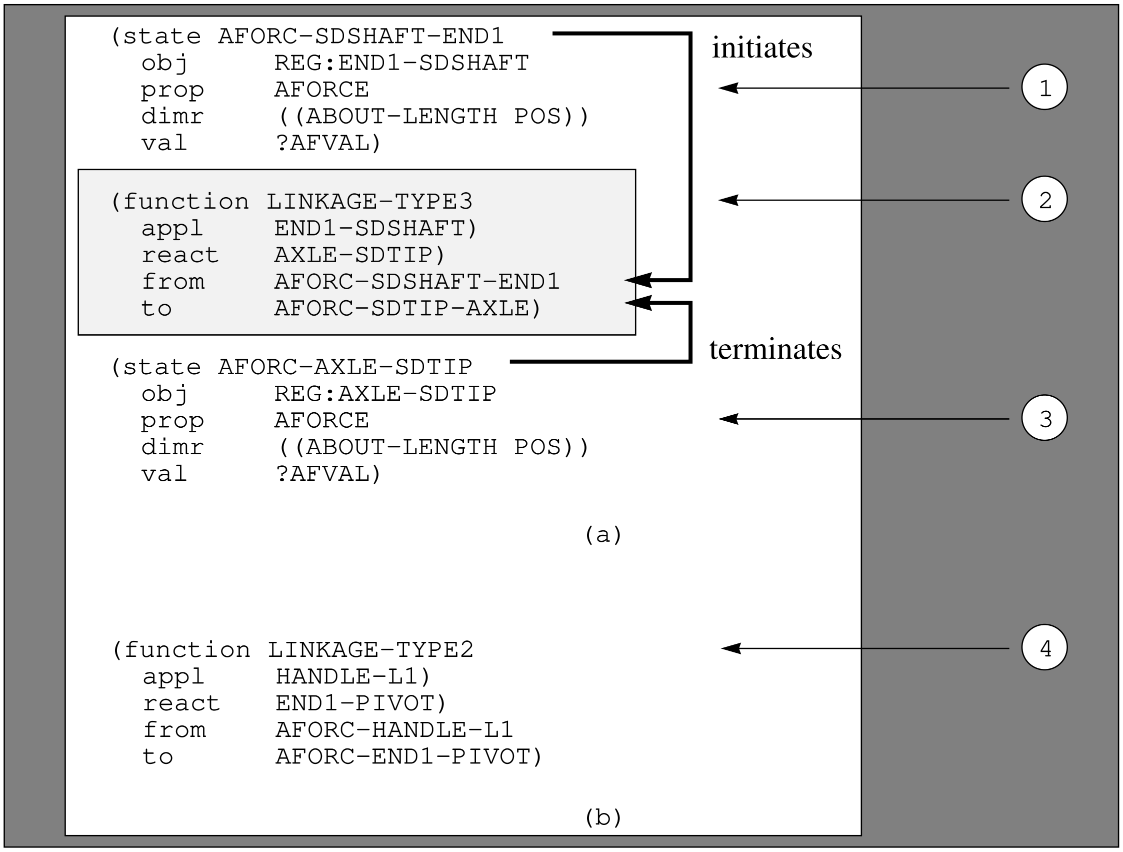


Figure 4.2 The relationship between device function and device behavior

The perturbation at 2 in Fig. 4.2 initiates the function. The connectivity between objects, and the static representation of the objects, enable the propagation of the perturbation states from one process to another. The behavioral state resulting from the BPP sequence (3) terminates the function. The black boxes depicted for screwdriver components in Fig. 4.1 have the same form as the left hand side of Fig. 4.2. Each box has an applied force which enables it, and a state resulting from it. These are identical to the initiating and terminating states of the function. Each box in Fig. 4.1 also has enablements based on the component's connectivity with the other components. In Fig. 4.1, the static enablements are separated from the component functions.

Device function is defined by the initial and final states, and not by the number of objects or processes needed to produce the final state. The function template in Fig. 4.2 is depicted as a three-process BPP sequence. A function can be comprised of a single process, such as the transmission of force by the screwdriver shaft in Fig. 3.31b (Page 113). A function can also have a number of processes where each is instantiated for the same object. For example, the function associated with slicing tomatoes with a knife can be represented with the BPP sequence RESTRAIN, TRANSFORM, STORE, and DEFORM. All four of the processes in this sequence are instantiated by the same (knife) object.

Device function, as depicted on the right hand side of Fig. 4.2, is represented with a knowledge structure having five roles: appl, react, pivot, from, and to. The function *appl*, *react*, and *pivot* roles describe region locations on the device where the primary interaction takes place. The *from* role describes the applied force which initiates the function, and the *to* role describes the state which terminates the function. As with object behavior, terminating states are local states, associated with the region at which the process takes place. The screwdriver shaft depicted in Fig. 3.31 represents a very simple function called LINKAGE, which describes force transmission and translation from one location to another. The LINKAGE function is shown instantiated for a screwdriver shaft in Fig. 4.3a. Another instantiation of LINKAGE is shown, for a nutcracker handle, in Fig. 4.3b. These device components differ in how the force is applied (rotational in the first case and translation in the second) and how the components are connected to other components (rigid in the first and pivoted in the second). In other respects these components are doing the same thing.



```
(state AFORC-SDSHAFT-END1
  obj      REG:END1-SDSHAFT
  prop     AFORCE
  dimr     ((ABOUT-LENGTH POS))
  val      ?AFVAL)

(function LINKAGE-TYPE3
  appl     END1-SDSHAFT)
  react    AXLE-SDTIP)
  from     AFORC-SDSHAFT-END1
  to       AFORC-SDTIP-AXLE)

(state AFORC-AXLE-SDTIP
  obj      REG:AXLE-SDTIP
  prop     AFORCE
  dimr     ((ABOUT-LENGTH POS))
  val      ?AFVAL)
```

```
(function LINKAGE-TYPE2
  appl     HANDLE-L1)
  react    END1-PIVOT)
  from     AFORC-HANDLE-L1
  to       AFORC-END1-PIVOT)
```

Figure 4.3 Instantiation of the LINKAGE function, (a) for a screwdriver shaft, (b) for a nutcracker handle.

LINKAGE represents the function associated with simple, uniaxial, force transmission: the extension of force from one location to another. In Fig. 4.3a, a rotational force AFORC-SDSHAFT-END1 is applied at region END1-SDSHAFT (1) and reacted at region AXLE-SDTIP (3). The function's *appl* role is filled by END1-SDSHAFT, because this is the location where the force is applied to the screwdriver shaft. The *react* role is similarly filled by AXLE-SDTIP, because this is the location where the function's output is produced. The *from* role is filled by the perturbation state AFORC-SDSHAFT-END1, because this is the force that initiates the function. Finally, the *to* role is filled by the state AFORC-AXLE-SDTIP, because this is the state which terminates the function.

4.2 Machine Primitives

LINKAGE is called a *machine primitive* (MP). Machine primitives represent the functions associated with primitive objects, and form a set of building blocks from which the functions of complex devices can be described and reasoned about. For example, LINKAGE represents any object which instantiates the function of uniaxial force transmission, and the associated objects are called linkage objects. There are eleven machine primitives in FONM: MP-LINKAGE, MP-LEVER, MP-WHEEL-AXLE, MP-GEAR, MP-PULLEY, MP-BEARING, MP-PLANE, MP-BLADE, MP-SCREW, MP-SPRING, and MP-CONTAINER. Like LINKAGE, each primitive describes a single function and the class of objects which is commonly associated with that single function. Fig. 4.4 illustrates the relationships between machine primitives and the objects which instantiate them.

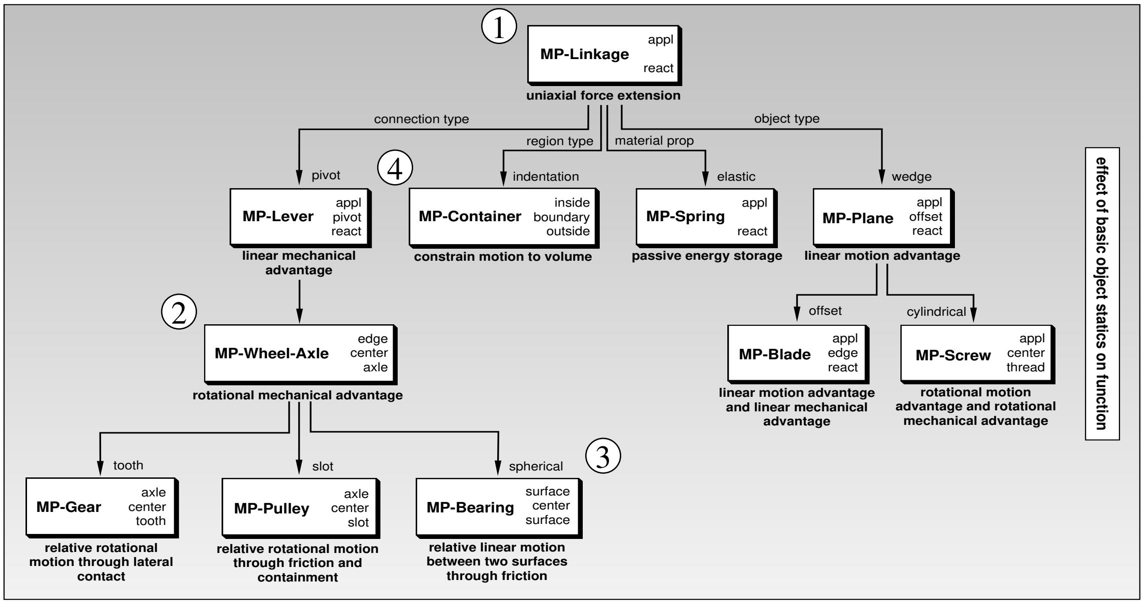


Figure 4.4 Machine primitive hierarchy, specialized by functional similarity and process role instantiation.

Fig. 4.4 is a hierarchy organized by the static differences between object primitives and the function classes they enable. Each box depicts the machine primitive and its role specializations as they instantiate the function appl, pivot, and react roles. For example, MP-LINKAGE (at 1) has only two roles, an appl and a react, which match the function appl and react roles. A stick can be used to extend and probe by transmitting force, and MP-LINKAGE can represent the stick and its function. Any object which can transmit force in at least one dimension and direction can instantiate MP-LINKAGE. The MP-LINKAGE *appl* role is instantiated by one stick end, and the MP-LINKAGE *react* role is instantiated by the other stick end.

MP-LEVER represents the function of magnifying or reducing applied force. The objects which instantiate MP-LEVER differ from those which instantiate MP-LINKAGE by the addition of a pivot, so they instantiate all three appl, pivot, and react function roles. When the force magnification is radial (2), the function/object is represented with MP-WHEEL-AXLE. If the wheel object's react region is toothed, then the function/object is represented with MP-GEAR, which describes motion control as well as force transmission. If the wheel object's react region is slotted, then the function/object is represented with MP-PULLEY, which describe transmission of rotational motion and force over distance. The primitive MP-BEARING (3) is used to represent relative motion between objects. The bearing object transmits force only on its outer surface, so the appl and react regions describe the same surface. The resulting bearing object must be cylindrical or spherical.

A linkage object with an indentation is capable of containment, which is represented with the primitive MP-CONTAINER (4). MP-SPRING is used to represent the function/objects capable of passive energy storage. Any object which is perturbed with force value large enough to enable the process STORE is represented with MP-SPRING.

Wedge shaped linkage objects are used to magnify or reduce the amount of motion in one dimension (and hence force) by transforming it into component motions in multiple dimensions. The wedge shape provides a built-in pivot in the form of the wedge offset angle. These function/object types are represented with MP-PLANE. When a person walks up a gangway, he or she is trading off the horizontal distance travelled in order to make the vertical distance travelled more enjoyable. Like MP-LEVER, the amount of work done remains the same. When the plane object is applied primarily along its edge region, instead of its react region, to enable DEFORM processes, then its function is represented with the primitive MP-BLADE. The plane object can be used linearly or rotationally. When used rotationally, the function/object is represented with MP-SCREW. Table 4.1: organizes the machine primitive classes shown in Fig. 4.4 by their physical characteristics.

Primitive Class	Parent Class	Physical Characteristics/ Parts	I/O Regions
MP-LINKAGE	Phys-Obj	Material: Elastic in dimension	Appl, React
MP-LEVER	MP-LINKAGE	Add Pivot Region	Appl, Pivot, React
MP-WHEEL-AXLE	MP-LEVER	Shape: Circular	Edge, Center, Axle
MP-BEARING	MP-WHEEL-AXLE	Shape: Spherical or Cylindrical	Surface, Center, Surface
MP-GEAR	MP-WHEEL-AXLE	Add Tooth Region; Parts: Gear or Chain	Tooth, Center, Axle
MP-PULLEY	MP-WHEEL-AXLE	Add Slot Region; Parts: Pulley and Belt	Slot, Center, Axle
MP-PLANE	MP-LINKAGE	Shape: Wedge (Offset Angle)	Appl, React
MP-BLADE	MP-PLANE	Attribute: Sharp	Appl, Edge, React

Table 4.1: Machine primitive (MP) classification.

Primitive Class	Parent Class	Physical Characteristics/ Parts	I/O Regions
MP-SCREW	MP-PLANE	Shape: Cylindrical	Appl, Center, Thread
MP-SPRING	MP-LINKAGE	Material: Elastic in dimr	Appl, React
MP-CONTAINER	MP-LINKAGE	Add Indentation Region	Inside, Boundary, Outside

Table 4.1: Machine primitive (MP) classification.

The objects which instantiate machine primitives and their function roles are presented in Table 4.1:. In the table, a parent describes the function class from which a primitive descends. MP-LINKAGE is dependent only on the material properties of an object. If the object is capable of transmitting force in at least the dimension and direction of perturbation, then it is a linkage object.

The physical characteristics and parts in column 3 of Table 4.1: describe the changes to the parent class objects which define the current object class. For example MP-PULLEY is instantiated by a pulley object which consists of a wheel object with a slot on its outer edge and a rope or belt. These physical characteristics are not invariant, but describe general classes which can instantiate the machine primitive. For example, MP-WHEEL-AXLE need not be instantiated by circular objects, as described in the table. The wheel object primitive is generally associated with circular objects, although a crank, which is not circular in general, transmits force radially and can be represented with MP-WHEEL-AXLE. The function enablement takes precedence over the physical appearance of the object. The gear object associated with MP-GEAR is instantiated by a combination of objects. MP-GEAR is enabled by two gears in tooth-contact. Likewise, the primitive MP-PULLEY is enabled for a pulley object which consists of a pulley and belt. Both MP-GEAR and MP-PULLEY describe the transmission of rotational force and motion from the region instantiating their appl role to that instantiating their react role, if the regions are the same. For example, a gear whose axle serves as a bearing, such as a pinion, transmits force from contact at one tooth region to that at another. The same primitives are used to represent force magnification from the region instantiating their appl role to that instantiating their react role if the regions are not the same. A drive shaft and gear are used to reduce force from the axle radius to that of the teeth.

4.2.1 MP-LINKAGE

MP-LINKAGE describes objects capable of, and used for, transmitting force to another object in the dimension and direction of an applied force. MP-LINKAGE is enabled by contact between the linkage object and another object, at the linkage object react region. MP-LINKAGE is initiated by an applied force at the linkage object appl region. An enabled MP-LINKAGE is terminated by a force on the second object at the region where it contacts the linkage object. MP-LINKAGE describes the input and output of the process TRANS-

FORM-TRANSMIT. These relationships are depicted in Fig. 4.5.

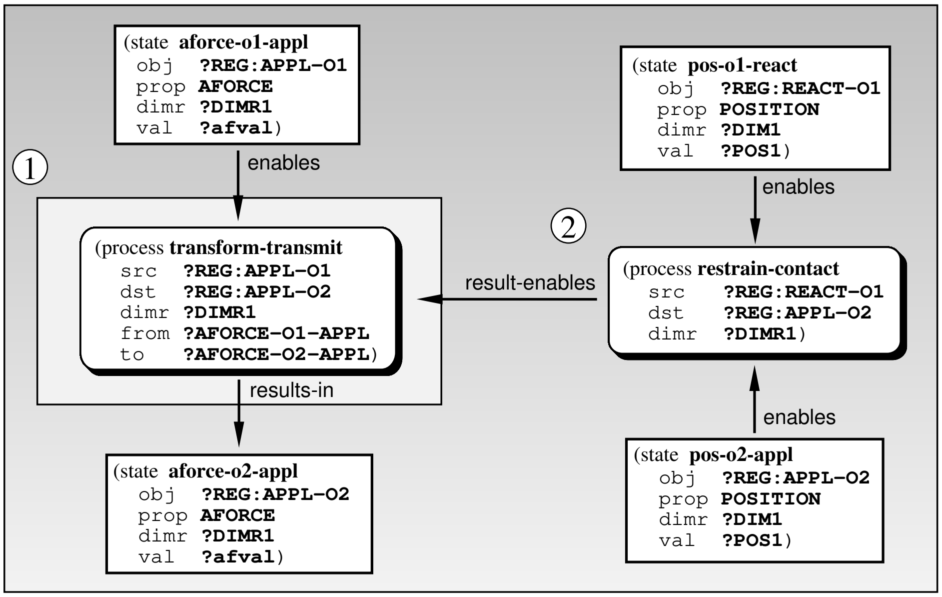
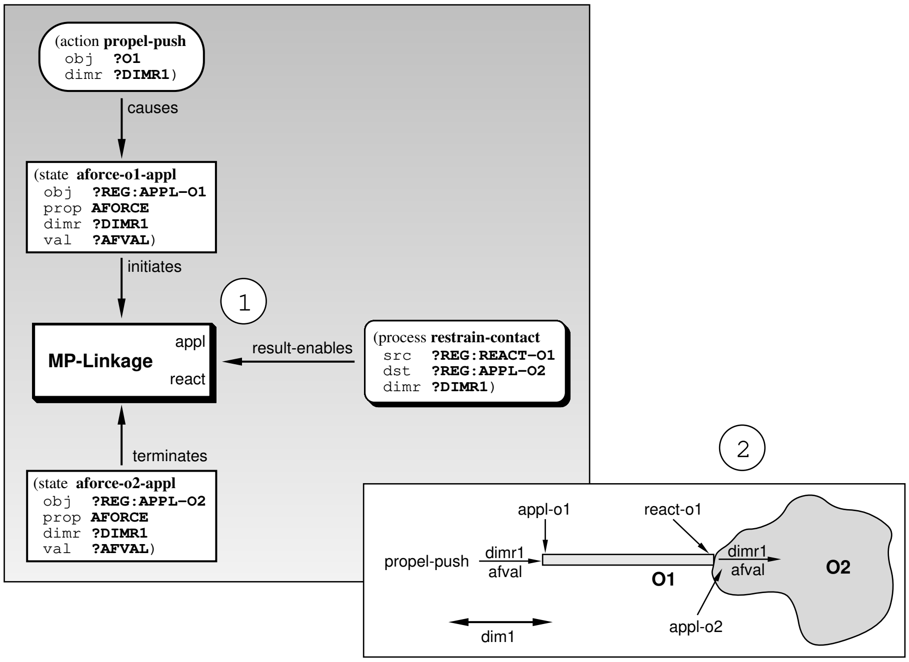


Figure 4.5 The behavioral sequence associated with the function of MP-LINKAGE.

Fig. 4.5 illustrates the behavioral sequence which represents the function associated with MP-LINKAGE. The shaded box (at 1) shows the function's primary process. The entire sequence is represented with two regions on the linkage object. These regions instantiate the appl and react roles of MP-LINKAGE. The appl role is instantiated by the region APPL-O1, and the react role is instantiated by the region REACT-O1. This representation is also shown in box form, instantiated for a stick linkage, in Fig. 4.6. The box labeled MP-LINKAGE (at 1 in Fig. 4.6a) is a shorthand form for the shaded box at (1) in Fig. 4.5. The box is shown with the states which initiate and terminate the function, and with the connectivity enablement between the stick and the amorphous blob depicted in Fig. 4.6a (at 2). The slot-filler notation associated with the box (at 1 in Fig. 4.6a) is presented in Fig. 4.6b .



(a)

```
(function LINKAGE-TYPE1
  appl     ?APPL-O1
  react    ?REACT-O1
  from     ?AFORCE-O1-APPL
  to       ?AFORCE-O2-APPL)
```

(b)

Figure 4.6 Block version of LINKAGE function, linkage object statics, and the associated slot-filler representation.

The graphic illustrates a stick linkage object (O1) being used to probe a general object (O2). The box labeled MP-LINKAGE (at 1) organizes the enablements, initiating, and terminating conditions associated with the sequence in Fig. 4.5. The process RESTRAIN-CONTACT between the stick (O1) and O2 enables the function. The function is shown being initiated by the force AFORCE-O1-APPL. The force is produced by pushing (PROPEL-PUSH) the stick. This causes a force at APPL-O1, which is transmitted internally to all contact locations. The state AFORCE-REACT-O1 is subsequently implied. The resulting state AFORCE-APPL-O2, which represents the force transmitted to O2, terminates MP-LINKAGE.

Machine primitives can be combined together as long as there are no discrepancies between their enabling conditions. The box used to depict the MP-LINKAGE in Fig. 4.6 can be combined to describe more complex devices. For example, consider the four components of a nutcracker, as depicted in Fig. 4.7.

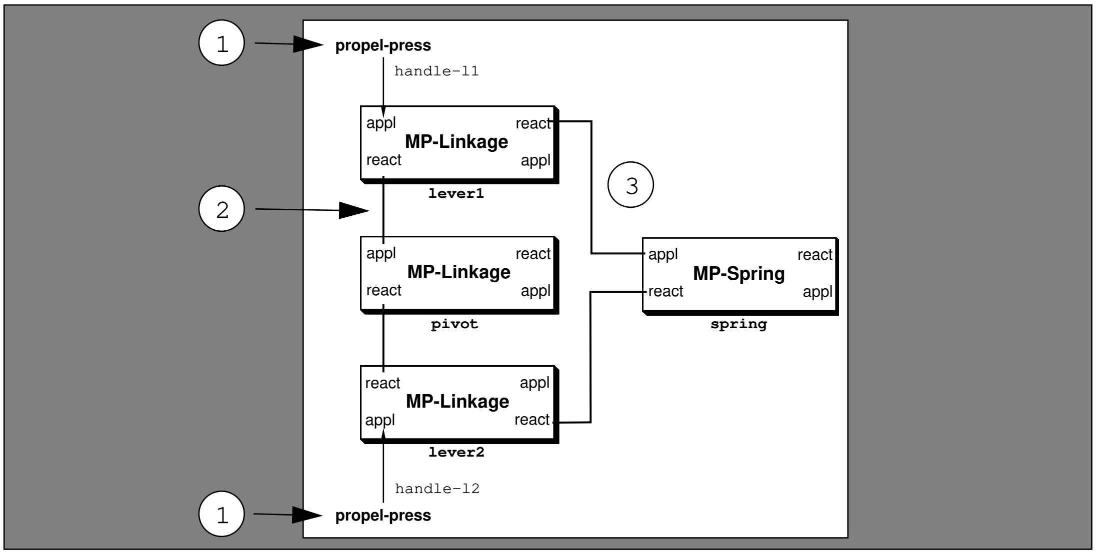


Figure 4.7 A partial MP diagram for a nutcracker without a nut.

Fig. 4.7 diagrams a nutcracker function. The blocks in the figure represent the functions which are enabled by the components of the nutcracker in the context of the action (at 1). In this figure, the components of the nutcracker are labeled below the block. Each component of the nutcracker instantiates the roles of a machine primitive. The figure shows only the blocks associated with the primitives. The actions applied during nutcracker operation are illustrated as thin lines with arrows. The solid links between the blocks (e.g., at 2) denote the static connection relations (inter-object dynamics) between the objects which instantiate the primitives. In the nutcracker, these relations are instantiated RESTRAIN-PIVOT (2) and RESTRAIN-CONTACT (3) processes. The states which initiate and terminate each function are assumed to be effected through these links. That is, the links represent where motion and force flow through the device. Each block has the primitive roles duplicated for ease in producing the drawing. Notice that the react role of the topmost MP-LINKAGE has two links emanating from it. When a link is associated with each region of the same primitive it simply means that the region engages in both interactions. In this case, the force which is reacted at the connection between the LEVER1 and the PIVOT is also reacted at the connection between the LEVER1 and the SPRING.

MP-LINKAGE is used to represent the nutcracker levers because, alone, they can only be used to transmit force. When a force is applied to the components of the nutcracker, as with the actions depicted at (1) in Fig. 4.7, it propagates along the links, initiating the respective machine primitive. In this example, when HANDLE-L1 and HANDLE-L2 are pressed, an applied force is generated in both LEVER1 and LEVER2. The linkage objects are connected at their other ends, to PIVOT, so the force is reacted (TRANSFORM-TRANSLATE) at PIVOT. When the local forces on PIVOT are considered together, the transmitted forces cancel. However, the local force is unbalanced at the handle regions, so the linkages are free to move (MOTION-ROTATE) about the pivot. The relative motion enables SPRING (STORE-COMPRESS) on SPRING, compressing the spring. Interobject motion has not been depicted in the figure.

Fig. 4.7 is called an *MP diagram* and is used to show how machine primitives interact to describe the function of arbitrary devices. Fig. 4.7 shows a single MP diagram for the nutcracker device. Whereas a device has a single static representation, it has a unique MP diagram for each function it can instantiate.

There are three MP-LINKAGE class specializations in FONM: longitudinal, lateral, and torsional. Each is distinguished by the type of force applied.

LINKAGE-TYPE1 (*longitudinal*): The applied force is reacted, in the linkage object, in the same dimension shared by the appl and react linkage object regions. For example, the stick depicted in Fig. 4.5, has the force applied at APPL-O1 and reacted at REACT-O1. Both regions lie along the length of O1, so they share the dimension ALONG-LENGTH. As long as the force dimension is ALONG-LENGTH, the LINKAGE is longitudinal.

LINKAGE-TYPE2 (*lateral*): Describes the function of translating force. The perturbation force is reacted in a dimension not shared by the appl and react linkage-object regions.

LINKAGE-TYPE3 (*torsional*): The perturbation force is rotational rather than translational.

The nutcracker levers shown in Fig. 4.7 are represented as LINKAGE-LATERAL, since the force is applied in a dimension other than the one commonly shared by the lever regions.

4.2.2 MP-LEVER

MP-LEVER describes devices used to produce in-plane mechanical advantage by enabling the process TRANSFORM-MAGNIFY. Lever objects have regions which instantiate each of the appl, pivot, and react device function roles. MP-LEVER is enabled by two contacts between the lever object and other objects: one at the lever object region that instantiates the function pivot role, and the other at the lever object region that instantiates the function react role. MP-LEVER is initiated by an applied force at the lever object region which instantiates the function appl role. An enabled MP-LEVER is terminated by a force on the object in contact with lever object. MP-LEVER describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the force is magnified.

These relationships are depicted in Fig. 4.8.

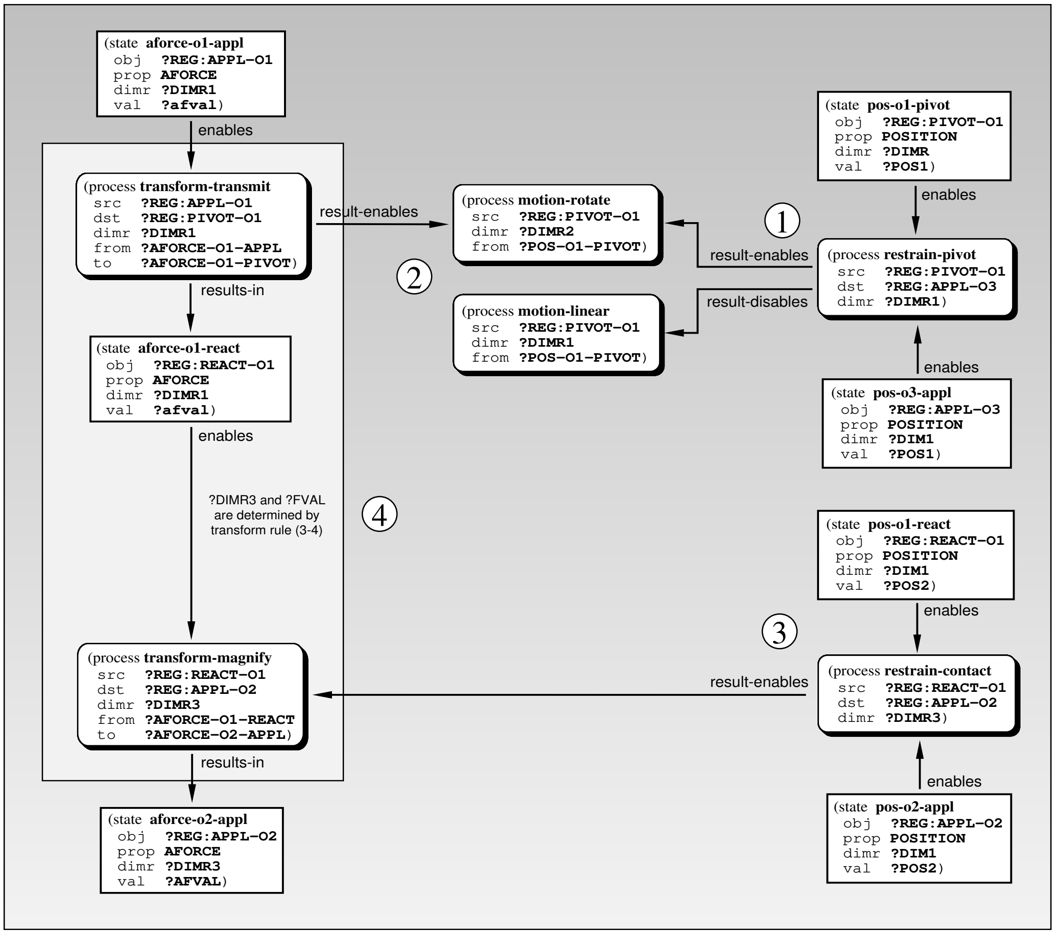
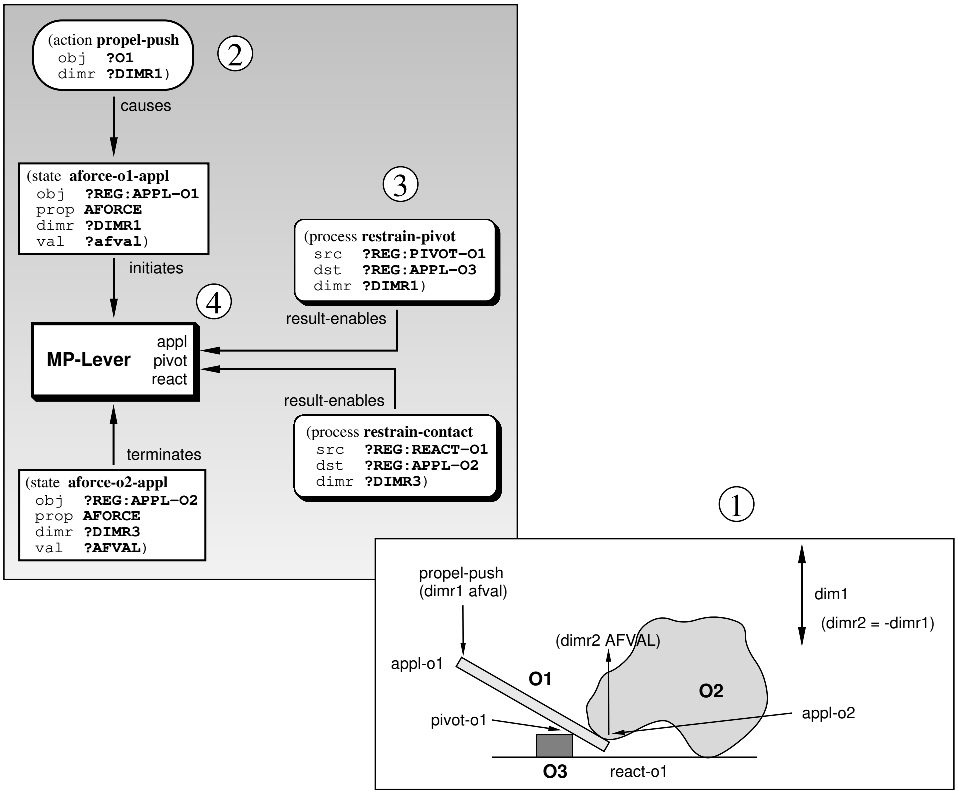


Figure 4.8 Process sequence associated with the LEVER machine primitive.

Fig. 4.8 depicts the sequence of behavioral processes which enable and describe MP-LEVER. The RESTRAIN-PIVOT process (1) enables MOTION-ROTATE and disables MOTION-LINEAR at the contact region of the lever object (2). The RESTRAIN-CONTACT process (3) provides the alternate force path which enables TRANSFORM-MAGNIFY, and the TRANSFORM rule TR-4 is used to determine the value of the magnified force. The shaded box on the left represents the intra-object dynamics of MP-LEVER. The representation for MP-LEVER in Fig. 4.8 is depicted in block form and slot-filler notation in Fig.

4.9.



(a)

```
(function LEVER-TYPE2
  appl     ?APPL-O1
  pivot    ?PIVOT-O1
  react    ?REACT-O1
  from     ?AFORCE-O1-APPL
  to       ?AFORCE-O2-APPL)
```

(b)

Figure 4.9 Block version of MP-LEVER function, lever object statics, and slot-filler representation

The graphic in Fig. 4.9a (at 1) illustrates a two-by-four (O1) and brick (O3) being used to lift a potato-shaped object (O2). O1 is propped up by an object O3, allowing O1 to rotate about the location PIVOT-O1 but disabling downward translational motion. This is a fulcrum pivot. The shaded portion of Fig. 4.9a illustrates the instantiation of MP-LEVER (shown as a block at 4) for the objects depicted in the figure. The action PROPEL-PUSH (2) causes the force AFORCE-O1-APPL which initiates MP-LEVER. The pivot (3) and contact processes defined by O1-O3 statics enable MP-LEVER, and the force AFORCE-O2-APPL terminates MP-LEVER.

Dimensions and Pivots

MP-LEVER causes *in-plane* force or motion magnification as a trade-off with the distance the lever object moves. MP-LEVER is enabled by a restraint (caused by RESTRAIN-PIVOT) which disables linear motion in a translational dimension and enables rotational motion in a rotational dimension. Determination of the dimensions of motion disablement/enablement is thus an important consideration in representing MP-LEVER. Determining the translational dimension is straightforward: the pivot must restrict lever object motion in the dimension and direction of the applied force. In Fig. 4.9a, the applied force is in DIMR1 so the pivot restricts O1 linear motion in DIMR1. In MP-LEVER, the applied force is a lateral force, meaning that its dimension is not the dimension shared by the appl and pivot regions. Dimension *sharing* means that the two regions have the same locations values in one dimension but different location values on another. They share the second dimension. For example, if the forced dimension is the O1 ALONG-DEPTH dimension, then the dimension shared by the appl and pivot regions is ALONG-LENGTH. These two dimensions determine the plane of motion for the lever object. The rotational dimension is about the third (or out-of-plane) translational dimension. In the previous example, the rotational dimension is ABOUT-WIDTH. If the lever object appl and react regions share the same translational dimension as the appl and pivot regions, then the magnification is in-plane.

Instantiated Machine Primitives

The context of a device function can change without changing how the force is applied. In such cases the device is represented with a new function, but the function is a specialization of the former function. For example, when the nutcracker handles were pressed together in the example illustrated in Fig. 4.7, the device was represented as a pivoted combination of MP-LINKAGES. When a nut is placed between the levers, however, the second contact location enables TRANSFORM-MAGNIFY and the device can be represented as a combination of MP-LEVERs. The pivot region is defined by the connection between the nutcracker levers (LEVER1 and LEVER2) and the pivot (PIVOT). The pivot region of each nutcracker lever is located at the linkage end regions instead of somewhere along them. However, this only affects the direction of magnified force and not the concept of "leverness." Fig. 4.10 illustrates how the MP-Diagram of Fig. 4.7 is modified by the addition of the new contact between the nutcracker jaws and the nut.

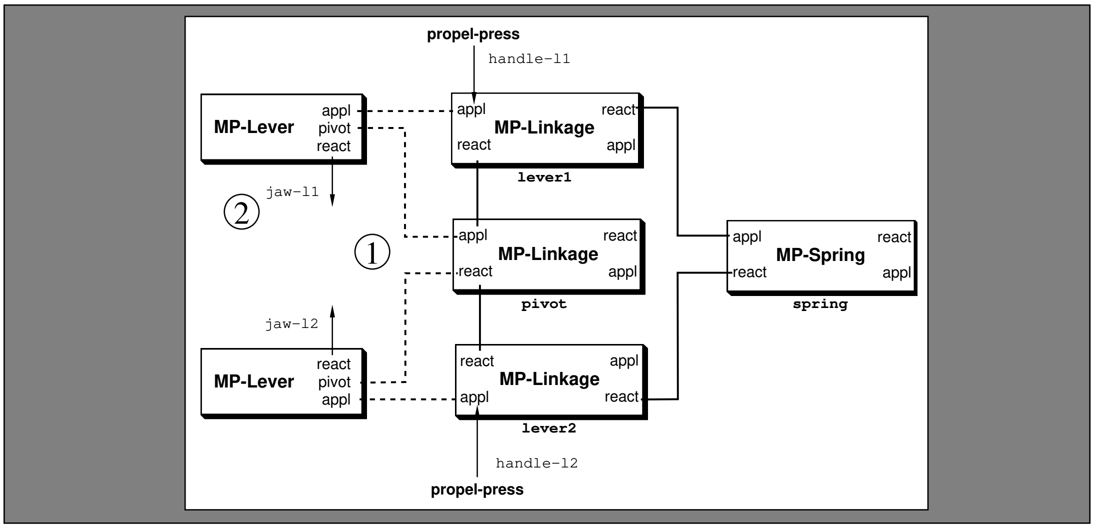


Figure 4.10 The full nutcracker MP diagram.

The LEVER1 region which instantiated the MP-LINKAGE appl role now instantiates the

MP-LEVER appl role. The linkage object react region is still connected to the PIVOT, which still instantiates an MP-LINKAGE, but the PIVOT region which instantiated the MP-LINKAGE appl role now instantiated the MP-LEVER pivot role (1). Finally, the JAW-L1 region (2) now instantiates the MP-LEVER react role. The nutcracker SPRING continues to instantiate MP-SPRING. The dotted lines in Fig. 4.10 (1) mean that the regions are the same rather than that they are connected. Thus the instantiation of MP-LEVER requires no different objects, just the addition of a contact location.

Lever Classes

The introduction of a second type of lever object, one in which the react region is located between the appl region and the pivot region, without adversely affecting the notion of ’leverness,’ suggests an MP-LEVER class hierarchy differentiated by how and where force is applied, pivoted and reacted. There are three specialization classes of MP-LEVER, each defined by the relative location of the lever object's appl, react, and pivot regions: (a) teeter-totter-like, (b) wheel-barrow-like, and (c) crane-like. These lever types are commonly referred to as Type 1, Type 2, and Type 3 levers, respectively, in the literature (e.g., [Macaulay 1988]), as illustrated in Fig. 4.11.

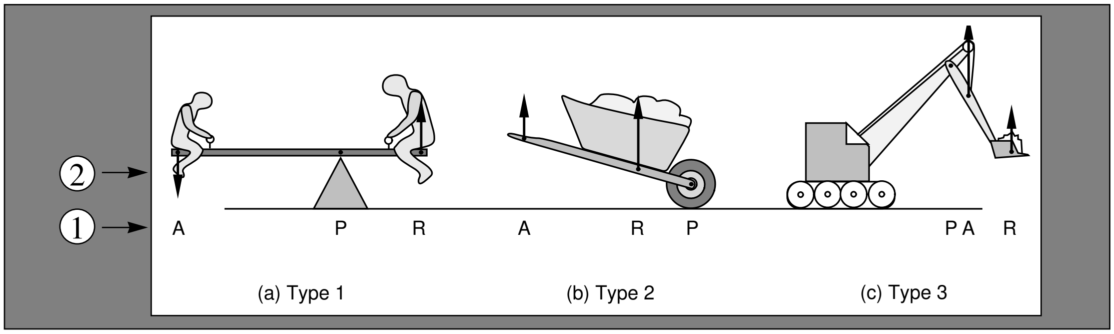


Figure 4.11 Three MP-LEVER classes: (a) a Type 1 lever, (b) a Type 2 lever, and (c) a Type 3 lever.

LEVER-TYPE1 (*Teeter-Totter*): Applied and reacted forces are located on opposite sides of the pivot region. Forcing regions move in opposite directions. Force is magnified.

LEVER-TYPE2 (*Wheel-Barrow*): Reacted force is located between the applied force and the pivot. Both forcing regions move in the same direction. Force is magnified.

LEVER-TYPE3 (*Crane*): Applied force is located between the reacted force and the pivot. Both forcing regions move in the same direction. Motion is magnified.

Fig. 4.11 depicts three different kinds of levers. The first two (a and b) both magnify force but differ in the direction of motion at the lever object react region. The A, P, and R labels beneath each device in Fig. 4.11 illustrates the relative positions of the appl, pivot, and react regions in each lever class, respectively (at 1). In (c), the use of the machine dramatically changes, since (c) has a force magnification less than unity. However, as in a human arm, the advantage gained with this design favors motion rather than force. Note that the only component of the crane used in this example is the last section, which is pivoted at the end and lifted from the center. The arrows shown in Fig. 4.11 (at 2) represent the applied and reacted forces at the appl and react region locations for their respective devices. The arrow length provides a measure of relative force magnitudes. The resultant force (dimension and value) is related to the applied force by comparing the relative appl-to-pivot and react-to-pivot lengths. In FONM, the result can be calculated explicitly if the lengths are known, using transform rule 4 (TR-4), which was presented as relation (3-4). Otherwise, the result takes one of three values: less than nominal, nominal, or greater than nominal (<NOM, NOM, >NOM), where nominal is the applied force value. For example, if AP is the length from the appl region to the pivot region, and PR is the length from the react region to the pivot region, then the ratio AP:RP in Fig. 4.11a is >NOM. The reacted force value is greater than the applied force value. The direction is reversed. The same is true of both Type 1 and Type 2 levers, and the opposite is true of Type 3 levers. These relationships are described by three rules for levers in Table 4.2:

Lever Rule	Lever Class	Resultant Force Relation/Value	Resultant Force Direction	Resultant Distance Value
LR-1	Type 1 (a p r)	TR-4	-DIMR1	TR-4
LR-2	Type 2 (a r p)	TR-4, >NOM	DIMR1	<NOM
LR-3	Type 3 (r a p)	TR-4, <NOM	DIMR1	>NOM

Table 4.2 Rules for determining force and motion in levers (and wheels).

The resultant force and distance entries for LR-1 in Table 4.2 can only be determined by knowing the location of the appl and react regions (relative or absolute), since the values can be <NOM, NOM, or >NOM depending on the locations. In LR-2 and LR-3 the relative locations are fixed so the resultant values are known before apriori.

4.2.3 MP-WHEEL-AXLE

MP-WHEEL-AXLE describes rotational force magnification. MP-WHEEL-AXLE is identical to MP-LEVER except that the applied force dimension is rotational. The wheel object has an edge region which instantiates the MP-LEVER appl role, a center region which instantiates the MP-LEVER pivot role, and an axle region which instantiates the MP-LEVER react role. MP-WHEEL-AXLE is enabled by a RESTRAIN-PIVOT for the center region, and a RESTRAIN-CONTACT between the wheel object axle region and the affected object. The restraint state resulting from the pivot enables rotational motion about the wheel object's center. The wheel object is often (but not invariably) circular, but the wheel object motion is always circular. For example, the crank on an egg beater is a non-circular wheel object, but is turned about its axle center and transforms force from the handle to the axle. MP-WHEEL-AXLE is initiated by a force at the wheel object edge region. An enabled MP-WHEEL-AXLE is terminated by a force on the object in contact with the wheel object axle region. MP-WHEEL-AXLE describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the perturbation force is magnified. The behavioral sequences which describe the function of MP-WHEEL-AXLE are depicted in Fig.

4.12.

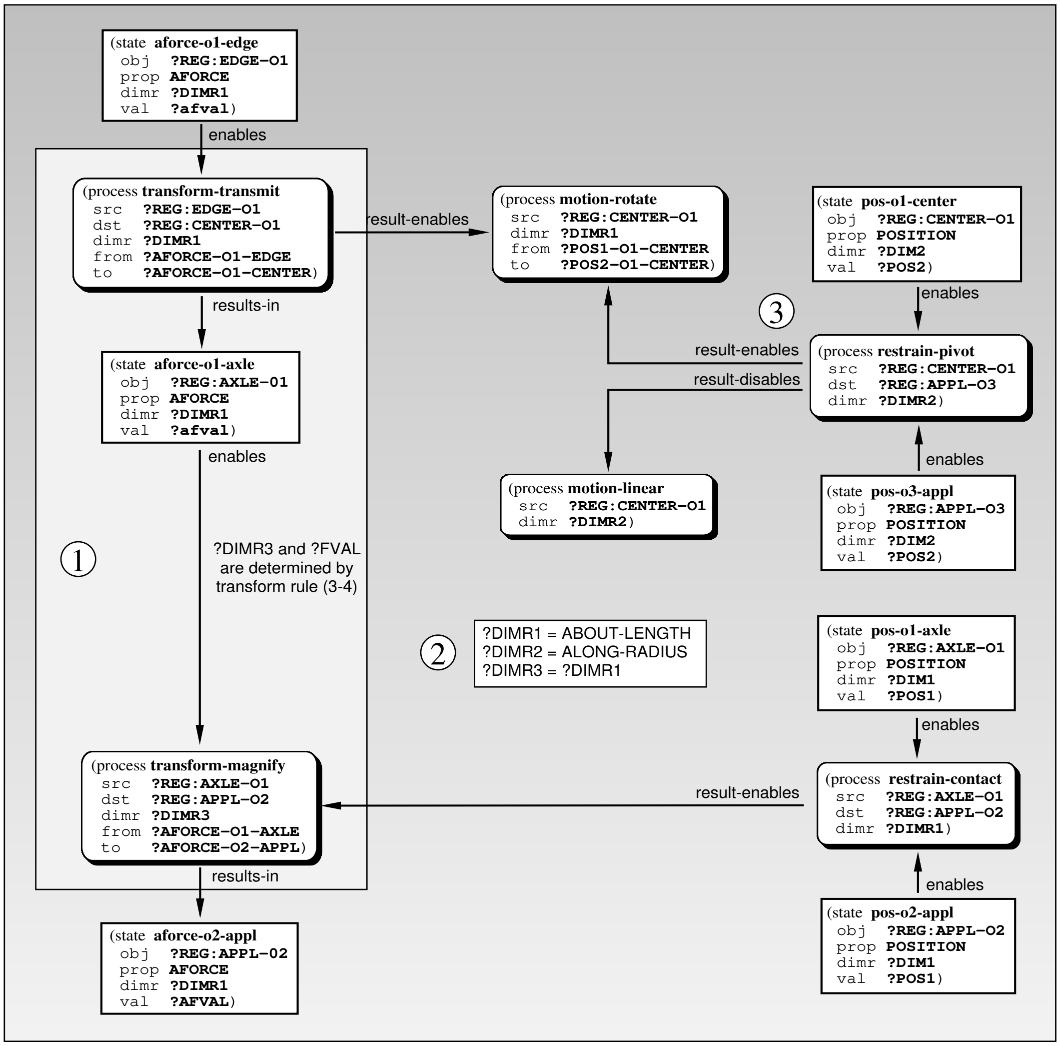
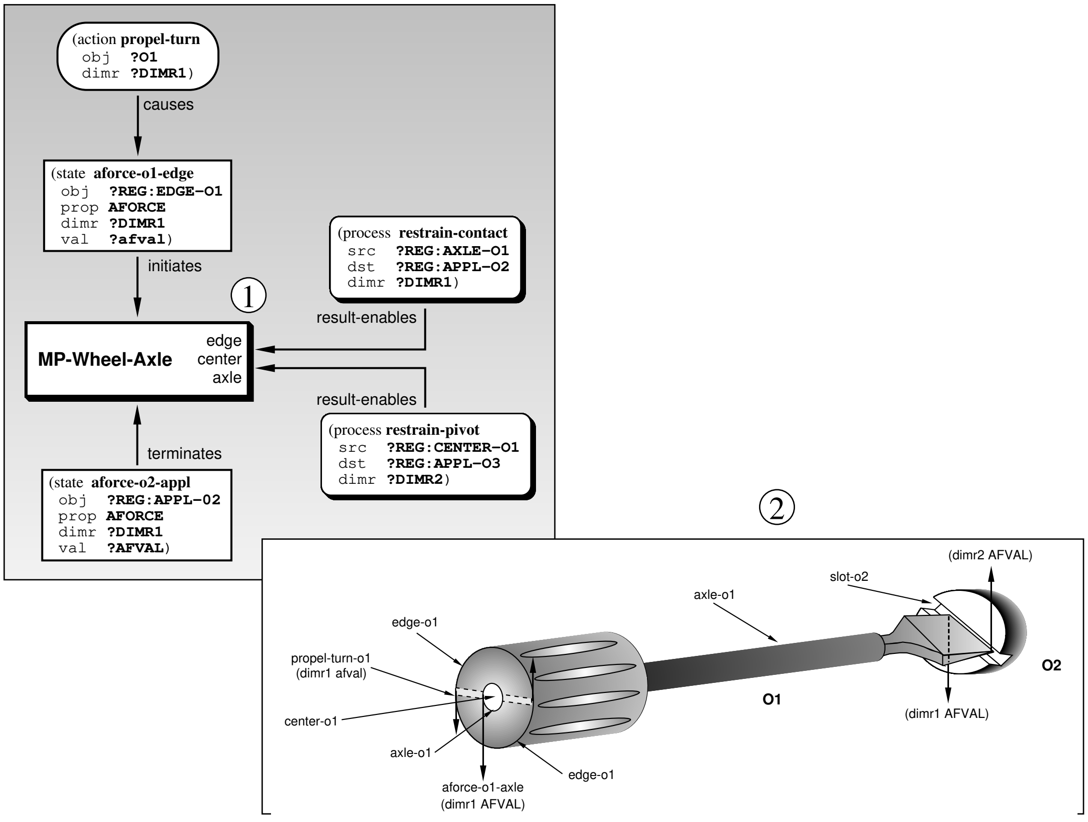


Figure 4.12 Behavioral process sequence associated with the WHEEl-AXLE machine primitive.

In Fig. 4.12, the shaded box (1) represents the intra-object dynamics of the wheel object. The figure is very similar to Fig. 4.5 (MP-LEVER), the differences between the two primitives are described in the inset box at (2). The wheel function dimensions are rotational and radial rather than translational. The representation for MP-WHEEL-AXLE in Fig. 4.12 is depicted in block form at (1) in Fig. 4.13a, and in slot-filler notation in Fig. 4.13b. Fig.

4.13a (2) illustrates a screwdriver (O1) driving a screw (O2).



(a)

```
(function WHEEL-AXLE-TYPE2
  appl     ?EDGE-O1
  pivot    ?CENTER-O1
  react    ?AXLE-O1
  from     ?AFORCE-O1-EDGE
  to       ?AFORCE-O2-APPL)
```

③

(b)

Figure 4.13 Block version of MP-WHEEL-AXLE function, wheel object statics, and associated slot-filler representation for a the Type-2 wheel illustrated

The screwdriver illustrated in Fig. 4.13 is considered a generalized wheel object rather than three individual components. The O1 center region (CENTER-O1) is supported by the user (O3), but not illustrated in the figure. When a force is applied at the screwdriver handle (i.e., outer radius), as a torque or moment, it is magnified at the axle radius. This force is translated along the axle length to the tip, where it is again magnified to the reaction location. The applied force dimension and direction at handle and tip are the same.

The upper portion of Fig. 4.13a illustrates the static preconditions (RESTRAIN-CONTACT and RESTRAIN-SUPPORT) which enable MP-WHEEL-AXLE. The process RESTRAIN-CONTACT between the wheel object (O1) and the screw object (O2) enables the process TRANSFORM-TRANSMIT to the screw. The process RESTRAIN-SUPPORT between O1 and O3 enables the process TRANSFORM-TRANSMIT to the supporting object O3. Note that in Fig. 4.12 (3) the connection precondition is a RESTRAIN-PIVOT class, whereas in this example (1) it is a RESTRAIN-SUPPORT class. These are not different. RESTRAIN-PIVOT is a subclass of RESTRAIN-SUPPORT specialized to enable rotation. When the user is holding the screwdriver rotation is still enabled.

MP-WHEEL-AXLE is shown being initiated by the force AFORCE-O1-EDGE, produced in this instance by turning (PROPEL-TURN) the screwdriver handle (region EDGE-O1). The state AFORCE-O2-APPL, which represents the force transmitted to, and magnified at, APPL-O2, terminates MP-WHEEL-AXLE.

Fig. 4.13b shows the mapping between the roles of machine primitives and the function class. The function appl, pivot, and react roles are shown instantiated by the wheel object edge, center, and axle roles.

The screwdriver in Fig. 4.13 magnifies force from an outer radius to an inner radius, but the traditional notion of a wheel is one of rolling and motion transmission on it edges. Both are wheel objects, but the location of the react region on the traditional wheel is the same as the appl region so there is no force magnification. Fig. 4.14 illustrates rotational translation and magnification in wheels perturbed on their edges.

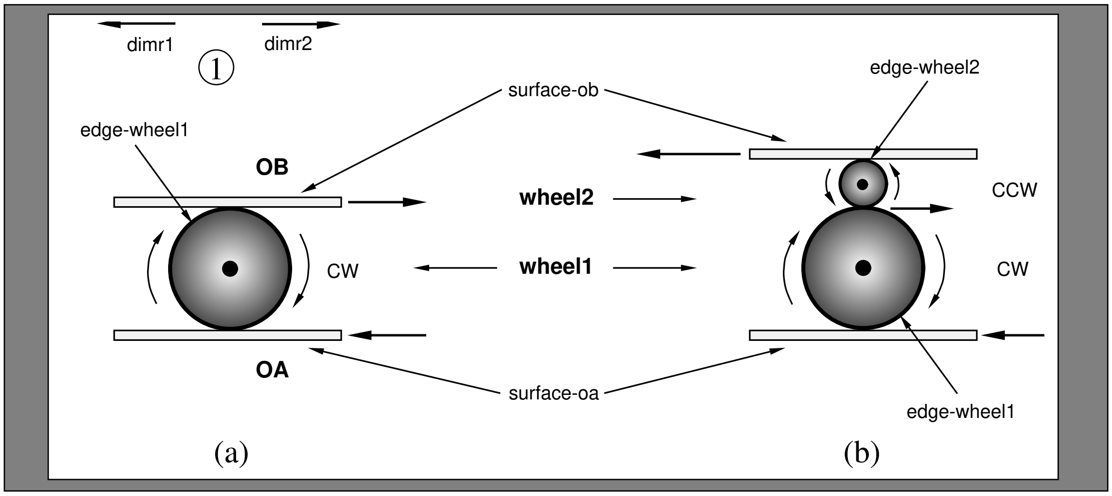


Figure 4.14 TRANSFORM processes in wheels.

Fig. 4.14a depicts a wheel (WHEEL1) in surface contact with two objects: at SURFACE-OA, and SURFACE-OB, respectively. SURFACE-OA moves in dimension/direction DIMR1 (1). The radial distance is the same from the center region of WHEEL1 (not illustrated) to any point on the edge. Applying lever rule LR-1, the resultant force value is the same (NOM) as the applied force, and the resultant force dimension is the opposite of the applied force (DIMR2 = -DIMR1), as noted in Table 4.2, Page 162:

Wheel Rule	Wheel Class	Resultant Force Value	Resultant Force Direction	Resultant Distance Value
WR-1	Type 1 (a p r)	LR-1, NOM	-DIMR1	LR-1, NOM
WR-2	Type 2 (a r p)	LR-2, >NOM	DIMR1	<NOM

Table 4.3 Rules for determining force and motion in single wheels.

Wheel Rule	Wheel Class	Resultant Force Value	Resultant Force Direction	Resultant Distance Value
WR-3	Type 3 (r a p)	LR-3, <NOM	DIMR1	>NOM

Table 4.3 Rules for determining force and motion in single wheels.

The result is force translation to SURFACE-OB by WHEEL1. WHEEL1 instantiates a wheel object, and MP-WHEEL-AXLE enables motion at SURFACE-OB in dimension/direction DIMR2. Rules WR-2 and WR-3 in Table 4.2 are repeated versions of LR-2 and LR-2 in Table 4.2.

The wheel in Fig. 4.14a translates force and motion. The wheels in Fig. 4.14b magnify force and motion. The motion applied to SURFACE-OA is transmitted directly to WHEEL1 in both examples. On the left, WHEEL1 is in direct contact with SURFACE-OB, so there is no mechanical advantage, but the direction of motion changes. On the right, WHEEL1 is in contact with a smaller wheel (WHEEL2) at region EDGE-WHEEL1. WHEEL2 is also in contact with SURFACE-OB at EDGE-WHEEL2. The direction of motion at the EDGE-WHEEL1/EDGE-WHEEL2 contact is DIMR2, so (applying LR-1 again) the direction of motion at SURFACE-OB changes back to DIMR1. In order to maintain contact with WHEEL1, WHEEL2 rotates faster, since WHEEL2 will rotate more than once for every rotation of WHEEL1. Mechanical advantage in wheel combinations is defined by the relative ratio of wheel radii, as represented in WR-1:

WR-4: dist(wh2) = dist(wh1) * (r1/r2) (4-1)

where dist is a relative measure of the amount of distance moved with respect to the wheel’s own radius, wh2 is WHEEL2, wh1 is WHEEL1, r1 is the radius of WHEEL1, r2 is the radius of WHEEL2, and r1/r2 is the *radius ratio* of the two wheels. WR-4 says that if the radius of the second wheel is smaller than the radius of the first wheel, then the radius ratio is greater than unity, and the distance moved by a surface region on the second wheel will be greater (with respect to its radius) than the corresponding surface region on the first wheel. In qualitative values, the radius ratio of a larger wheel to a smaller wheel is >NOM, where NOM is the speed of the smaller wheel, so the motion at SURFACE-OB is >NOM. The direction of motion also changes with wheel combinations, since the applied force is on different ’sides’ of the pivot for the different wheels. Rule WR-5 represents the change in direction.

WR-5: dimr(wh2) = -dimr(wh1) (4-2)

where dimr is the dimension and direction of motion, wh2 is the second wheel, and wh1 is the first wheel. WR-4 and WR-5 have the same form as rules TR-4, the difference being that the wheel rules apply to wheel compositions rather than single wheels. Gear and pulley combinations also produce mechanical advantage and motion direction changes the same way.

Wheel Classes

The screwdriver and wheel MP-WHEEL-AXLE examples illustrate two types of wheel function. As with MP-LEVER, there are three corresponding MP-WHEEL-AXLE specializations, defined by the relative location of the wheel object regions which instantiate the MP-WHEEL-AXLE edge, axle, and center roles: (a) gear-like, (b) screwdriver-like, and (c) drive-shaft-like, as shown in Fig. 4.15. Type 1 MP-WHEEL-AXLE describes traditional wheel motion and force transmission along the wheel object edge, while Type 2 and Type 3 MP-WHEEL-AXLEs describe radial (outer-inner, inner-outer) force magnifications.

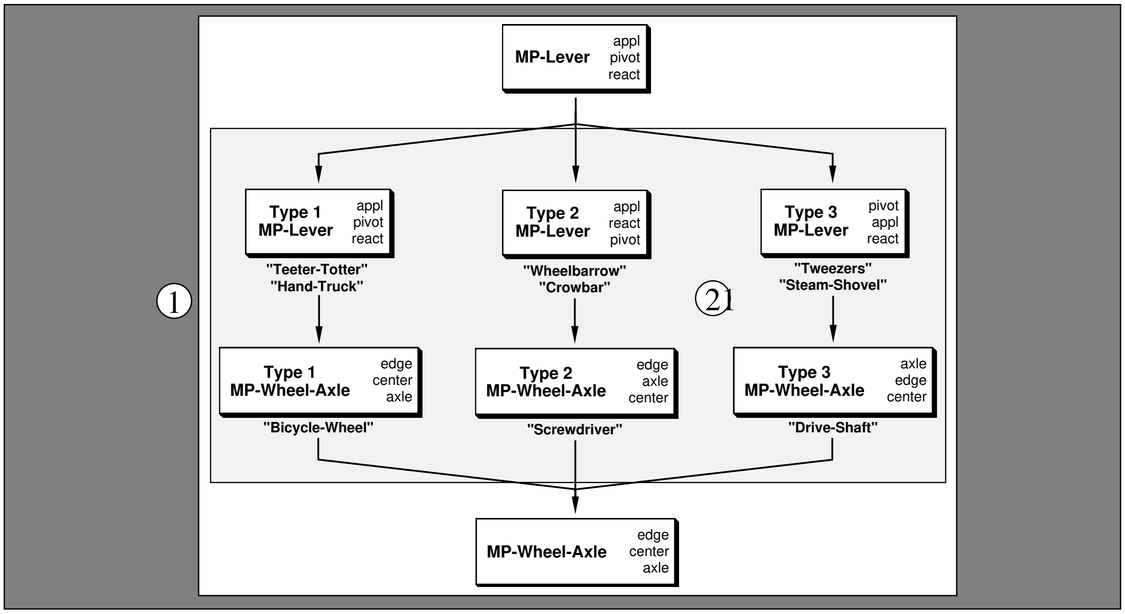


Figure 4.15 MP-LEVER and MP-WHEEL-AXLE classes.

Fig. 4.15 shows the correspondence between MP-LEVER and MP-WHEEL-AXLE classes. MP-WHEEL-AXLE also has three classes, referred to as Type 1, Type 2, and Type 3 wheels. Table 4.1: defined MP-WHEEL-AXLE as an MP-LEVER applied to a circularly shaped object. In this capacity, an MP-WHEEL-AXLE is specialized from a Type 1 MP-LEVER (1), as is the case for other *free-axle* devices: WHEEL, PULLEY, and GEAR combinations. MP-WHEEL-AXLE also describes *fixed-axle* leverage from outer-to-inner radius or vice-versa, corresponding to Type 2 and Type 3 MP-LEVERs (2). The wheel types depicted in Fig. 4.15 are illustrated below in Fig. 4.16.

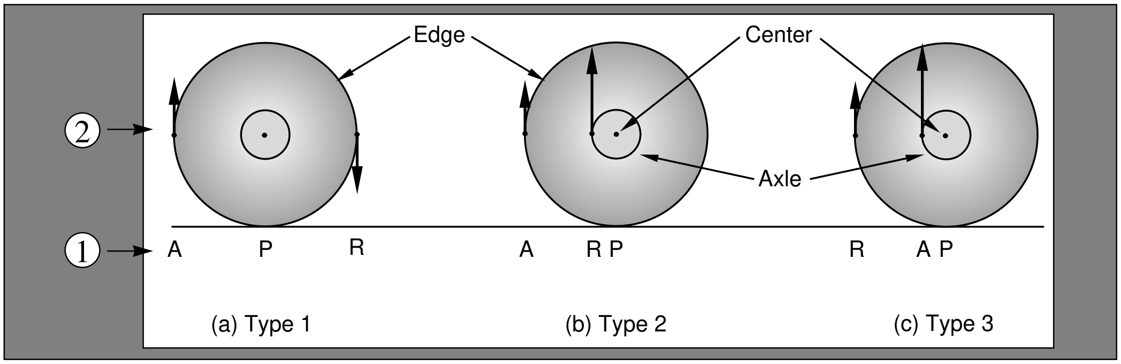


Figure 4.16 Three wheel device classes: (a) a Type 1 wheel, (b) a Type 2 wheel, and (c) a Type 3 wheel.

MP-WHEEL-AXLE-TYPE1 (*Gear*): Applied and reacted forces are both edge regions, on opposite sides of the center. Forcing regions move in opposite directions. Force is unchanged.

MP-WHEEL-AXLE-TYPE2 (*Screwdriver*): Force is applied at the edge region and reacted at the axle region, along the axle. Force dimension and direction is the same. Force value is magnified.

MP-WHEEL-AXLE-TYPE3 (*Drive Shaft*): Applied force is located at the axle region, and reacted at the edge region. Both forcing regions move in the same direction. Motion is magnified.

The A, P, and R beneath each device in Fig. 4.16 (1) represent the relative positions of wheel object edge, center, and axle regions with respect to the lever object function appl, pivot and react locations, respectively.

In Fig. 4.16, the arrows are shown at the edge and axle region locations for their respective objects (2). The arrow length provides a measure of relative force magnitude values. The value of resultant force is related to the applied force by comparing the relative edge-to-center and axle-to-center lengths, as with lever specializations. The resultant forces and distances are also identical to the lever classes identified in Fig. 4.2 except for Type 1 wheels. Type 1 wheels result in a NOM reacted force value because the fixed radii forces the edge-to-center and axle-to-center lengths to be equivalent.

4.2.4 MP-GEAR

MP-GEAR describes devices used to control the transformation of rotational force and motion by lateral interference between gear teeth. A gear object is a specialized wheel object with a toothed edge region, located at the wheel object surface1 and surface2 regions. The tooth region is a protuberance, so it can catch on objects as the gear rotates. MP-GEAR has tooth, center, and axle function roles. MP-GEAR is enabled by a RESTRAIN-PIVOT for the gear object region which instantiates the MP-GEAR center role, and a RESTRAIN-CONTACT between the gear object axle region and a second object. MP-GEAR is initiated by a force at the gear object axle region. An enabled MP-GEAR is terminated by a force state on the object in contact with the gear object tooth region. The teeth of both objects are the same size and shape. MP-GEAR describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the applied force is magnified. This relationship is depicted in Fig. 4.17, which illustrates the behavioral processes which describe MP-GEAR function. The shaded box represents the intra-object dynamics of the gear object. The static enablements for MP-GEAR are shown on the right of the box (a RESTRAIN-PIVOT at label 1 and a RESTRAIN-CONTACT at 2). The force (AFORCE-GEAR1-AXLE) which enables the TRANSFORM-MAGNIFY is applied at the gear object region which instantiates the MP-GEAR center role. The applied force and the pivoted center (1) enable rotation of the gear object. The contact with the second gear object (2) enables TRANSFORM-TRANSMIT to the tooth of the second gear (AFORCE-GEAR2-TOOTH1), which is the resulting state of MP-GEAR. The figure also illustrates the dynamics of the second gear object.

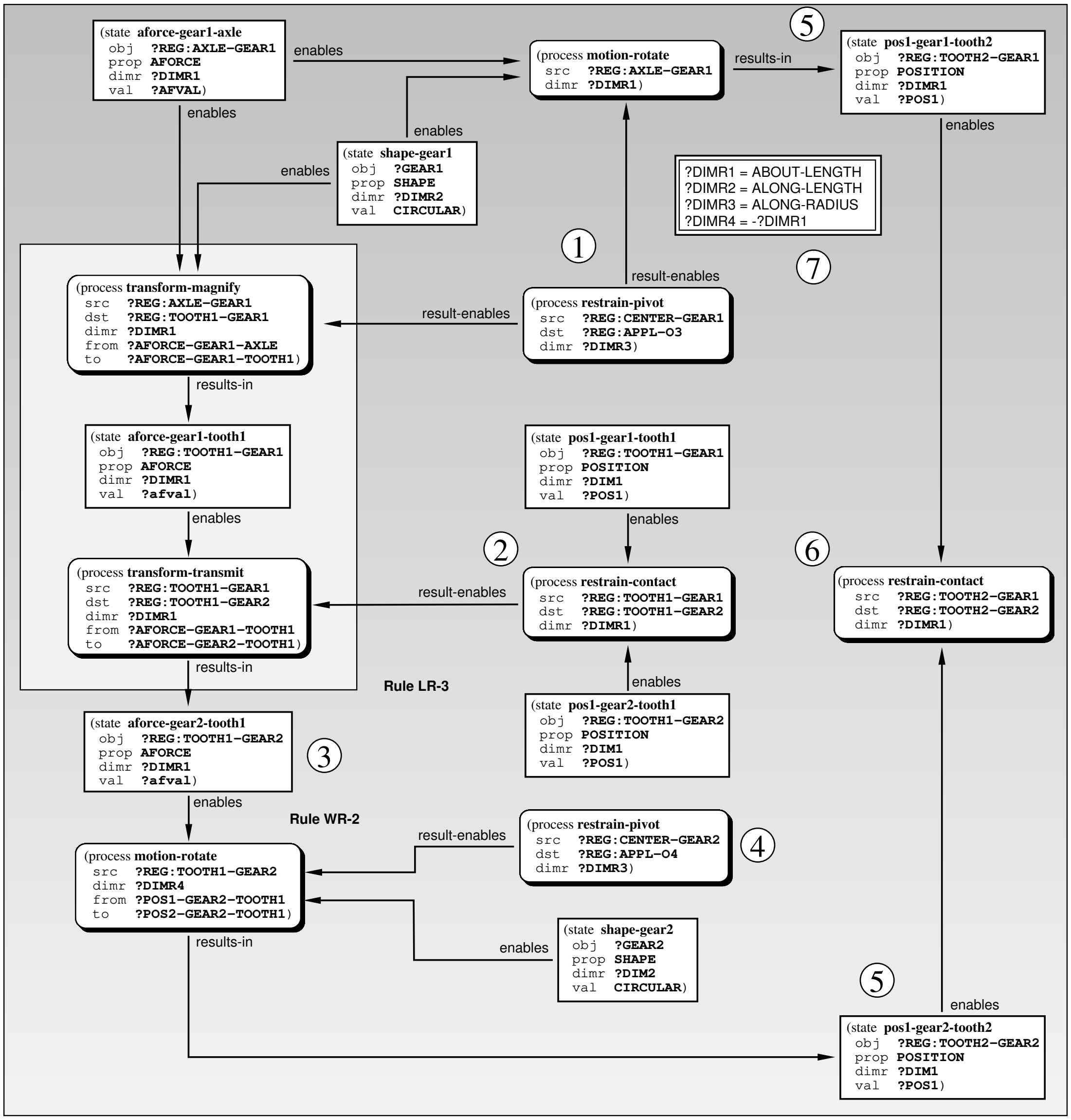
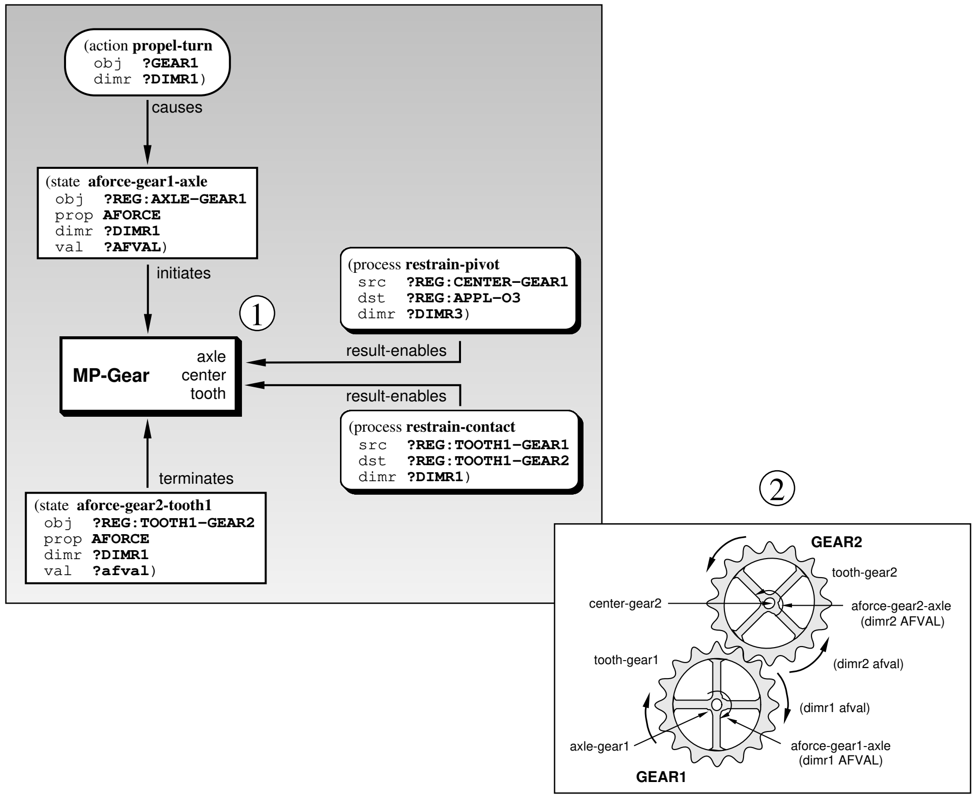


Figure 4.17 Behavioral process sequence associated with the GEAR machine primitive.

The force transmitted from the first gear (3), and the pivoted support of the second gear (4) enable its rotation, and the two gear objects advance by one tooth each (5). The advance places a second tooth of each gear in contact (6) at the same position which the first tooth of each gear were originally positioned. The process RESTRAIN-PIVOT between GEAR1 and O3, and between GEAR2 and O4, enables the process TRANSFORM-TRANSMIT to the supporting object (O3 and O4). These relationships have not been illustrated in the figure.

The representation for MP-GEAR in Fig. 4.17 is illustrated graphically at (2) in Fig. 4.18a. The machine primitive associated with the shaded box in Fig. 4.17 is also replaced with its block equivalent at (1) in Fig. 4.18a.



(a)

```
(function GEAR
  appl     ?AXLE-GEAR1
  pivot    ?CENTER-GEAR1         ◄──────── ③
  react    ?TOOTH1-GEAR1
  from     ?AFORCE-GEAR1-AXLE
  to       ?AFORCE-GEAR2-TOOTH1)
```

(b)

Figure 4.18 Block version of MP-GEAR function, gear object statics, and slot-filler representation.

The graphic illustrates two gears (GEAR1 and GEAR2) in motion. The size and shape requirement on tooth contact enables RESTRAIN-INTERFERE. Representing the motion of gear objects in tooth-contact as a one-tooth change in position has two consequences on the function of MP-GEAR. First, since there are a finite number of teeth on a gear, motion control is possible. Second, because the restraining process is lateral contact (using a protuberance), rather than surface contact, large force values can be transmitted because there will be no slipping.

Gear object statics can be specialized to control the direction and value of the force caused by MP-GEAR. For example, a gear object could be comprised of two gears with beveled edges. The result would be a change in the direction of the transmitted force/motion. A gear-object could be comprised of a gear and a chain, or a gear and a rack. The result would be the distance through which the force/motion are transmitted. The size of gears and the number of teeth on their edge regions affect the degree of control and the amount of force magnification, respectively. The amount of resultant distance magnification is controlled by the rule WR-1, which is based on radius ratio between the gear objects. The same result can be obtained with the ratio of teeth between the gear objects, as noted in rule GR-1:

$$\text{GR-1:} \qquad dist(g2) = dist(g1) * (t1/t2) \tag{4-3}$$

where dist is the relative distance travelled, with respect to the radius of the respective gear, g2 is the second gear, g1 is the first gear, t1 is the number of teeth on the first gear, and t2 is the number of teeth on the second gear.

4.2.5 MP-PULLEY

MP-PULLEY describes devices used to control the transformation of force and motion with edge-edge friction (surface-contact) and interference. The pulley object is a specialized wheel object with a slotted edge region. The slot is an indentation so it enables RESTRAIN-INTERFERE (it is not RESTRAIN-CONTAIN because the pulley belt center of gravity is outside that of the pulley). MP-PULLEY has slot, center, and axle function roles, which are associated with the MP-WHEEL-AXLE edge, center, and axle roles. MP-PULLEY is enabled by a RESTRAIN-PIVOT for the region which instantiates the center role, and a RESTRAIN-INTERFERE between the pulley object slot and belt inside-surface regions. MP-PULLEY is initiated by a forcing state at the pulley object axle region. An enabled MP-PULLEY is terminated by a force state on the object in contact with the pulley object slot region. MP-PULLEY describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the perturbation force is magnified. These relationships are depicted in Fig. 4.19.

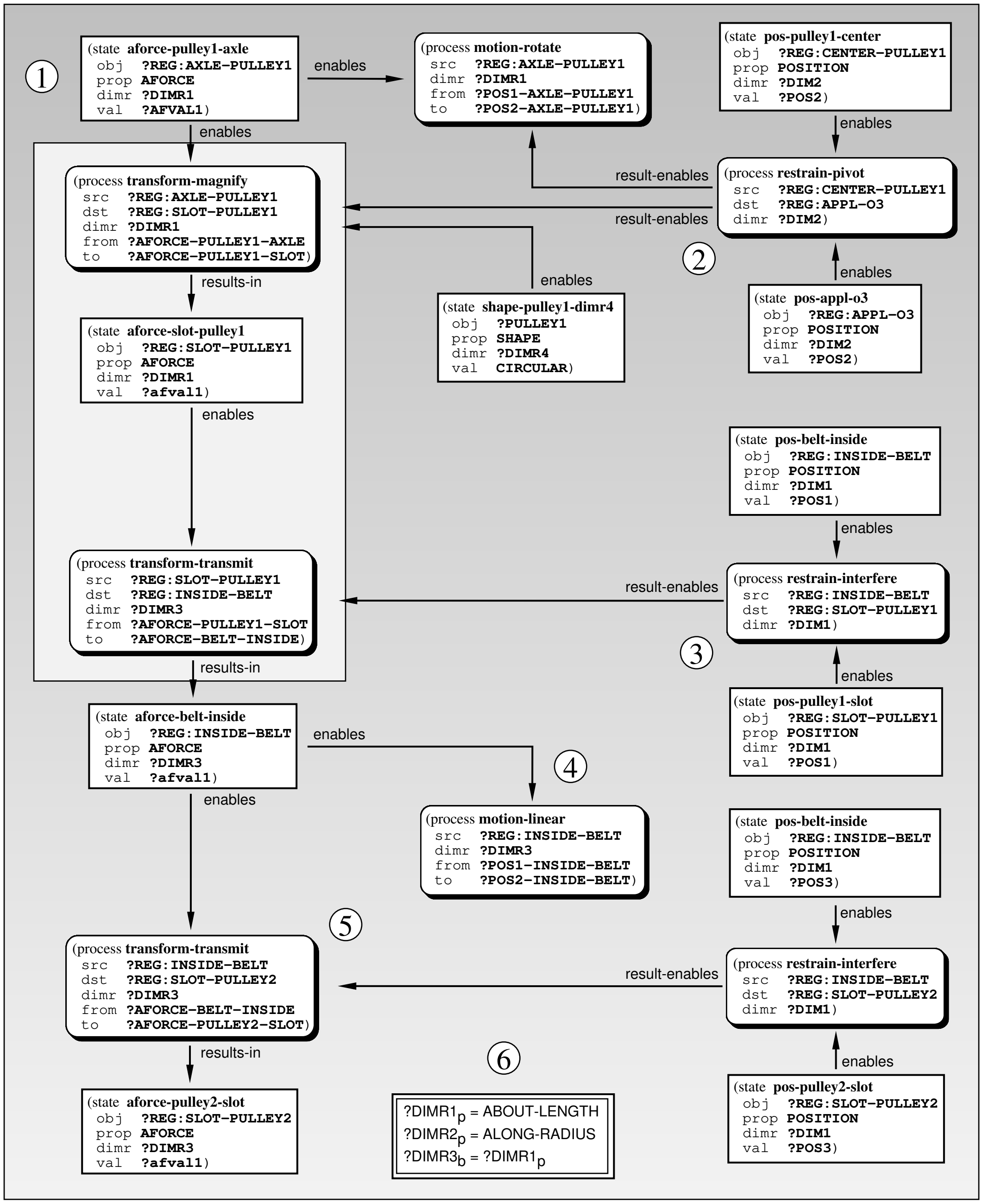
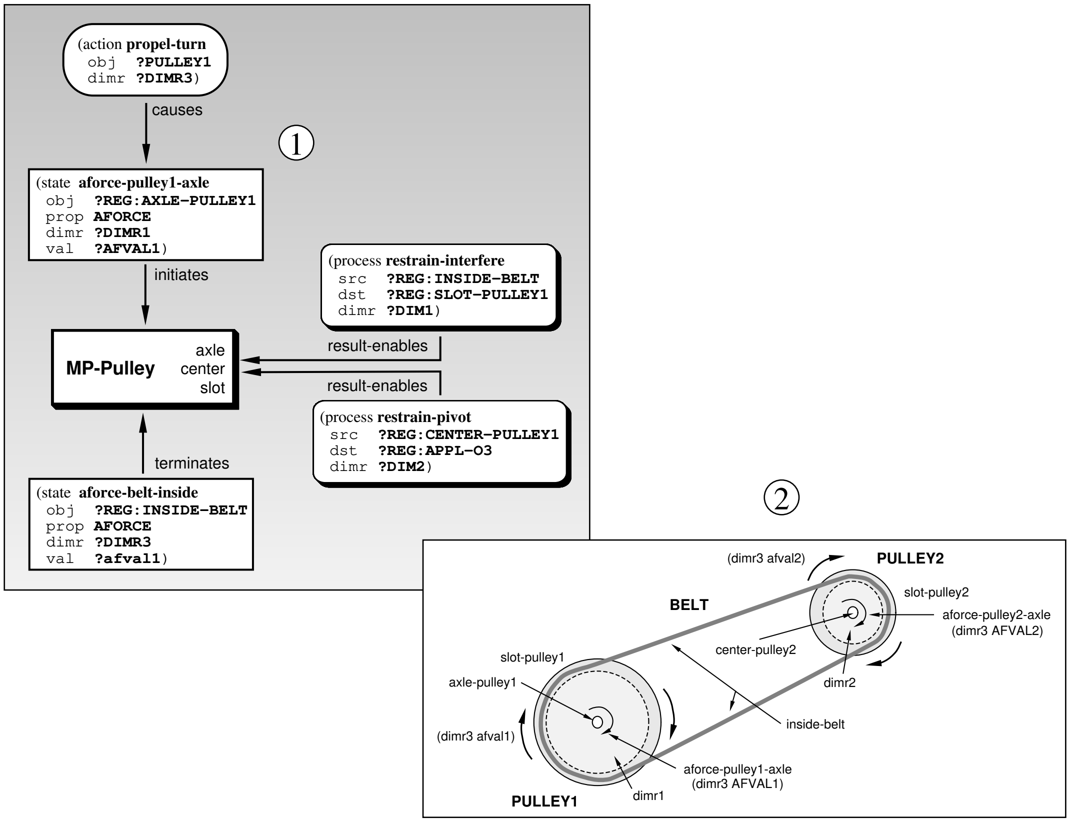


Figure 4.19 Behavioral process sequence associated with the PULLEY machine primitive.

Fig. 4.19 illustrates the behavioral processes which describe pulley function. The shaded box describes the intra-object dynamics of the pulley object. The applied force (AFORCE-PULLEY1-AXLE, at label 1) enables force magnification (TRANSFORM-MAGNIFY) from the pulley object axle to its slot contact. The RESTRAIN-PIVOT process (2) enables force magnification and rotation of the pulley object. The contact between the pulley object slot and the pulley belt (3) enables force transmission to the belt. The interference constrains the motion of the belt to the plane of the pulley object. Out-of-plane motion disablement is not illustrated in the figure. A second pulley object is also illustrated in the figure. The force resulting from MP-PULLEY (AFORCE-BELT-INSIDE) enables linear motion of the belt (4). If the belt is in contact with another gear (5), then the process is reversed. The legend labeled (6) is shows sample dimensions associated with the illustration of pulley object statics shown at (2) in Fig. 4.20a. The machine primitive associated with the shaded box in Fig. 4.19 is also replaced with its block equivalent at (1) in Fig. 4.20a.



(a)

```
(function PULLEY
  appl     ?AXLE-PULLEY1
  pivot    ?CENTER-PULLEY1         ←———— ③
  react    ?SLOT-PULLEY1
  from     ?AFORCE-PULLEY1-AXLE
  to       ?AFORCE-BELT-INSIDE)
```

(b)

Figure 4.20 Block version of MP-PULLEY function, pulley objects, and slot-filler notation for MP-PULLEY.

Fig. 4.20a illustrates two pulley objects (PULLEY1 and PULLEY2), and their common belt (BELT), in motion. The process RESTRAIN-PIVOT between PULLEY1 and O3, and between PULLEY2 and O4, enables a TRANSFORM-TRANSMIT to the supporting object (O3 and O4), but these relationships have not been illustrated. The MP-PULLEY block in Fig. 4.20a is shown being initiated by the force AFORCE-PULLEY1-AXLE, produced in this instance by turning (PROPEL-TURN) the pulley object at its axle (region AXLE-PULLEY1). The force (AFORCE-PULLEY2-SLOT), which represents the force transmitted to, and magnified at, SLOT-PULLEY2, terminates MP-PULLEY. The slot-filler notation for PULLEY function represented in Fig. 4.20b shows the mapping between MP-PULLEY roles and the function class roles.

Surface (or friction) contact between the pulley slot region (SLOT-PULLEY1) and the belt inner surface (INSIDE-BELT) has two consequences on the function of MP-PULLEY. First, since there is no way to guarantee relative position between a pulley and a belt, MP-PULLEY is not used as a motion control device. Second, because the restraining process is surface contact, transmitted forces are approximated by the amount of friction between the belt and the pulley slot.

Pulley object statics can be specialized to control the direction and value of the force caused by MP-PULLEY, similar to wheels and gears. For example, a pulley object could be comprised of a pulley and belt and, by twisting the belt, the dimension of motion can be arbitrarily changed. This cannot be done with MP-GEAR because a chain is usually fairly rigid so that forces can be transmitted. The force magnification produced by axle-to-slot transmission is determined with the rule WR-3 (or WR-2 in the reverse case). By cascading pulley objects, as depicted in Fig. 4.20a, mechanical advantage on the belt can be obtained without consideration of the axle forces whatsoever. For example, a block and tackle is a composition of many pulley objects sharing a single rope. Any multiple of input force magnification can be realized by increasing the number of pulley objects in the block. Force magnification in pulleys can be determined with radius ratios in the same way it was with wheels and gears, using WR-4. If the force of interest is the one in the belt, then the radii are not important but the number of pulleys is, and can be determined with the rule PR-1:

$$\text{PR-1:} \qquad \text{afval2} = \text{afval1} / N \qquad (4\text{-}4)$$

where afval1 is the applied load, afval2 is the resultant force value, and N is the number of pulley objects using the same belt in such a way that the belt travel is increased N-fold.

4.2.6 MP-BEARING

MP-BEARING describes devices that enable relative surface translation by reducing the friction associated with surface contact. The bearing object is a cylindrical or spherical object with a surface region that instantiates the wheel object edge and react regions. MP-BEARING has surface, center, and surface roles which play the role of the function appl, pivot, and react roles, respectively. MP-BEARING is enabled by a RESTRAIN-CONTACT processes between each surface and the surface regions of the bearing object. The bearing object shape and contact with the surface enables rolling. MP-BEARING is initiated by an applied force on a surface in contact with the bearing object surface1 region. An enabled MP-BEARING is terminated by a force on the second surface in contact with the bearing object surface2 region. MP-BEARING describes the enablements and resulting state of the process TRANSFORM-TRANSMIT, so the value of the applied force is unchanged. These relationships are depicted in Fig. 4.21.

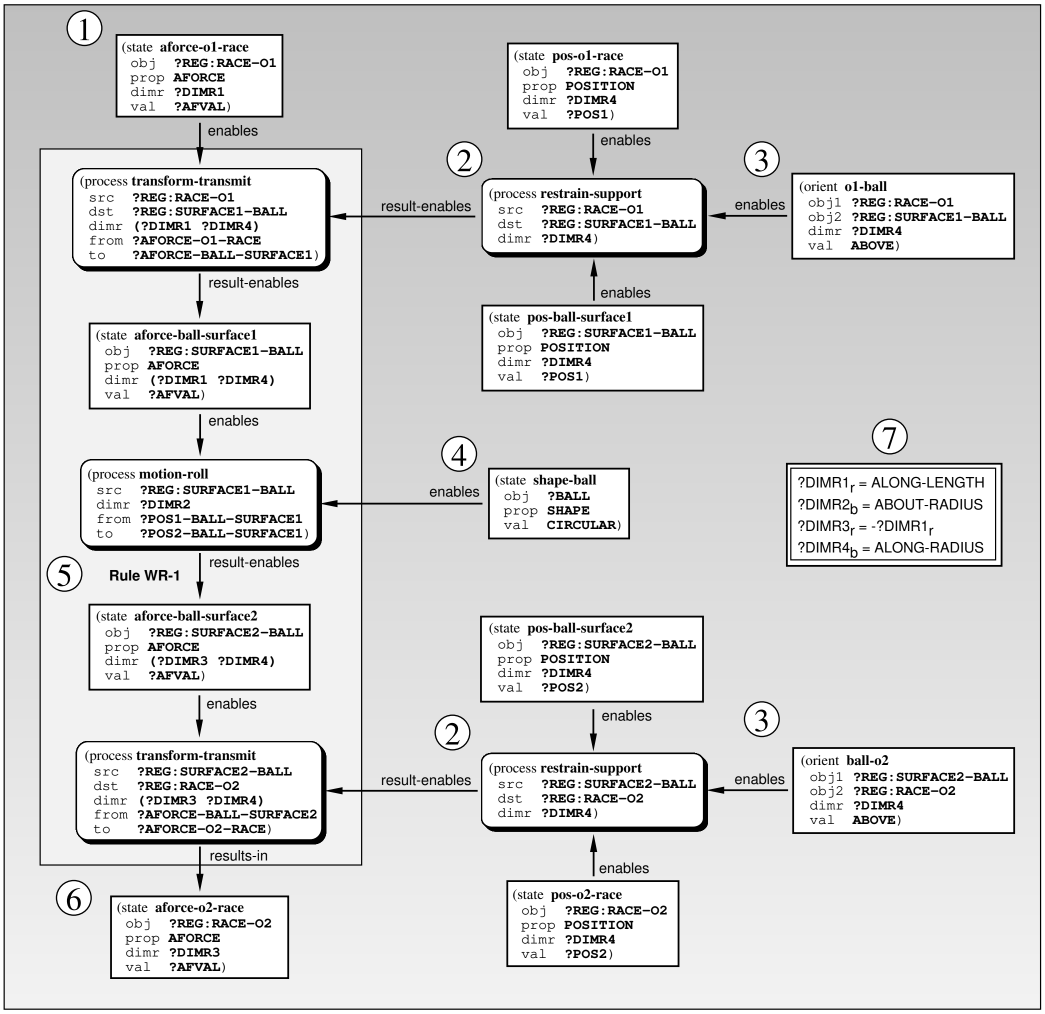
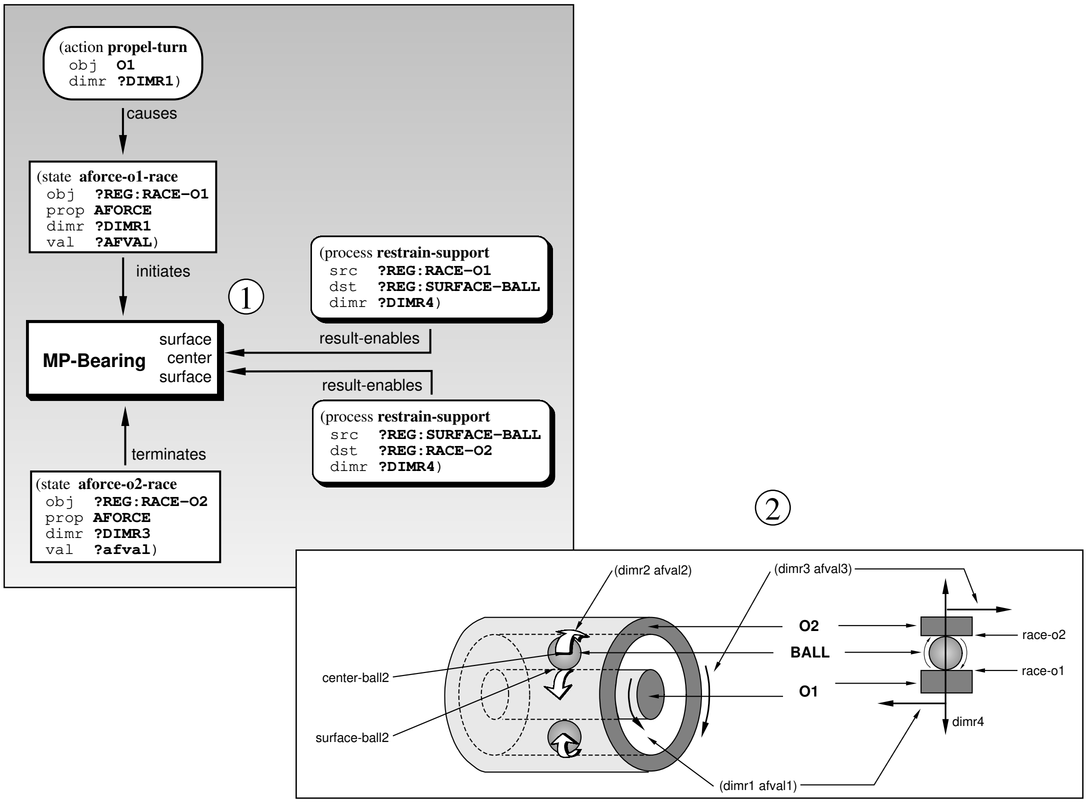


Figure 4.21 Behavioral process sequence associated with the BEARING machine primitive.

Fig. 4.21 illustrates the behavioral processes which describe bearing function. The shaded box represents the intra-object dynamics of the bearing object. The applied force (AFORCE-O1-RACE, at label 1) enables lateral force transmission (TRANSFORM-TRANSMIT) from the bearing race to the bearing object surface region, and motion of the bearing race. The RESTRAIN-SUPPORT processes (2 and 3) between the bearing object and the bearing races enable lateral force transmission from the bearing race to the ball, and radial force transmission to the other bearing race. The bearing object (BALL) supports radial load (DIMR4) because it maintains radial contact between one bearing race (O1), and the other (O2). The circular shape of the bearing object (4), the lateral applied force, and the surface contact between the bearing object and its races enable rolling (MOTION-ROLL) between the bearing object and its races. The force changes direction at the surface2 region on the bearing object according to WR-1 (5), which is then transmitted to the second bearing race (6). The legend labeled (7) shows sample dimensions associated with the illustration of bearing object statics shown at (2) in Fig. 4.22a. The machine primitive associated with the shaded box in Fig. 4.21 is also replaced with its block equivalent at (1) in

Fig. 4.22a.



(a)

```
(function BEARING
  appl     ?SURFACE1-BALL
  pivot    ?CENTER-BALL          ←——— ③
  react    ?SURFACE2-BALL
  from     ?AFORCE-O1-RACE
  to       ?AFORCE-O2-RACE)
```

(b)

Figure 4.22 Block version of MP-BEARING function, bearing object statics, and associated slot-filler representation

The graphic in Fig. 4.22a illustrates two surfaces (O1 and O2) in relative motion, mediated by the rolling contact of the bearing object (BALL). The block version of MP-BEARING (1) is shown being initiated by the force AFORCE-O1-RACE. The state AFORCE-O2-RACE, which represents the force transmitted to RACE-O2, terminates MP-BEARING. The dimensions DIMR1, DIMR2, and DIMR3 are used in the figures to illustrate the relations which are maintained between the processes. DIMR1 and DIMR3 are simply opposing directions of the same dimension, while DIMR2 is the rotational dimension associated with DIMR1 and DIMR3. The force values AFVAL1, AFVAL2, and AFVAL3 are also used for clarity, and have the same value.

4.2.7 MP-CONTAINER

MP-CONTAINER describes devices used to restrict object motion to a specific volume by enabling the process RESTRAIN-CONTACT. MP-CONTAINER has three roles: inside, boundary, and outside which instantiate the function appl and react roles. The container object is a linkage object specialized by an indentation. The container object inside region (the indentation) describes a volume bounded by the container object's inside surface and the container object boundary region. The contained object is free to move within the container object inside region, but not beyond it, except via the boundary. The container object outside and boundary regions can instantiate the MP-LINKAGE appl and react roles. MP-CONTAINER is enabled by a RESTRAIN-CONTACT between the contained object and the container object inside region. MP-CONTAINER is initiated by a force on the container object, or the contained object, enabling the process MOTION for the forced object. An enabled MP-CONTAINER is terminated by a force on the contained object, opposing and disabling its motion. This relationship is depicted in Fig. 4.23.

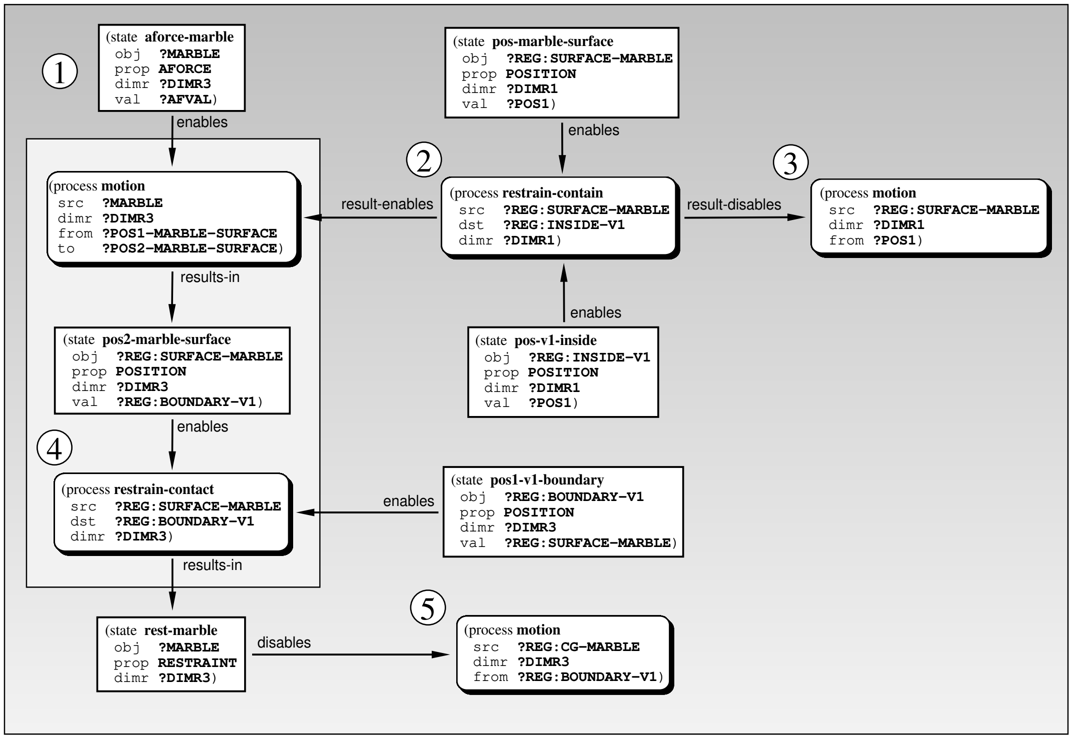
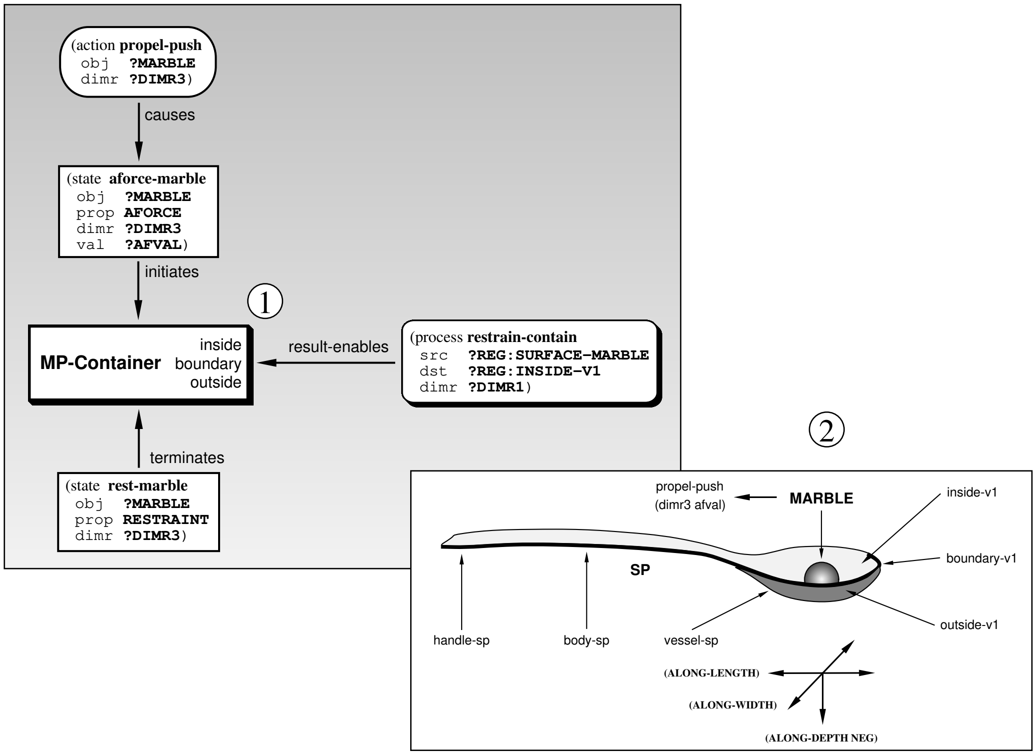


Figure 4.23 Behavioral process sequence associated with the CONTAINER machine primitive.

Fig. 4.23 illustrates the behavioral processes which describe container function, instantiated for the objects depicted in Fig. 4.24. The shaded box represents the intra-object dynamics of the container object. The force (AFORCE-MARBLE, at label 1) enables marble motion. The position of the marble at the same place as the inside of the vessel enables RESTRAIN-CONTAIN (2), which enables lateral motion but disables motion into the boundary (3). The lateral motion results in a new position, which eventually coincides with the vessel boundary (4). The new position enables RESTRAINT-CONTACT and the resulting restraint state disables marble motion (5). Fig. 4.24a (at 1) depicts the block version of the shaded box in Fig. 4.23. The accompanying illustration (at 2) depicts an example container object and contained object statics.



(a)

```
(function CONTAINER
  appl     ?INSIDE-V1
  react    ?BOUNDARY-V1          ← ③
  from     ?AFORCE-MARBLE
  to       ?REST-MARBLE)
```

(b)

Figure 4.24 Block version of MP-CONTAINER function, container object statics, and the associated slot-filler representation

The graphic in Fig. 4.24a illustrates a spoon (SP) as a container object, and a marble (MARBLE) as the contained object. The spoon has three regions: HANDLE-SP, BODY-SP, and VESSEL-SP. VESSEL-SP region is the container object. MP-CONTAINER is shown being initiated by the force AFORCE-MARBLE in DIMR3. The connectivity precondition is that the contained object is within the boundary of the container object, which is represented as an enabled RESTRAIN-CONTAIN. The contact between the MARBLE and the vessel boundary results in a restraint state that terminates MP-CONTAINER.

In Fig. 4.24a, DIMR1 is instantiated by the spoon's thickness axis in the downward direction (ALONG-DEPTH NEG). The MP-CONTAINER behavioral enablement is position of, and contact between, MARBLE and INSIDE-V1 in the (ALONG-DEPTH NEG) dimension/direction. This relationship disables MARBLE motion downward, but not upward or laterally. If DIMR3 is instantiated by the spoon's longitudinal axis, toward its CG (ALONG-LENGTH POS), then MP-CONTAINER causes a motion disablement when

MARBLE comes in contact with INSIDE-V1 in the (ALONG-LENGTH POS) dimension/direction. The same result is caused if the forcing dimension/direction is (ALONG-LENGTH NEG), (ALONG-WIDTH POS), or (ALONG-WIDTH NEG). In FONM, MARBLE's exact location within the container-object's inside region unimportant.

Container-objects may take many static forms. For example, link chains, cables, and ropes instantiate MP-CONTAINER. The link of a link chain instantiates both the boundary and outside container object regions, while the link hole instantiates the inside container-object region. As long as one link is not touching another, it is free to move in any dimension. Cables and ropes can instantiate MP-CONTAINER, since their end region can describe an imaginary volume of containment. If an object is connected to the end region, then it can move freely within the imaginary volume of containment, but not beyond it. Leaks in container objects are represented as additional boundaries. If the size of the boundary is large enough for the contained object, then MP-CONTAINER is disabled because the RESTRAIN-CONTACT is disabled.

4.2.8 MP-SPRING

MP-SPRING describes devices used to produce passive energy storage, in the form of internal force, by enabling the process STORE. MP-SPRING functions by pushing back against a perturbation force, and by returning to its original size and shape, if allowed, when the perturbation force is released. MP-SPRING is a specialization of MP-LINKAGE where the force applied to the spring object is large enough to produce visible size and shape changes but doesn’t exceed the linkage object's material yield strength. MP-SPRING has the same appl and react roles as does MP-LINKAGE. MP-SPRING is enabled by a transmission of the applied force to another object which is, itself, restrained from motion in the forced dimension/direction. MP-SPRING is initiated by a forcing state at the spring object appl region. An enabled MP-SPRING is terminated by two states: an internal force which opposes the perturbation force state, and a size change which follows the perturbation force dimension/direction. An applied force toward the spring object CG results in shortening and an applied force away from the spring object CG results in a lengthening. The amount of internal force and size change is independent of the applied force direction. These relationships are depicted in Fig. 4.25.

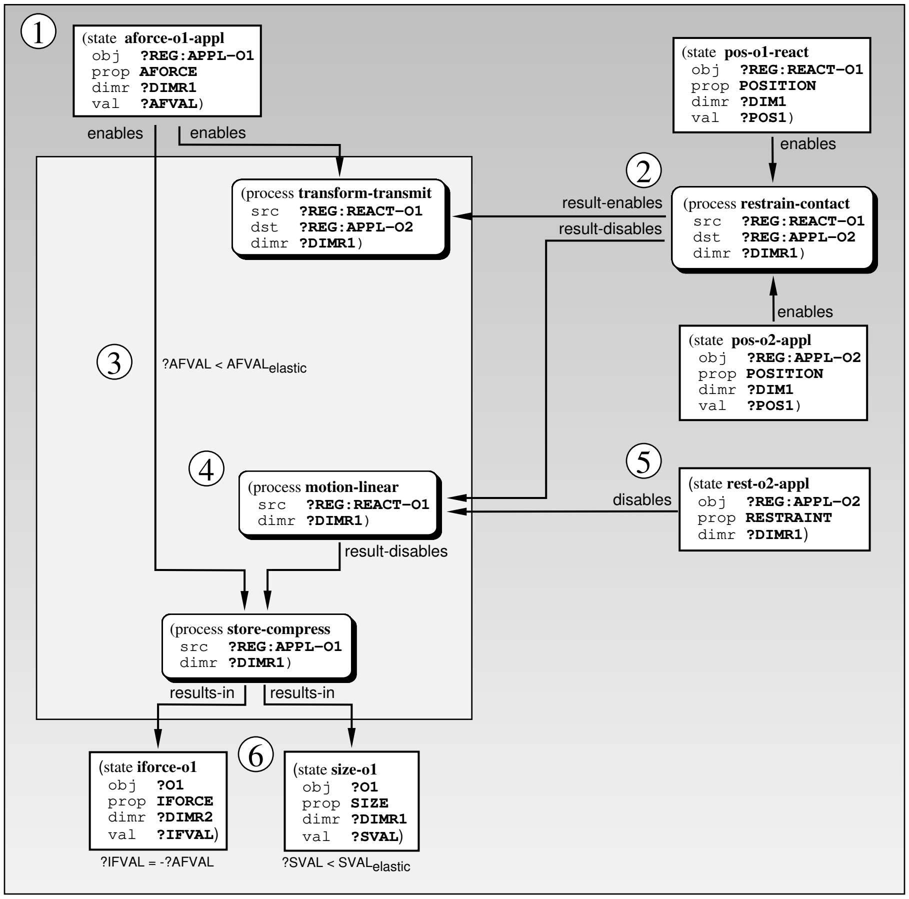
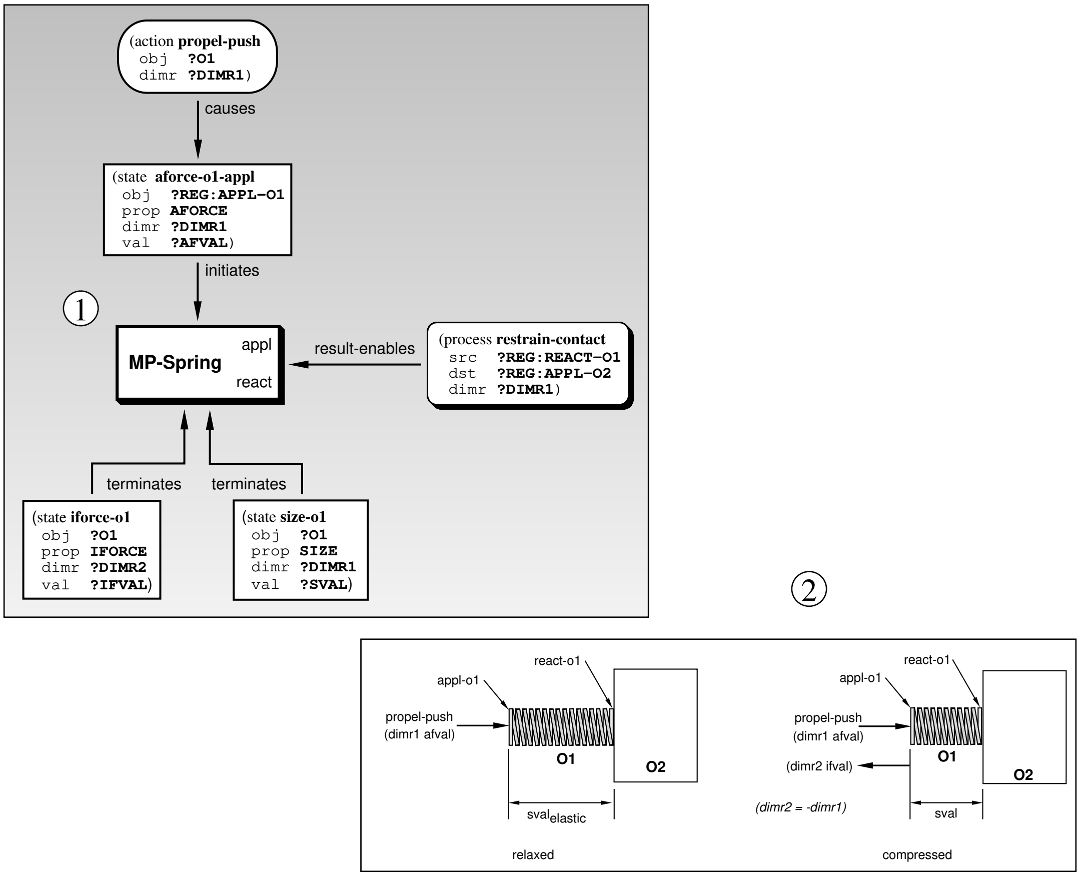


Figure 4.25 Behavioral process sequence associated with the SPRING machine primitive.

Fig. 4.25 illustrates the behavioral processes which describe spring function. The shaded box represents the intra-object dynamics of the spring object. The applied force (AFORCE-O1-APPL, at label 1) enables force transmission to the object (O2) which the spring object is in contact (RESTRAIN-CONTACT, at 2). The applied force also enables passive energy storage (STORE) as long as the force value is less than the spring object's material elastic limit (3). STORE is also enabled by disabled spring object motion (4), which is the combined result of contact with the base object (O2) and a restraint on O2 (5). The states resulting from STORE (IFORCE-O1 and SIZE-O1) are the effect associated with springs. The block version of MP-SPRING is shown in Fig. 4.26a (at 1), along with a example of a coil spring in compression, Fig. 4.26a (2). The slot-filler in Fig. 4.26b shows the function role mapping (3) from MP-SPRING for the block function in Fig. 4.26a.



(a)

```
(function SPRING
  appl     ?APPL-O1
  react    ?REACT-O1
  from     ?AFORCE-O1-APPL
  to       (?IFORCE-O1
            ?SIZE-O1))
```

③

(b)

Figure 4.26 Block version of MP-SPRING function, spring object statics, and the associated slot-filler representation

Fig. 4.26a illustrates a helical-coil spring object (O1) in a relaxed state ($SVAL = SVAL_{elastic}$) and in contact with a block (O2). Once enabled, MP-SPRING causes a change in O1 length from $SVAL_{elastic}$ to ($SVAL < SVAL_{elastic}$), and an internal force in the applied force dimension, opposing direction (DIMR2 IFVAL).

4.2.9 MP-PLANE

MP-PLANE describes devices used to reduce the effort of moving an object in one dimension by moving it in a combination of dimensions. A person walking up a hill is effectively walking up an inclined plane. It is easier to ascend a hill by walking up an incline, or slope, than by climbing straight up. As the hill's slope increases, the ease with which the person ascends the hill, as depicted in Fig. 4.27, decreases. From Applied Mechanics, the overall work is constant regardless of slope, however, lower slopes distribute the work over a larger distance. Therefore the inclined plane, like a lever, is a device which trades off the force applied and the distance through which it is applied.

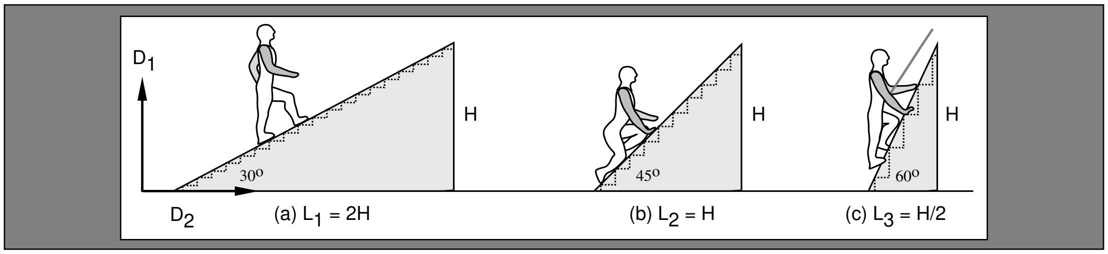


Figure 4.27 Hills as inclined planes.

Fig. 4.27 illustrates a person ascending three slopes, each defined by the ratio of the height he or she will ascend to the horizontal distance they travel to ascend that height. Walking up a shallow-sloped hill is much easier than crawling or climbing steeper slopes, all of which are easier than lifting oneself vertically the same distance. The lower the ratio of H to L, the easier the climb will be but the further the travel.

There is a difference between an inclined plane and a lever. In the lever, there is a tradeoff between force value and the distance between the location of its application and reaction with respect to a pivot location. Neither the applied nor reacted force changes its location on the lever object. The inclined plane trades off the force required for an object to move a distance in dimension D1, with the distance traveled along another dimension D2 to do so. An analogy can be made using the number of steps required to climb a hill. To raise oneself from ground to the top of the hill in a single step takes the greatest single force. For each illustration, right to left, in Fig. 4.27, the number of steps taken increases and the height climbed in each step decreases. The only change is the slope of the hill. An infinite slope is a one-step climb, and requires the maximum force. A zero slope is an infinite-step climb, and requires no force. All slope gradations in-between represent fractions of the maximum force for each step.

The plane object is a linkage object specialized by is triangular shape. The plane object is described by two of the wedge surfaces and one of their edges. One surface is where the perturbation force is applied, and the other surface is where the perturbation force is transmitted to a second object. The edge formed by the intersection of these surfaces forms an acute angle, called an *offset* angle, which acts as a pivot in producing force magnification. MP-PLANE has three roles: appl, offset, and react, which correspond to the MP-LINKAGE appl and react roles, and to the MP-LEVER appl, pivot and react roles. MP-PLANE is enabled by a RESTRAIN-CONTACT between an object and the plane object region which instantiates the MP-PLANE react role. MP-PLANE is initiated by a force at the plane object region which instantiates the MP-PLANE appl role. An enabled MP-PLANE is terminated by a force on the second object in the dimensions and directions described by the plane object region which instantiates the MP-PLANE offset role. MP-PLANE describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the resulting force is magnified. These relationships are illustrated in Fig. 4.28.

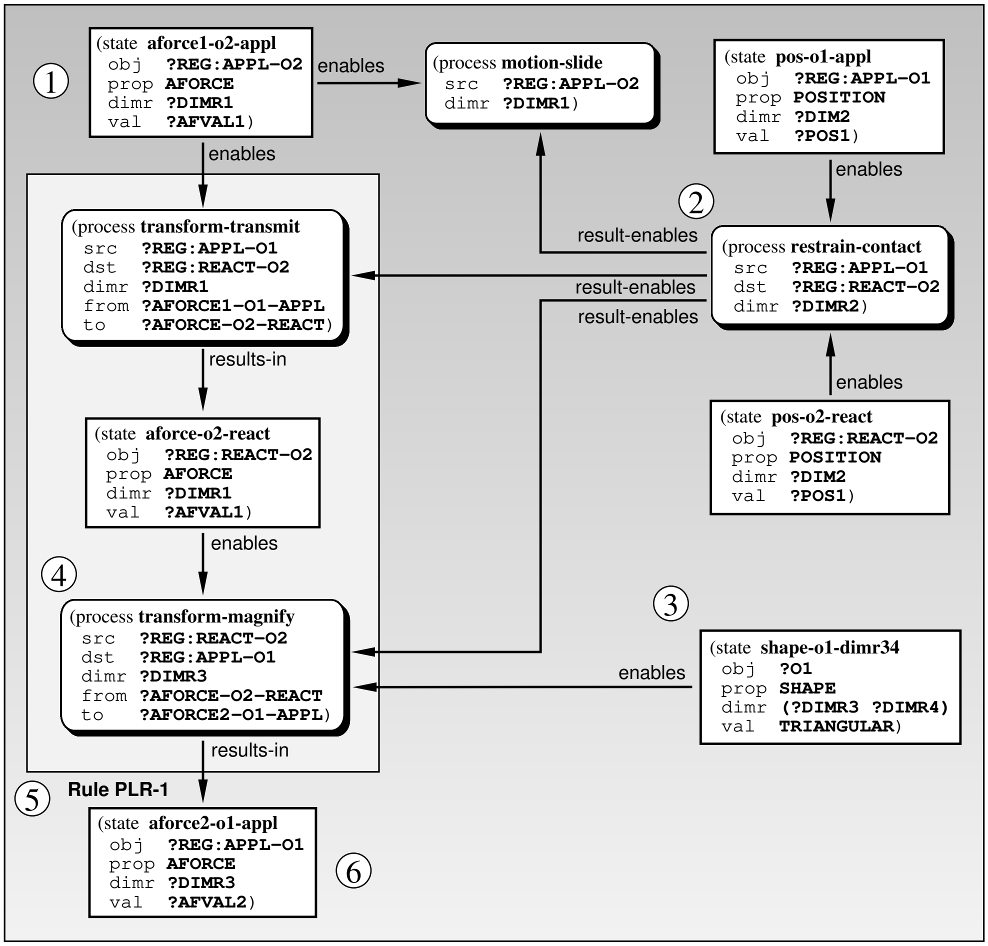
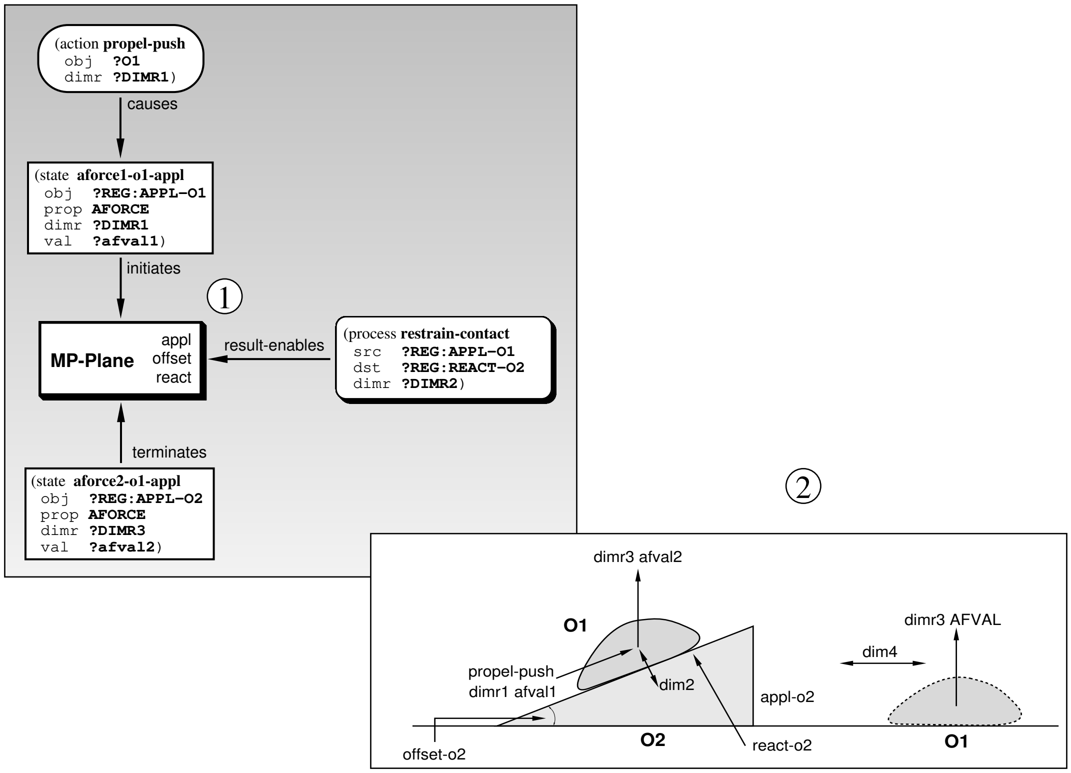


Figure 4.28 Behavioral process sequence associated with the PLANE machine primitive.

Fig. 4.28 illustrates the behavioral processes which describe plane function. The shaded box represents the intra-object dynamics of the plane object. The force (AFORCE-O1-APPL, at label 1) lies in the plane of the wedge (DIM1) and enables sliding along the surface because of the RESTRAIN-CONTACT in the normal dimension (DIM2). The force also enables transmission to the plane object (O2) react region (REACT-O2). The triangular shape of the plane object enables force magnification by dividing it into components in different dimensions (DIM3 and DIM4), resulting in a lower force in dimension DIM3 than if no plane were used. The block version of the shaded box representation of MP-PLANE in Fig. 4.28 is depicted at (1) in Fig. 4.29a. The accompanying illustration, at (2) in Fig. 4.29a depicts plane object statics. The function represented in Fig. 4.29b shows the role mapping from MP-PLANE (3).



(a)

```
(function PLANE
  appl     ?APPL-O2
  pivot    ?OFFSET-O2     ←———————— ③
  react    ?REACT-O2
  from     ?AFORCE1-O1-APPL
  to       ?AFORCE2-O1-APPL)
```

(b)

Figure 4.29 Block version of MP-PLANE function, plane object statics, and the associated slot-filler function representation

The graphic in Fig. 4.29a illustrates a wedge (O2) and a rock (O1). In this illustration, the rock is shown as the object in motion. As with all processes in FONM, there is no difference between the wedge moving toward the rock or the rock moving toward the wedge. The force AFORCE1-O1-APPL enables O1 sliding and initiates MP-PLANE. The contact between the rock and the wedge, and the shape of the wedge, enable MP-PLANE. The state AFORCE2-O1-APPL, which represents the force converted by the wedge into dimensional components, in DIMR3, terminates MP-PLANE.

When considered as an MP-LEVER pivot point, the plane object offset angle provides an effective measure of motion advantage. Consider a plane object with equal appl and base surface lengths. The ratio of these lengths (L_{APPL}/L_{BASE}), called an *offset ratio*, is 1. This value is defined as nominal, or NOM. This plane object is analogous to MP-LEVER-TYPE1, where the lever object pivot region is located halfway between the lever object appl and react regions. Movement of the lever object appl region results in an equal and opposite movement of the react region. In MP-PLANE, the movement of the plane object O1 along DIMR1 results in an equal movements of O2 along DIMR3 and DIMR4, as shown in rule PLR-1:

PLR-1: $dist_{d3}/dist_{d4} = l_{appl}/l_{base}$ (4-5)

PLR-2: $afval_{d3} = afval1_{d3} * l_{appl}/(l_{appl} + l_{base})$ (4-6)

where dist is distance travelled, d2 is the dimension of the appl surface, d1 is the dimension of the base surface, l_{appl} is the length of the appl surface, and l_{base} is the length of the base surface. When the plane object offset ratio is less than nominal ($L_{APPL}/L_{BASE} < 1$), MP-PLANE is similar to MP-LEVER-TYPE2: it is most effective for moving an object in DIMR4, with a <NOM force value (rule PLR-2). When the offset ratio is greater than nominal ($L_{APPL}/L_{BASE} > 1$), MP-PLANE is similar to MP-LEVER-TYPE3: it is most effective for moving an object in DIMR3, with a greater rate than the nominal rate, and a resultant force approaching the original nominal force as the base length approaches zero.

4.2.10 MP-BLADE

MP-BLADE describes devices used to cut or otherwise permanently deform objects, by enabling the DEFORM process, through the application of linear mechanical advantage and linear motion advantage. MP-BLADE function results from the combination of two effects. First, the blade object is a wedge. The appl surface area of the wedge is much larger than the edge surface area, so forces applied there are magnified at the edge (see Fig. 3.15). This is linear mechanical advantage. Second, as a wedge the blade object is also a plane object. Once a cut is initiated the MP-PLANE enables the blade object to advance by pushing the material aside. This is linear motion advantage. The blade object has three regions that instantiate MP-BLADE roles: appl, edge, and react. The blade object edge region describes the intersection of the plane object react surface and base surface. MP-BLADE is specialized from MP-PLANE by perturbing the plane object toward the second object edge on, as illustrated in Fig. 4.30.

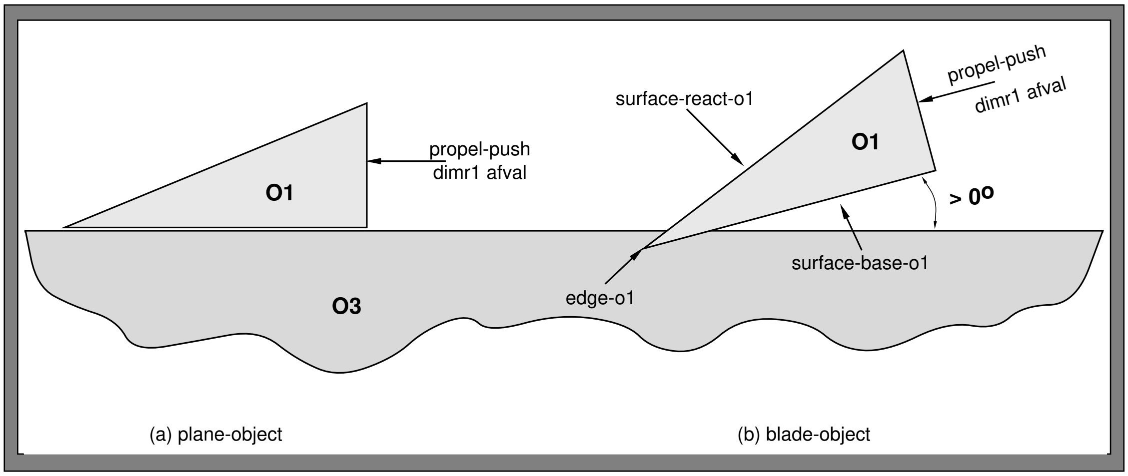


Figure 4.30 Application of (a) plane object versus (b) blade object.

Fig. 4.30 illustrates the difference in application that distinguishes the MP-BLADE from the MP-PLANE, depicted with plane objects. The term edge on refers to the orientation of the blade object to the surface it contacts. In MP-PLANE, the plane object edge is parallel (angle = 0^{o}) with respect to the surface upon which it rests, so there is no edge contact, Fig. 4.30a. In MP-BLADE, the blade object itself is oriented at an offset to (angle > 0^{o}) substrate, so its edge impinges on the substrate as it moves. The blade object react region describes both top and bottom wedge faces.

MP-BLADE is enabled by a RESTRAIN-CONTACT between the blade object react region and the substrate. MP-BLADE is initiated by a force applied at the blade object appl region. An enabled MP-BLADE is terminated by a change in size of the substrate object. MP-BLADE describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the resulting force is magnified. These relationships are depicted in Fig. 4.31.

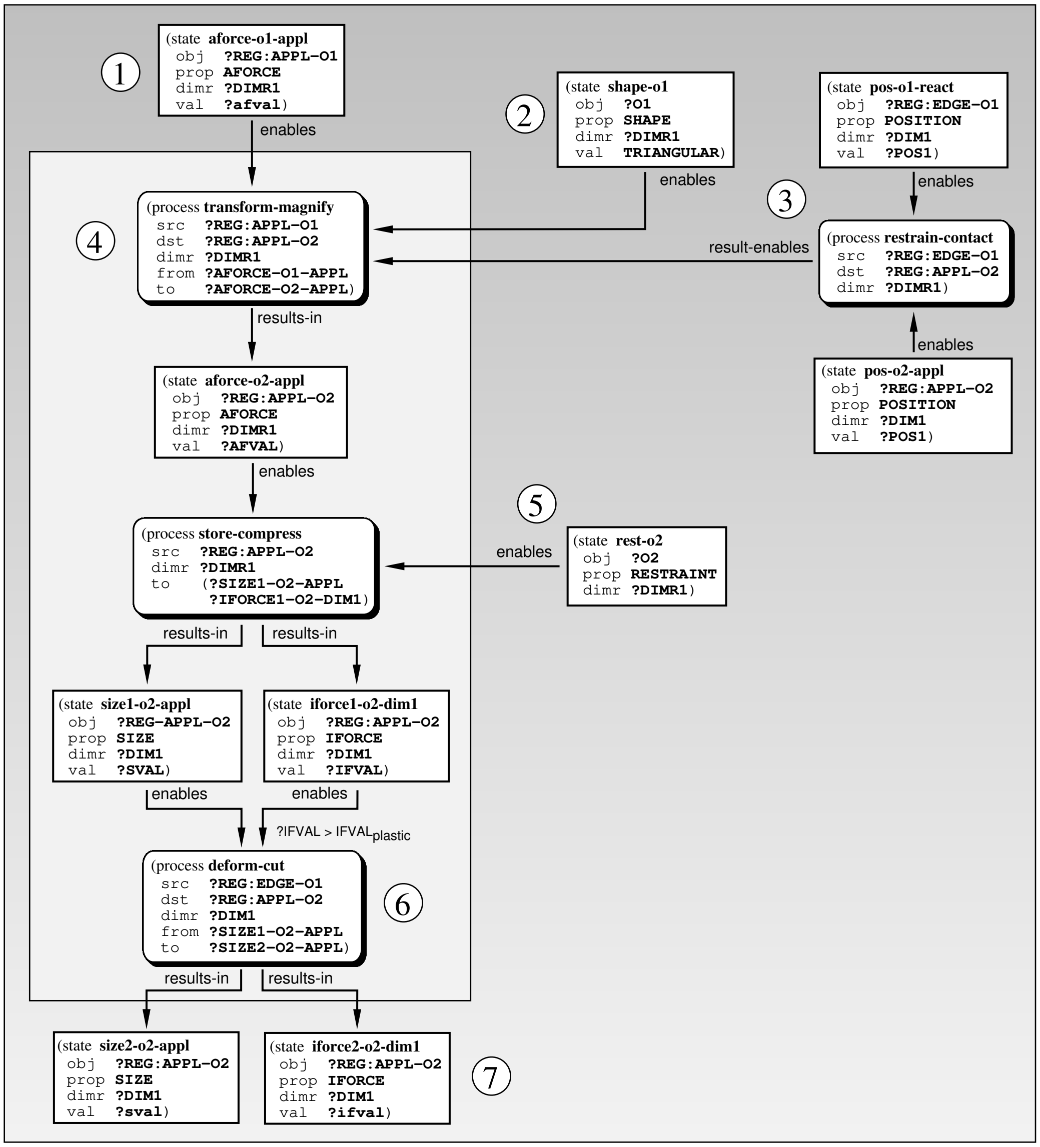
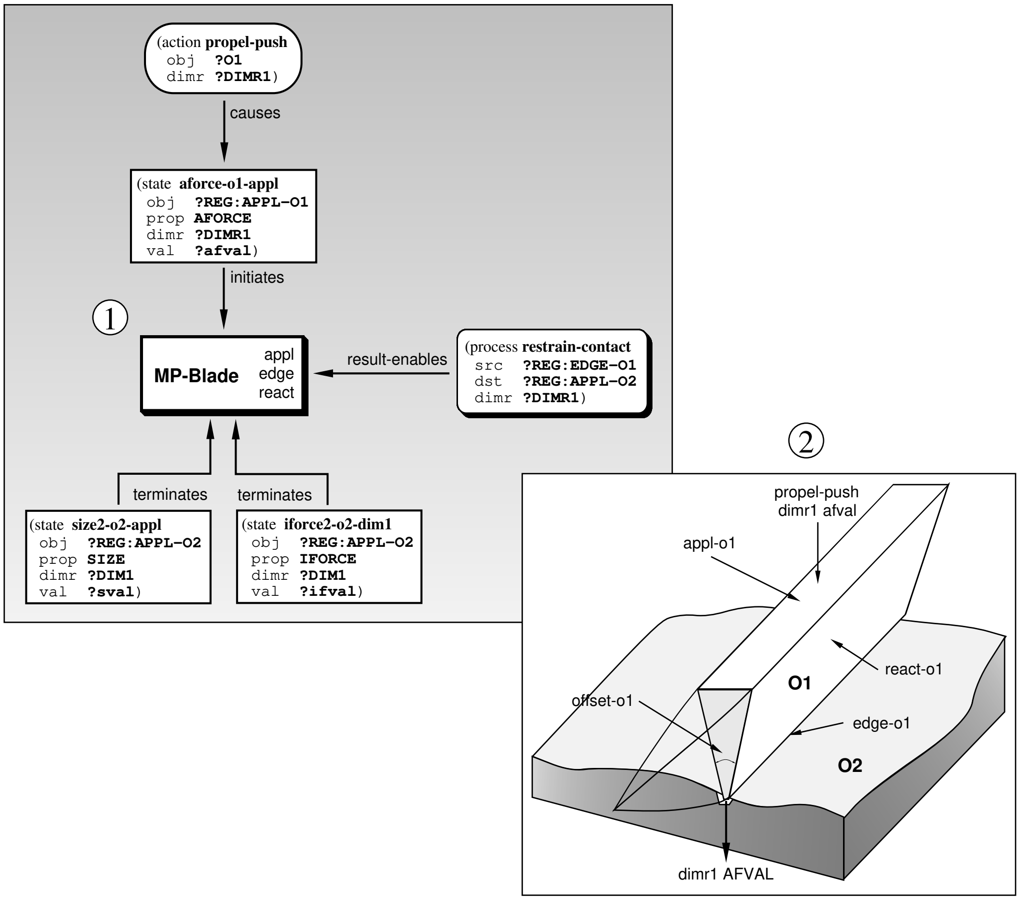


Figure 4.31 Behavioral process sequence associated with the BLADE machine primitive.

Fig. 4.31 depicts the behavioral processes which describe blade function. The shaded box represents the intra-object dynamics of the blade object. The force (AFORCE-O1-APPL, at label 1), triangular shape (2) of the blade object (O1), and edge contact (RESTRAIN-CONTACT, at 3) with the substrate (O2) enable magnification of force (4). The restraint on O2 (5) motion enables compression, and, if the applied force exceeds the breaking strength of O2, deformation is enabled (6) and O2 undergoes local deformation (7). The block version of the shaded box in Fig. 4.31 is illustrated at (1) in Fig. 4.32a. An example depicting blade object statics, with knife slicing, is shown at (2) in Fig. 4.32a. A mapping from MP-BLADE roles to function roles is shown in Fig. 4.32b.



(a)

```
(function BLADE
  appl     ?APPL-O1
  pivot    ?EDDGE-O1      ◄──────────────── ③
  react    ?REACT-O1
  from     ?AFORCE-APPL-O1
  to       (?SIZE-O2-APPL
            ?IFORCE2-O2-DIM1))
```

(b)

Figure 4.32 Block version of MP-BLADE function, blade object statics, and the associated slot-filler function representation.

The graphic inFig. 4.32a illustrates a knife (O1) and a substrate material (O2). The blade object (O1) is propelled (PROPEL-PUSH) in DIMR1 toward the surface of O2. The process RESTRAIN-CONTACT between the O1 edge and O2 enables the process TRANSFORM-MAGNIFY to O2. MP-BLADE is initiated by the force AFORCE-O1-APPL. The state SIZE-O2-APPL terminates MP-BLADE.

4.2.11 MP-SCREW

MP-SCREW describes devices used to transform rotational motion to linear motion by applying rotational mechanical advantage and rotational motion advantage. A screw object is a plane object wrapped around a cylindrical or conical object, as shown in Fig. 4.33a. The screw object has three regions: appl, thread, and react. The appl and react regions are surfaces. The react region is shaped like a blade, called a *thread*, and, depending on whether the thread is sharp or blunt, the screw can be used to cut. The screw object cylindrical shape, and location of its appl region, instantiates a Type 2 MP-WHEEL-AXLE. MP-SCREW is enabled by a RESTRAIN-CONTACT between the screw object react region and the substrate material appl region. MP-SCREW is initiated by a forcing state at the screw object appl region. An enabled MP-SCREW is terminated by a force on the substrate object. MP-SCREW describes the enablements and resulting state of the process TRANSFORM-MAGNIFY, so the value of the resulting force is magnified. These relationships are depicted in Fig. 4.33.

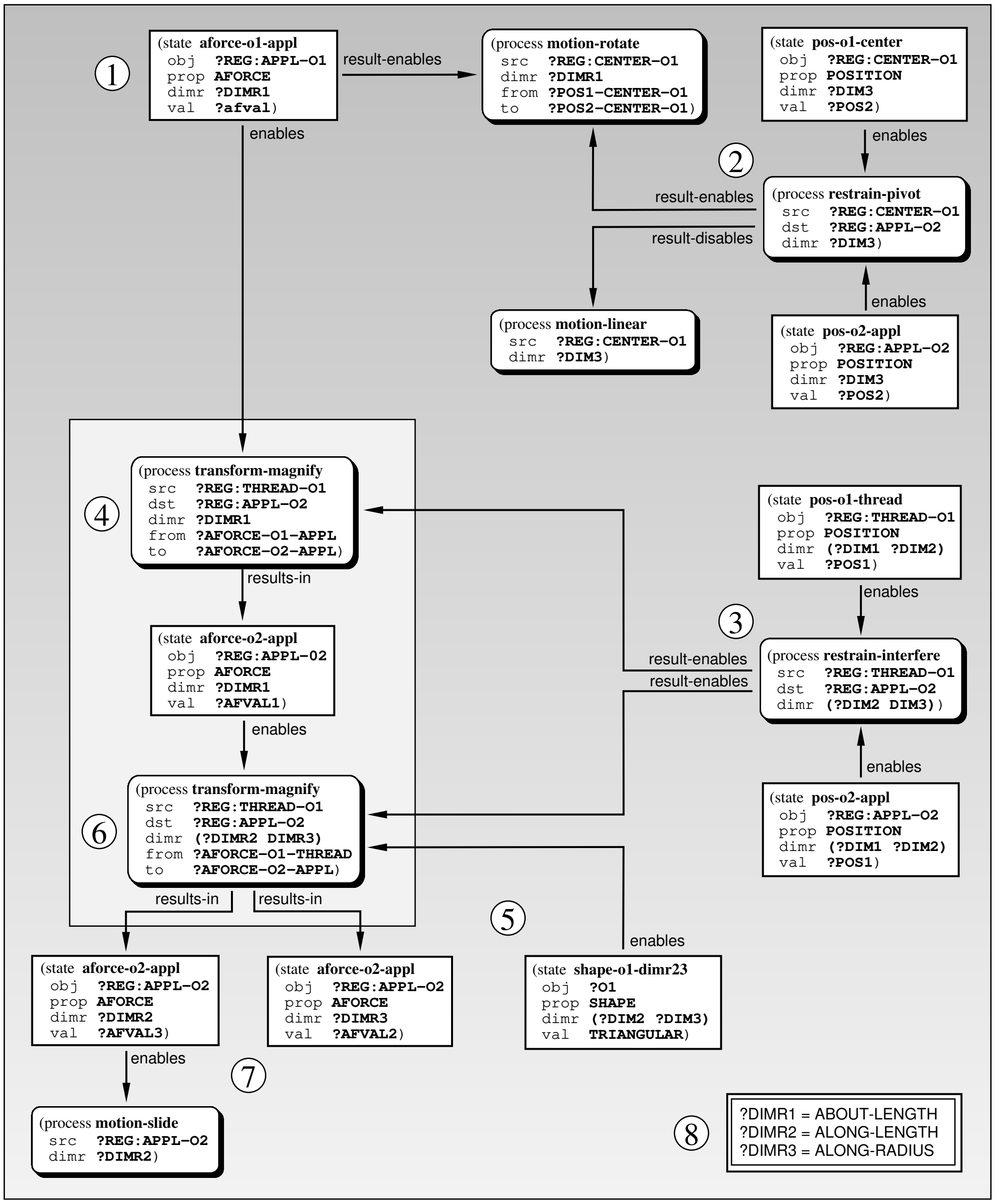
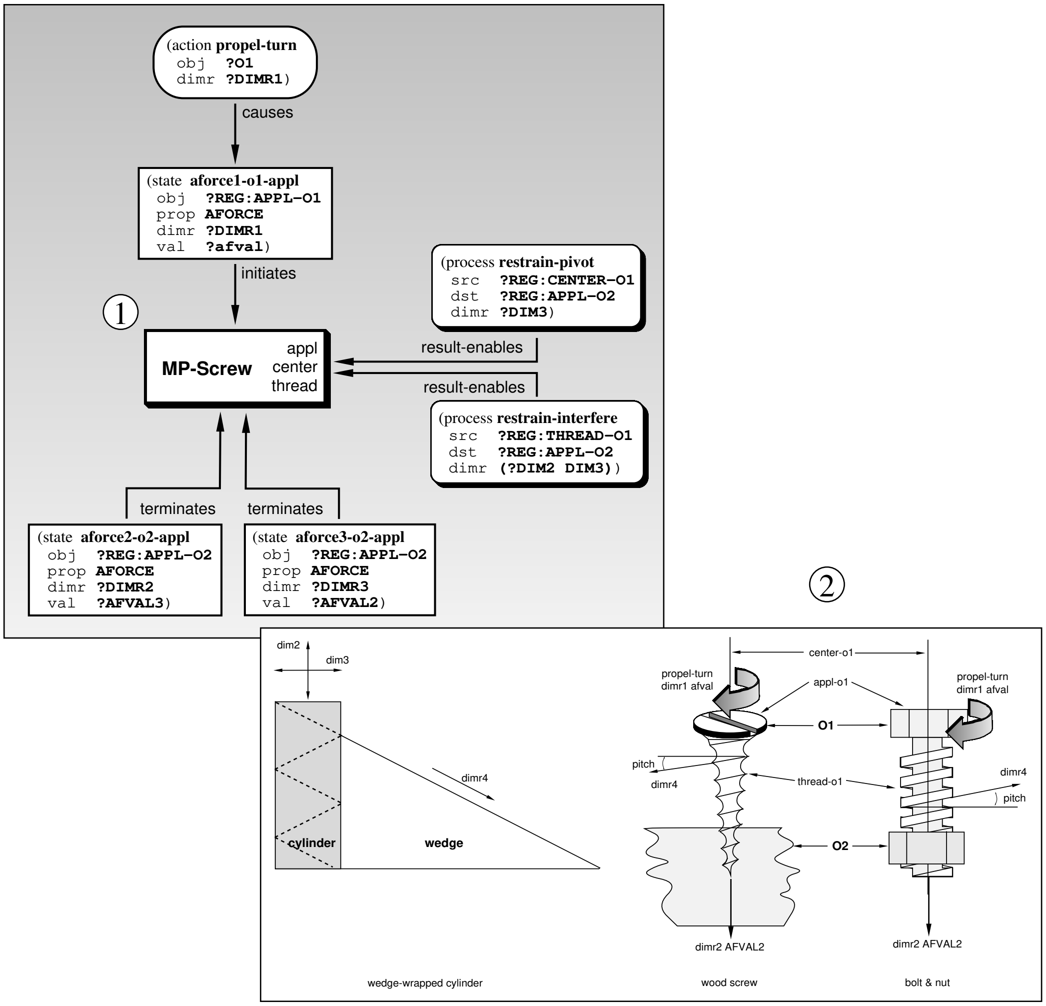


Figure 4.33 Behavioral process sequence associated with the SCREW machine primitive.

Fig. 4.33 illustrates the behavioral processes which describe screw function. The shaded box represents the intra-object dynamics of the screw object. The force (AFORCE1-O1-APPL, at label 1) enables rotation about the center, which is pivoted (RESTRAIN-PIVOT, at 2) by the substrate (02). The screw object thread (a protuberance) is in interference contact (RESTRAIN-INTERFERE, at 3) with the substrate. The contact relation at (2 and 3) are identical, separated in Fig. 4.33 to show the different effects they have on screw function. The pivot and applied force enable rotational mechanical advantage (TRANSFORM-MAGNIFY, at 4), whereas the applied force, interference, and triangular shape (6) enable rotational motion advantage (TRANSFORM-MAGNIFY, at 7). The result is applied force in a translational dimension. The legend at (8) provides an example of the dimension dependencies between the processes in Fig. 4.33, and corresponds to those depicted at (2) in Fig. 4.34a. The block version of the shaded box in Fig. 4.33 is illustrated at (1) in Fig. 4.34a. The role mapping between MP-SCREW and the generic function class is shown with the slot-filler notation for MP-SCREW function in Fig. 4.34b.



(a)

```
(function SCREW
  appl     ?APPL-O1
  pivot    ?CENTER-O1        ←        ③
  react    ?THREAD-O1
  from     ?AFORCE1-O1-APPL
  to       (?AFORCE2-O2-APPL
            ?AFORCE3-O2-APPL)
```

(b)

Figure 4.34 Block version of MP-SCREW function, screw object statics, and the associated slot-filler function representation.

The graphic in Fig. 4.34a illustrates the concept of screw objects and three screw objects: a wood screw, a bolt, and a nut (the bolt and nut are illustrated together). The wood screw object (O1) has a head and a point. Its threads (THREAD-O1) are wedge-shaped. This screw object is a combination of MP-SCREW and MP-BLADE, because the wedge-shaped threads and point both instantiate MP-BLADE.

4.3 MP Combinations and Mechanical Device Representation

A device has one static description at any point in time and that description can change based on the device's behavioral interactions with other objects. A device has one dynamic description for each function it can be applied to, but the function, being a dynamic description of the device, doesn't change with time. Behavioral processes can be used to describe an object's function, but they tend to be too low-level for most mechanical problem solving tasks, and are best suited for simulation and diagnostic analysis. Machine primitives can be associated with device components or the functions of device components, and can be combined to describe the functions of entire devices. Two machine primitives can be combined (e.g., the gears in Fig. 4.17 and the pulleys in Fig. 4.19) as long as their roles, enabling preconditions, and resulting states match. Machine functions can be conceptually put together, using the block versions of machine primitives (e.g., Fig. 4.10), like Tinker Toy pieces.

Consider a screwdriver like the one illustrated in Fig. 4.35a, which is designed to drive screws. The same screwdriver device can also be used to pry the lid off a can or to puncture materials and so it has more than one function, depending on the context in which it is applied. Fig. 4.35 illustrates the relationship between device statics and the representation of device functions using MP blocks. The screwdriver in Fig. 4.35a consists of three regions (or components): HANDLE-SD, SHAFT-SD, and TIP-SD. Fig. 4.35b shows the FONM idealizations of the screwdriver components and their static appearance. Fig. 4.35c illustrates the FONM static representation of the screwdriver.

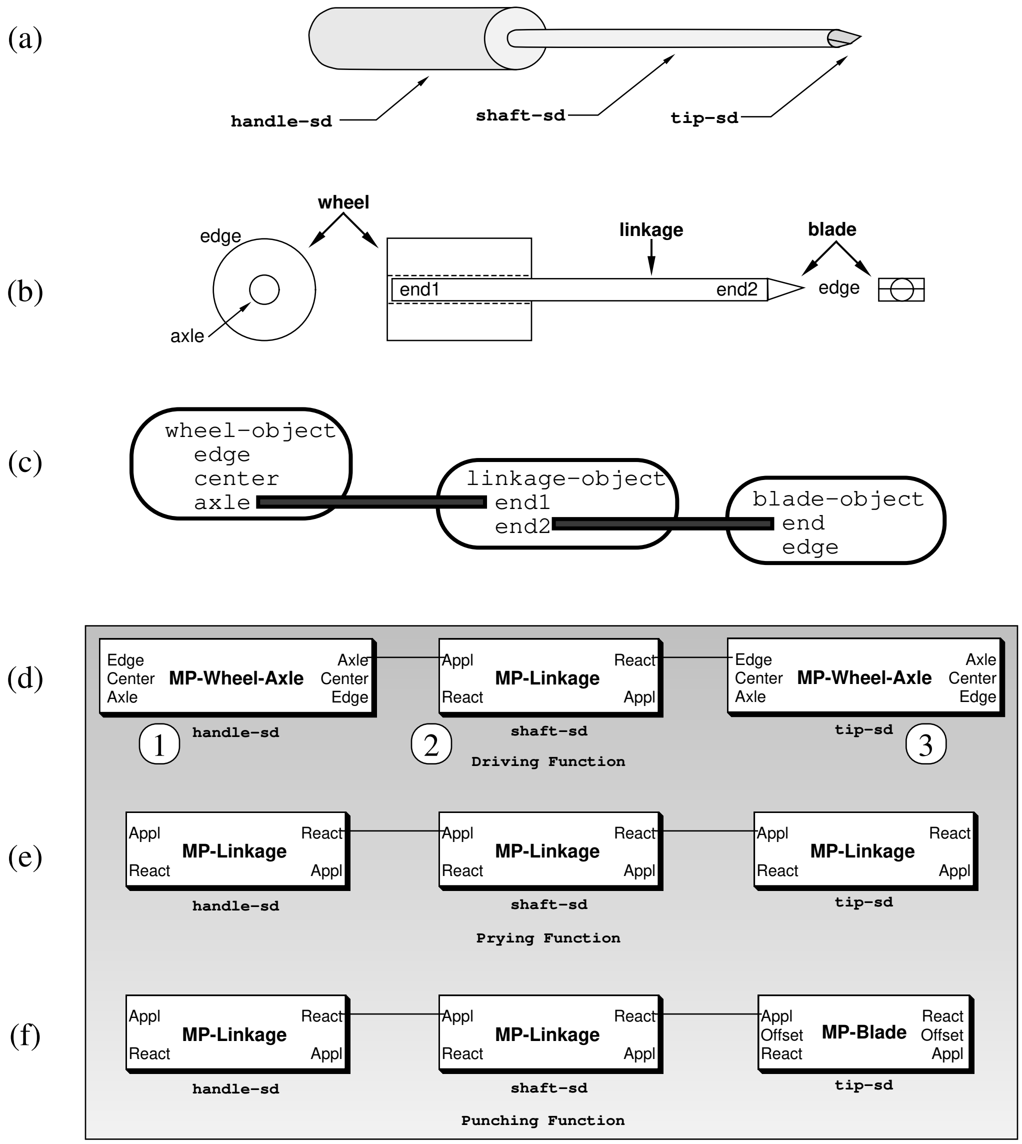


Figure 4.35 Screwdriver driving function as a combination of machine primitives.

A representation of the screwdriver driving function can be constructed from machine primitive blocks, as shown by the MP-Diagram in Fig. 4.35d. Using the same static representation, two alternate MP-Diagrams, one each for the prying and punching functions, can be constructed, and are illustrated in Fig. 4.35e and Fig. 4.35f.

The MP blocks in these figures are related to one another through links which represent the dynamic interaction of their instantiating components. For example, the static diagram in Fig. 4.35c shows that the screwdriver handle (HANDLE-SD) and shaft (SHAFT-SD) are rigidly connected. The links in Fig. 4.35d-f don't specify the type of connection, because the primitives must each satisfy the other's preconditions. Labeling the connection between components as a dynamic link indicates how force and motion are propagated throughout the device. For example, MP-WHEEL-AXLE magnifies force at its axle, which is where the screwdriver shaft is connected. The force resulting from MP-WHEEL-AXLE both initiates MP-LINKAGE and specializes MP-LINKAGE to MP-LINKAGE-TYPE3 (torsional).

Each MP-Diagram in Fig. 4.35 (d-f) represents the dynamics of the screwdriver under a different applied force. In Fig. 4.35d, the force is applied by turning the handle, so it is a rotational force (or torque) that is applied. Fig. 4.35d represents the screwdriver driving function. The function of the handle component (HANDLE-SD) is represented with MP-WHEEL-AXLE (1), and is used to produce the necessary torque for driving screws. The function of the shaft (SHAFT-SD) is represented with MP-LINKAGE (2), and extends the torque force from HANDLE-SD to TIP-SD where the screw is located. The tip is wedge-shaped so that it will fit the screw slot, but magnifies the applied torque so its function is represented with MP-WHEEL-AXLE (3).

In Fig. 4.35e the applied force is lateral, and in Fig. 4.35f the applied force is longitudinal. Since machine primitives are represented with their initiating conditions, the effect of changing the type of applied force is to change the primitives which can represent what the associated object can be used to accomplish. The lateral force cannot instantiate MP-WHEEL-AXLE unless there is a different kind of pivot (as for a crank), but it can instantiate MP-LINKAGE-TYPE2. Similarly, the longitudinal force can instantiate MP-LINKAGE-TYPE1. The primitives in a combination are dependent on one another, to satisfy their enablements, so changing the primitives that can be instantiated by the applied force has an effect on the representation of the entire function. The functions illustrated in Fig. 4.35e and f show that the same static representation and a different applied force can instantiate different functions.

4.4 High-Level Device Representations

This section illustrates the dynamic representations of the 6 devices whose static and low-level dynamic representations were presented at the end of Chapters 2, and 3. The four device types presented: (1) simple devices, (2) simple compound devices, (3) multiple compound devices, and (3) complex devices are revisited with an emphasis on their high-level dynamics. The screwdriver, for example, is a simple device. Statically it is simple because all of its components are rigidly connected. Dynamically it is simple because all of its components move together, so the device function is effectively the function associated with the handle. That is, if one turns the handle, the entire screwdriver functions like an MP-WHEEL-AXLE, and if one pushes on the handle the entire screwdriver functions like an MP-LINKAGE. When an entire device functions as the machine primitive which one or more of its components instantiates, it is called a *generalized machine*. The screwdriver in the driving function is a generalized MP-WHEEL-AXLE.

The device representations presented in the following sections will take the form of the first example in this section. An idealized illustration of a possible device geometry, based on object primitives, will be shown, along with a static representation diagram for the primitives and one or more MP-Diagrams for the device pictured. It is assumed that the behavioral sequences associated with each MP block, and in particular the inter-object dynamics (e.g., motion) can be instantiated for each MP-Diagram, rather than fleshing them out explicitly. When appropriate, representation examples will be presented. As the static description becomes more complex, fewer independent MP-Diagrams will be presented, but different ways to perturb them will be discussed.

4.4.1 Simple Devices

Simple device dynamics are representationally significant because even simple devices can be used to accomplish multiple tasks. The dynamic representation must make a device's different functions explicit with the same static representation. The distinction between device functions is often seen as a specialization on the type of input applied to a device, or on the physical properties of objects which comprise the device. Simple devices are also useful for showing that device descriptions which represent the same function, whether based on components or regions, are equivalent. The high-level dynamic representations of two simple devices: (1) a bottle opener, and (2) a screwdriver, are presented in this section. Each device is comprised of components which are rigidly connected, or regions of a single component. The high-level dynamic representations for four additional devices: (3) a carving knife, (4) a hammer, (5) a shovel, and (6) a spoon are presented in Appendix A.1.

Bottle Opener

A bottle opener is designed to open cans using one end and to open bottle caps at the other. Both ends make use of mechanical advantage, but one end is pointed so the magnified force is used to puncture. An idealized bottle opener is represented as two lever-objects and a blade object, as shown in Fig. 4.36 (redrawn from Figs. 2.58, 2.59) below.

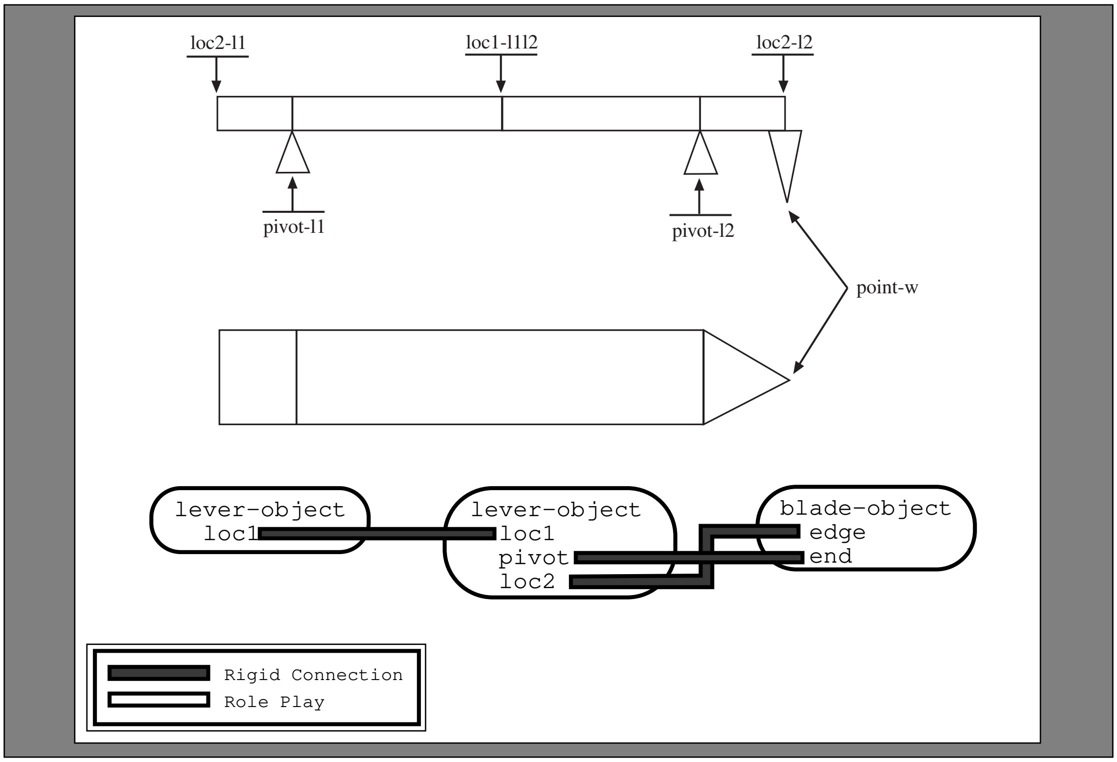


Figure 4.36 Bottle opener object primitives and static device diagram

The static representation for the bottle opener depicted in Fig. 4.36 can instantiate different functions depending on how and where the force is applied. Three bottle opener functions are represented as MP-Diagrams in Fig. 4.37: lid puncturing, lid prying, and a probing function. In each case, the regions shown in Fig. 4.36 and Fig. 2.59 are shown instantiating the roles of the machine primitives they instantiate in the respective function. The legend in Fig. 4.37 describes the arcs between the blocks, and will be replicated in all MP-Diagrams in this section. Each function depicted differs by where the force is applied and what type of force is applied.

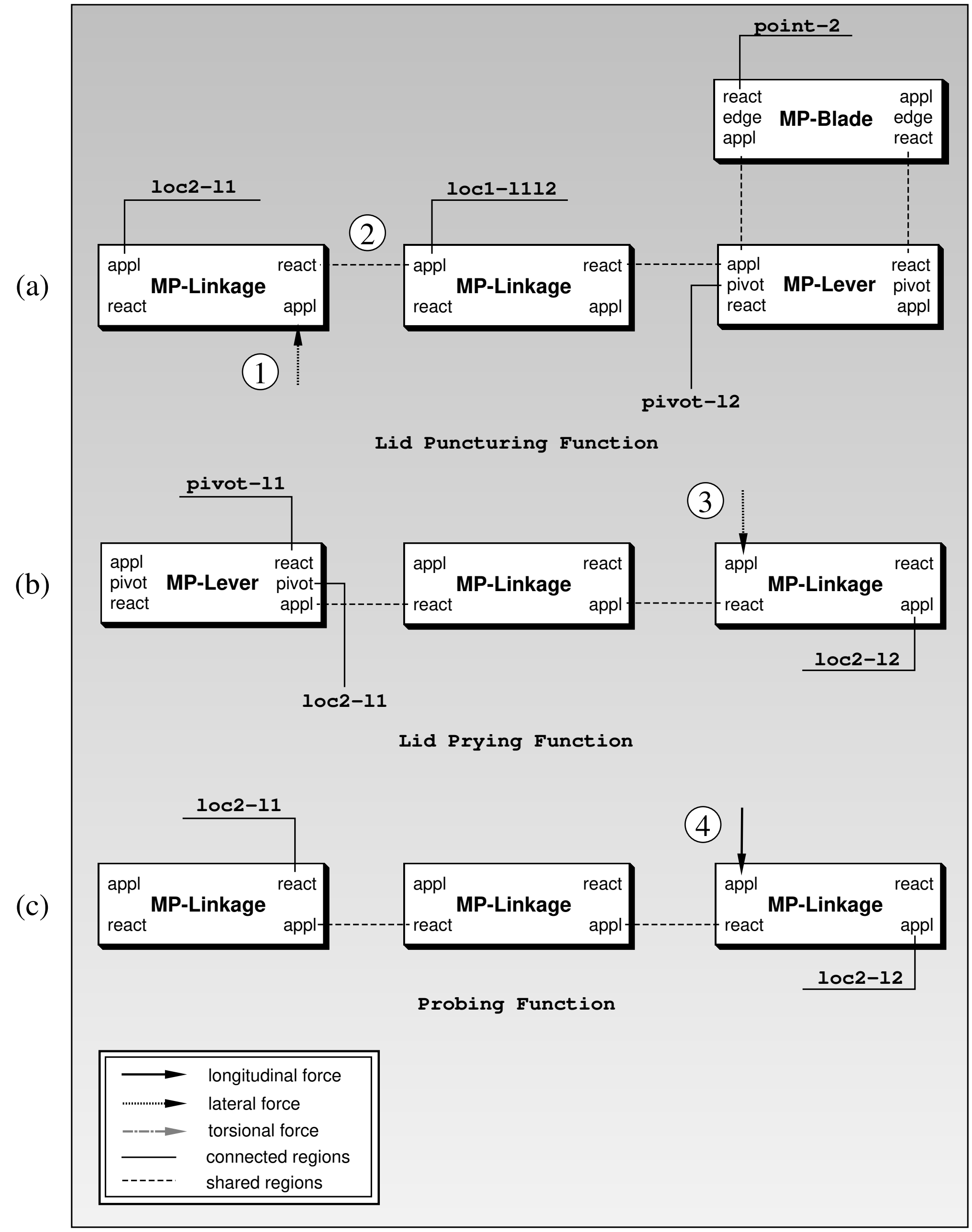


Figure 4.37 MP-Diagrams for bottle opener functions.

The input which identifies the bottle opener function for puncturing cans is a lateral applied force at the end opposite the wedge object (region LOC2-L1, shown at 1). The applied force is lateral, which, based on the static device representation, enables MP-LINKAGE. The connection between the objects is illustrated in the figure as a dotted line (2), which means that the roles of the respective machine primitives are instantiated by the same object region. In the idealized bottle opener, the device is represented as three objects connected together, so this arc could also be a solid line. The central component transmits the lateral force from one end to the other, so it is represented with MP-LINKAGE also. The pointed end has a pivot location and, when the point is in contact with a can, can be represented with MP-LEVER-TYPE1.

The lid prying function illustrated in Fig. 4.37b is similar to the lid prying function, in that the same type of force is applied on the other end of the device (3), and the component functions are all represented with the same machine primitives. However, when applied in removing bottle caps, the end of the bottle opener becomes the region which instantiates the lever pivot role, and so the component is represented with MP-LEVER-TYPE2. The difference is in the shape of the components which instantiate the primitives. The pointed end of the bottle opener cannot be applied as a pivot without enabling the DEFORM process.

The last function illustrated shows the effect of changing to a longitudinal force (4). Similar to the functions illustrated in Fig. 4.37a and b, had the force been applied at the other end, another function would be instantiated, that of longitudinal puncturing. It is significant to note the graphical similarity between the lid puncturing function and the lid prying function. In general, devices which instantiate the same combination of machine primitives are functionally similar (but not necessarily functionally equivalent).

Screwdriver

The idealized screwdriver and its static device diagram are depicted in Fig. 4.35a-c. The functions illustrated in Fig. 4.35d-f are re-illustrated in Fig. 4.38 for continuity.

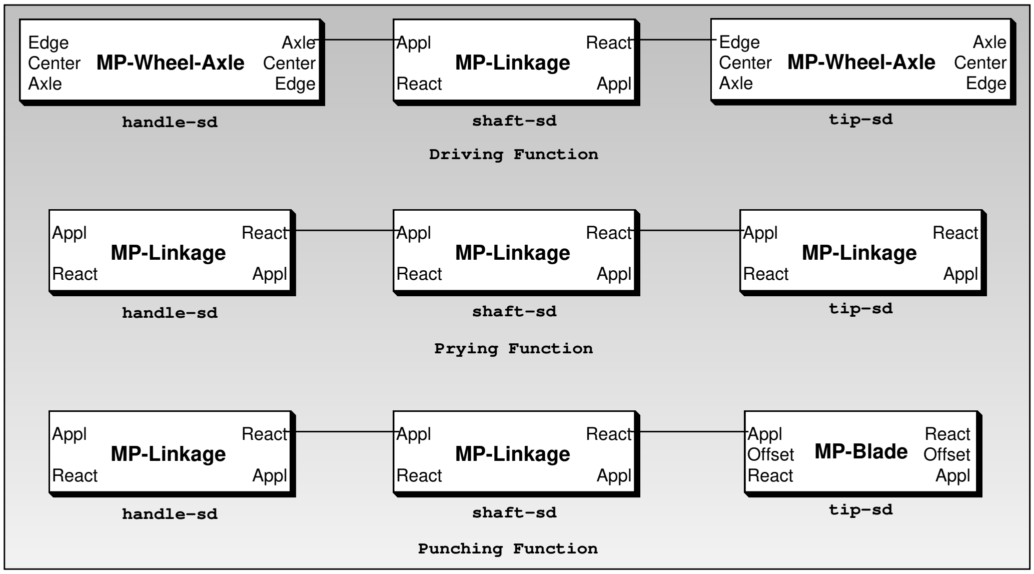


Figure 4.38 MP-Diagrams for screwdriver functions.

4.4.2 Simple Compound Devices

Devices in which relative motion between components is possible are *compound* devices. Compound device dynamics are representationally significant because components can function independently. This has two affects on the representation and interpretation of compound devices. First, if a device component is not rigidly connected to other components, then the functions it can instantiate can be invoked without considering the other device components. Second, if a machine primitive participates in a device's function representation, then its function can be considered independently of the function in which it participates. In this section, two devices: (1) a nutcracker, and (2) a clothes pin are presented. Appendix A.2 presents the MP representations of three additional compound devices: (3) a pair of scissors, (4) a pair of pliers, and (5) a door.

Nutcracker

A nutcracker is used to restrain the motion of and crack the shell of a nut. The nutcracker and pliers are similar in that they perform the same function in different ways. An idealized nutcracker and its static device diagram are illustrated in Fig. 4.39.

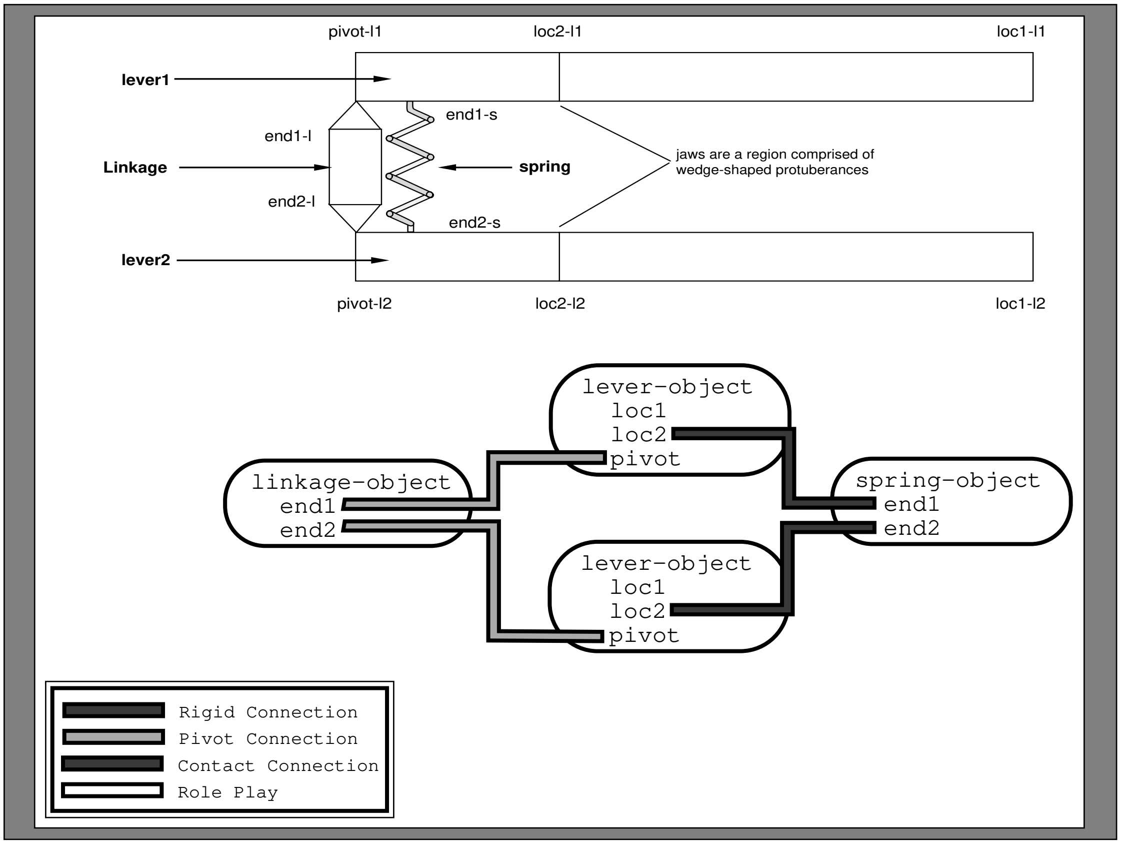


Figure 4.39 Nutcracker object primitives and static device diagram.

The cracking nutcracker function is depicted in Fig. 4.39. The function is initiated by pressing the handles together (at 1 and 2). The handles, being pivoted at their ends, can rotate, and independently instantiate MP-LEVER-TYPE2. When an object is placed between and in contact with each handle they work together to provide the leverage needed to exceed the object's breaking strength. The nutcracker can also instantiate the pliers clamping function, the hammer pounding function, and the bottle opener probing function.

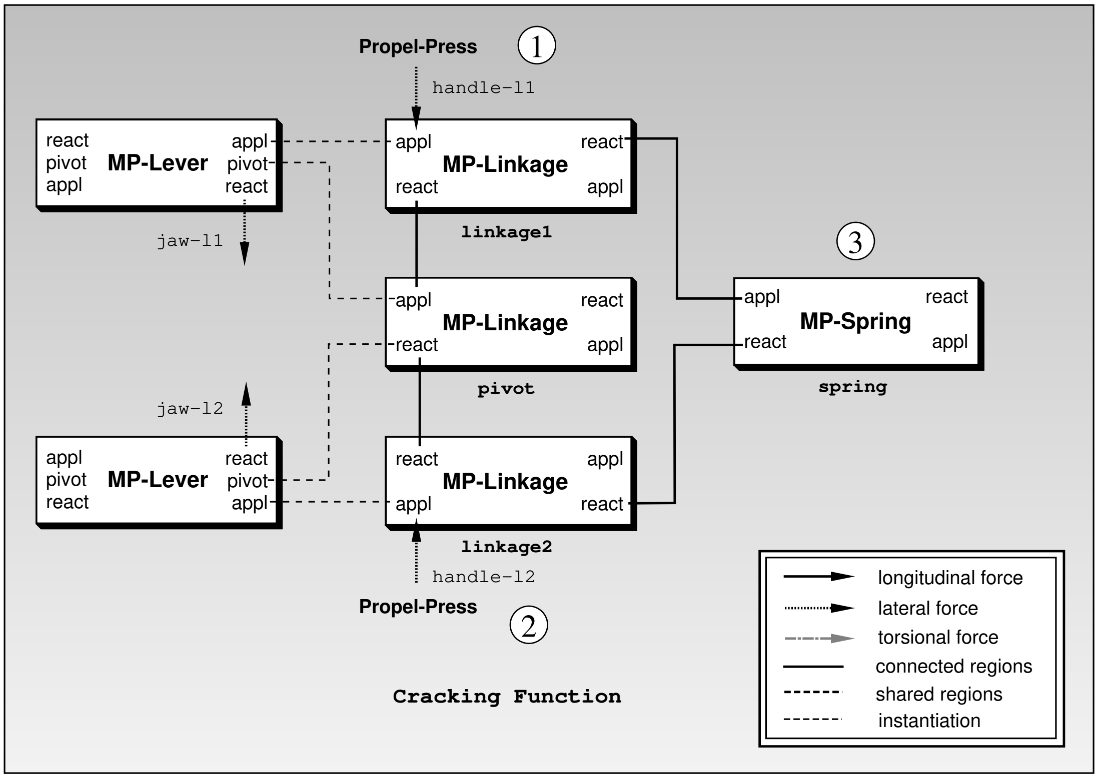


Figure 4.40 MP-Diagram for nutcracker cracking function.

In Fig. 4.39, the two handles are initially represented as linkage-objects. However, when an object is place in contact with the jaw regions, the linkage-object/pivot/object combination instantiates an MP-LEVER-TYPE2. To instantiate a hammering function, or the bottle-opener probing function, the nutcracker handles are used as linkage-objects directly.

Clothes Pin

A clothes pin holds objects together without manual control. The device is comprised of three components: two lever-objects and a spring-object. The spring is employed to keep the levers together, as well as to keep the jaws closed, as shown in the idealized clothes pin and static device diagram depicted in Fig. 4.41.

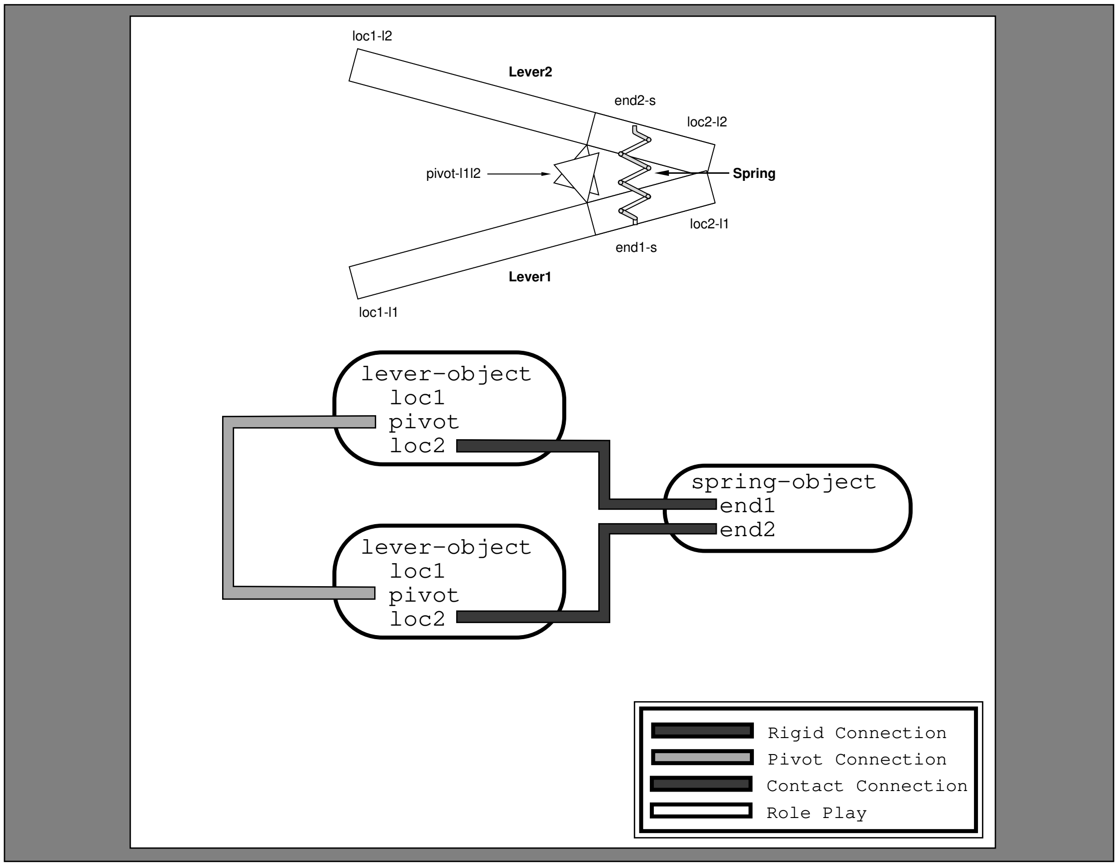


Figure 4.41 Clothes pin object primitives and static device diagram.

The static representation diagram for the clothes pin is almost identical to that of the nutcracker in Fig. 4.39, but their functions are dissimilar. In the clothes pin, the device is normally closed and the spring keeps it that way, whereas in the nutcracker the spring keeps the handles apart. In both devices the spring returns the device to its initial state, unlike scissors and pliers. In the clothes pin, the spring is part of the pivot and the lever object LOC2 regions are on the opposite side from the LOC1 regions. The effect is that LOC2 and LOC1 move in opposite directions, so having the spring in contact with this region provides opening resistance. The clothes pin clasping and probing functions are shown as MP-Diagrams in Fig. 4.41. In the clasping function, the handles are pressed together, which causes a lateral force (at 1) in both handles.

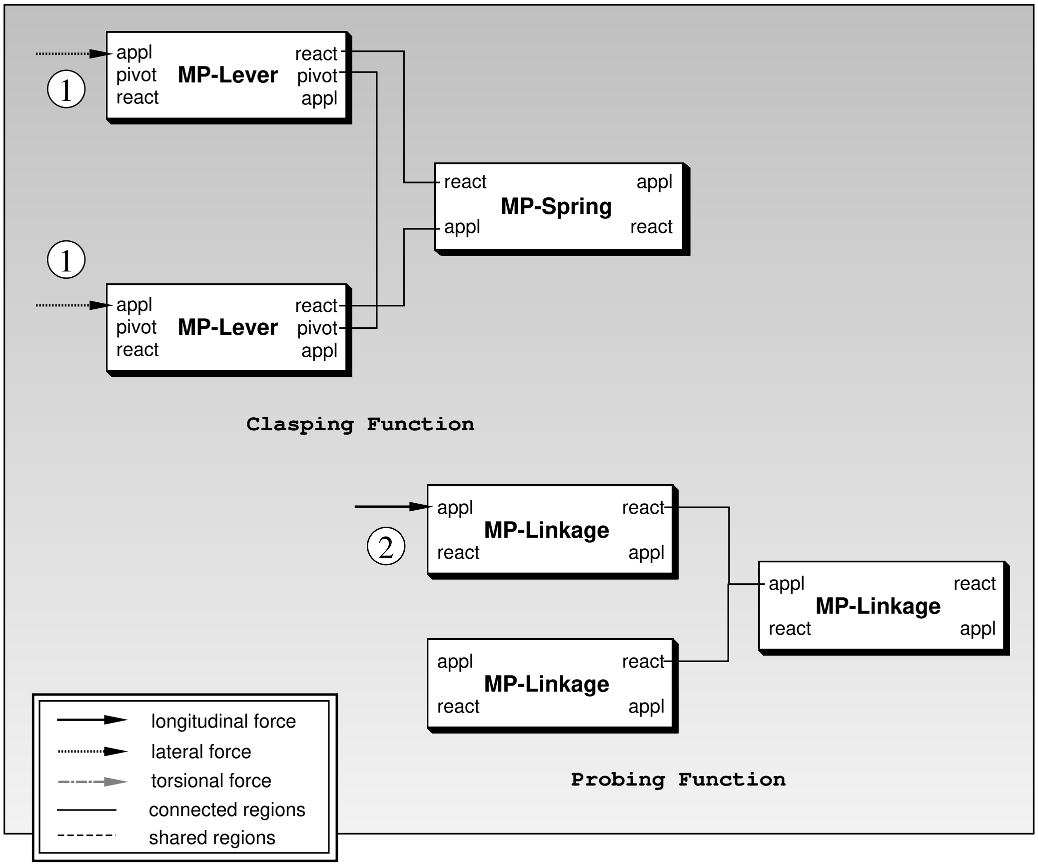


Figure 4.42 MP-Diagram for clothes pin function.

The lever pivot roles are shared, and the spring is connected to the lever react role regions. The probing function is initiated by a longitudinal force (2) and is represented with MP-LINKAGE.

4.4.3 Multiple Compound Devices

Whereas a pair of scissors or a clothes pin have two components which can move independently, a multiple-compound device has more than one independently moving component. Although the statics and intra-object dynamics are fairly straightforward, the inter-object dynamics begins to become complex when fully elaborated. One such device is the fingernail clipper, whose MP representation is presented here. Two additional devices, a press and a crank can opener are presented in Appendix A.3.

Fingernail Clipper

A fingernail clipper clips nails by forcing two blades together. The force necessary to clip is achieved with a lever, and the device is returned to its initial state with a spring. An idealized fingernail clipper and its associated static device diagram are shown in Fig. 4.43. The only components which are rigidly connected are the linkages. The handle can move independently of the blades, and the blades can move with respect to each other. The handle is pivoted to the pin (at 1 in Fig. 4.43). The spring is illustrated as playing the role of `end1` in both linkage objects, rather than being connected to them, to emphasize that the linkage objects are being used as springs.

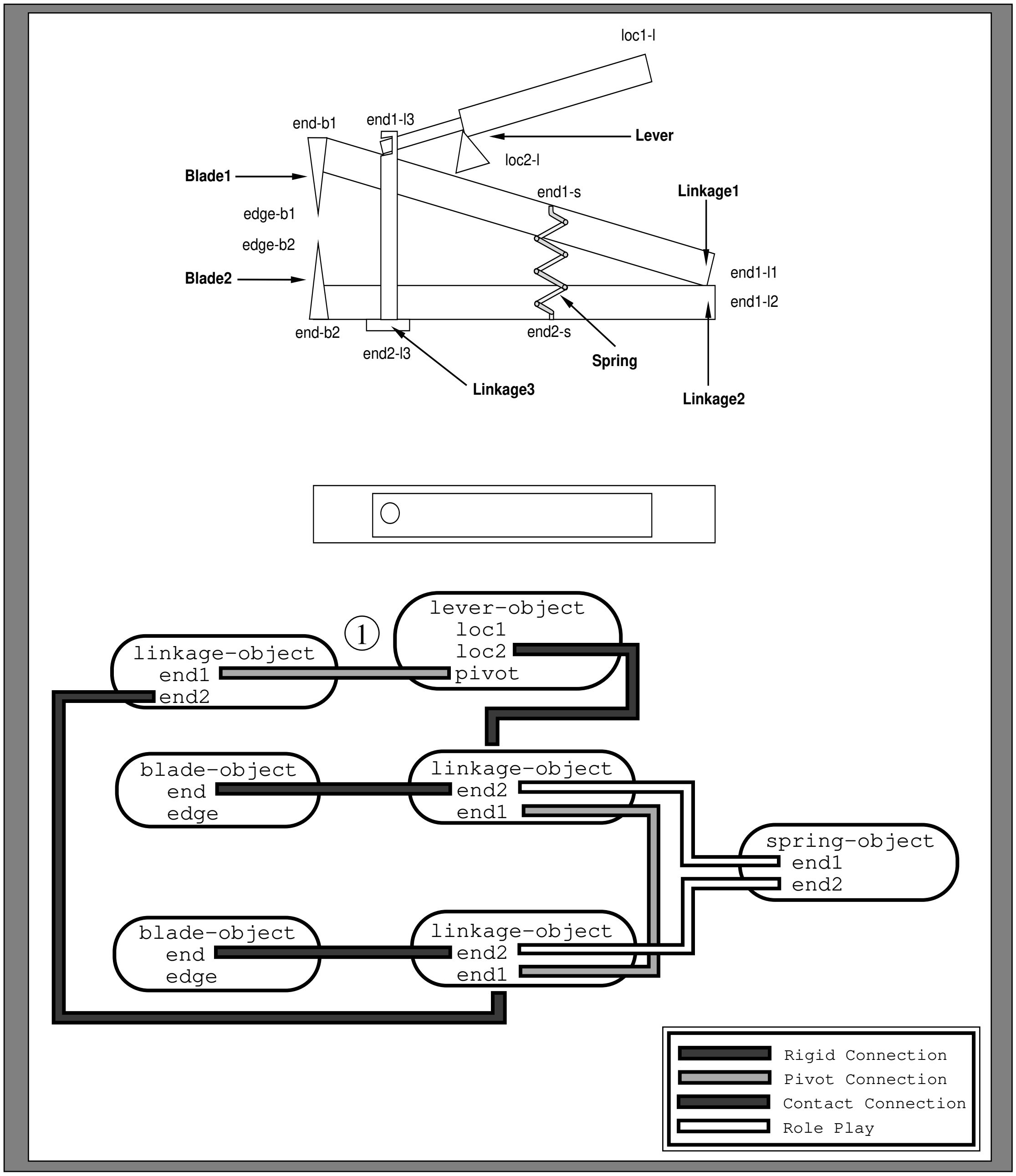


Figure 4.43 Fingernail clipper object primitives and static device diagram.

The fingernail clipper function is represented as an MP-Diagram in Fig. 4.44.

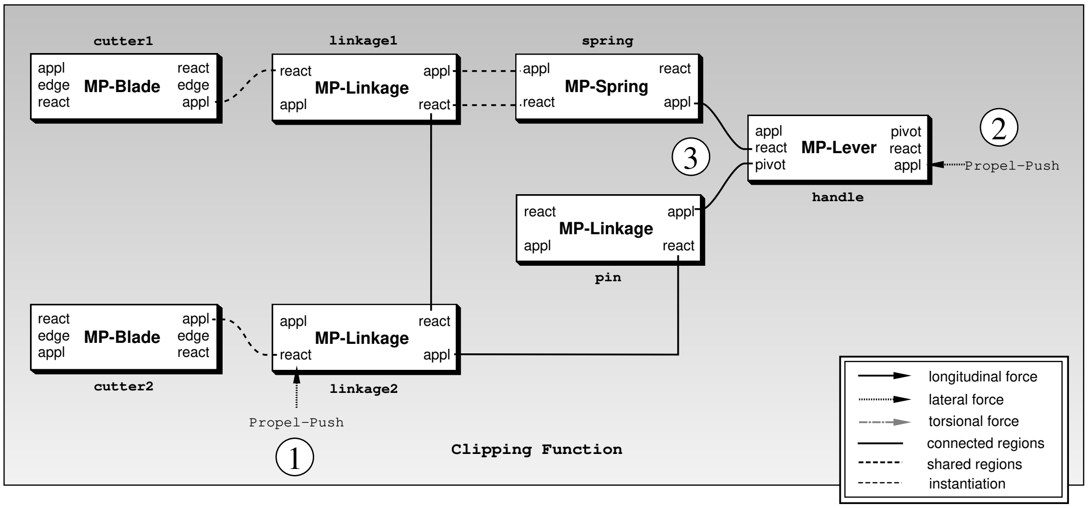


Figure 4.44 MP-Diagram for fingernail clipper function.

The clipping function is initiated by holding the lower piece (at 1) and pressing the handle (at 2). The handle is pivoted (see 1 in Fig. 4.43) to the pin and is also in contact with the upper piece (3). This is represented with MP-LEVER, which magnifies the applied force at the upper piece, while the pin acts as a fulcrum. As the handle rotates, the magnified force increases, and the blades move closer together. The shape of the blades enables further magnification with MP-BLADE. The resulting change in finger nail size terminates the function. The device is reset by releasing the handle (not shown in this figure), whereupon the blades open and the handle returns to its original position.

4.4.4 Complex Devices

Complex devices involve objects which may or may not be permanent components of the device, have complex motions, or have large numbers of components. One such example is a toy gun, which has a projectile which is only connected to the device when being armed and preparing to fire. Another complex device is a mousetrap, since the hook and baitplate motion are both confined to the limits imposed by their connectors, but free to move in every other respect. In this section, the toy gun's MP block representation is presented, while that of the mouse trap, a corkscrew, and an egg beater are presented in Appendix A.4.

Toy Gun

The idealized toy gun and static device diagram are illustrated in Fig. 4.45.

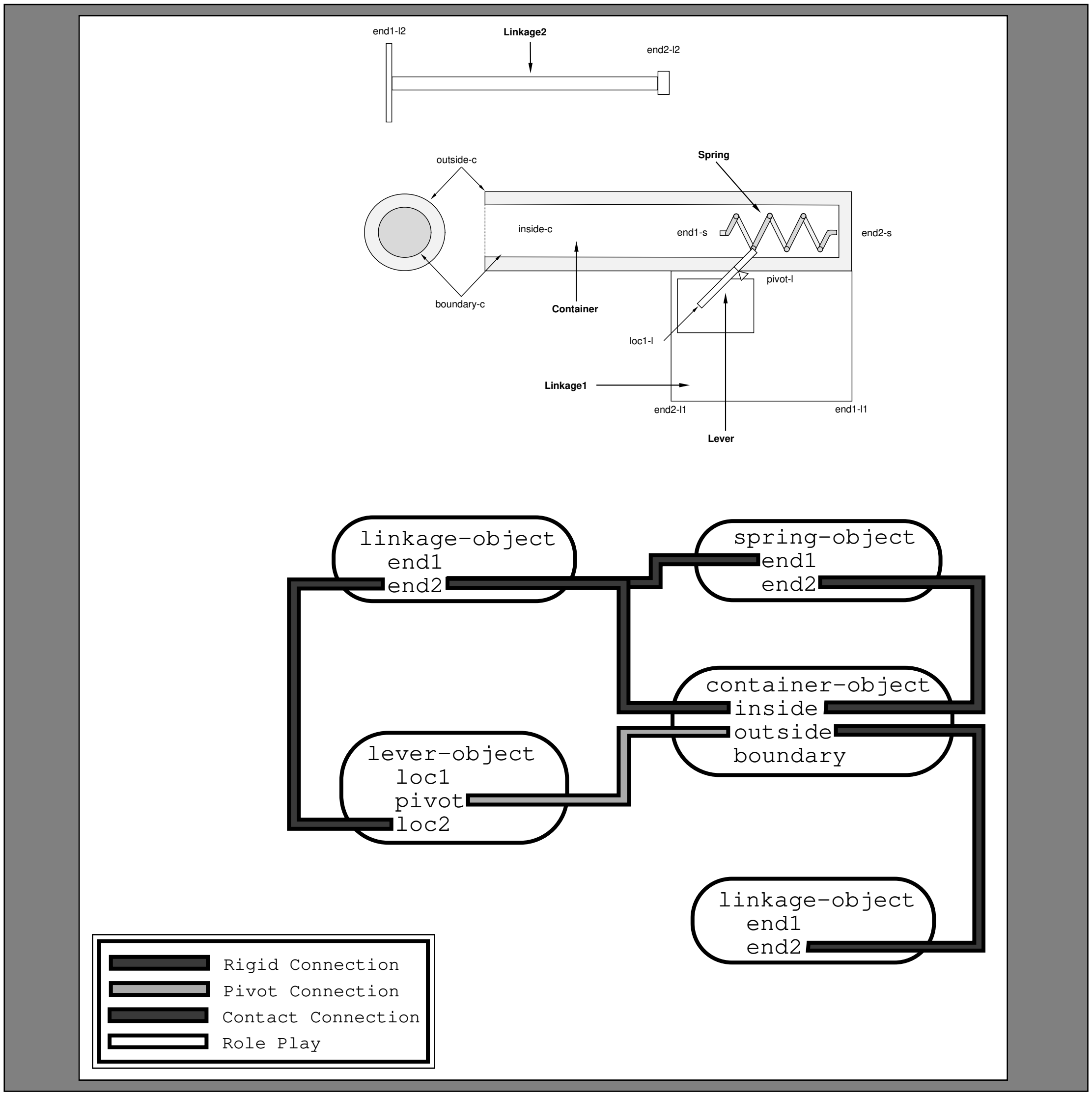


Figure 4.45 Toy gun object primitives and static device diagram.

The arming and firing functions of the toy gun are illustrated in Fig. 4.46. In both functions, the handle is held and does nothing more than transmit force to the barrel and other components. It is thus represented with MP-LINKAGE. In the arming function, the dart is pushed into the barrel. The dart is a separate object which comes into contact with the device during both functions. The force produced by pushing the dart initiates the arming function.

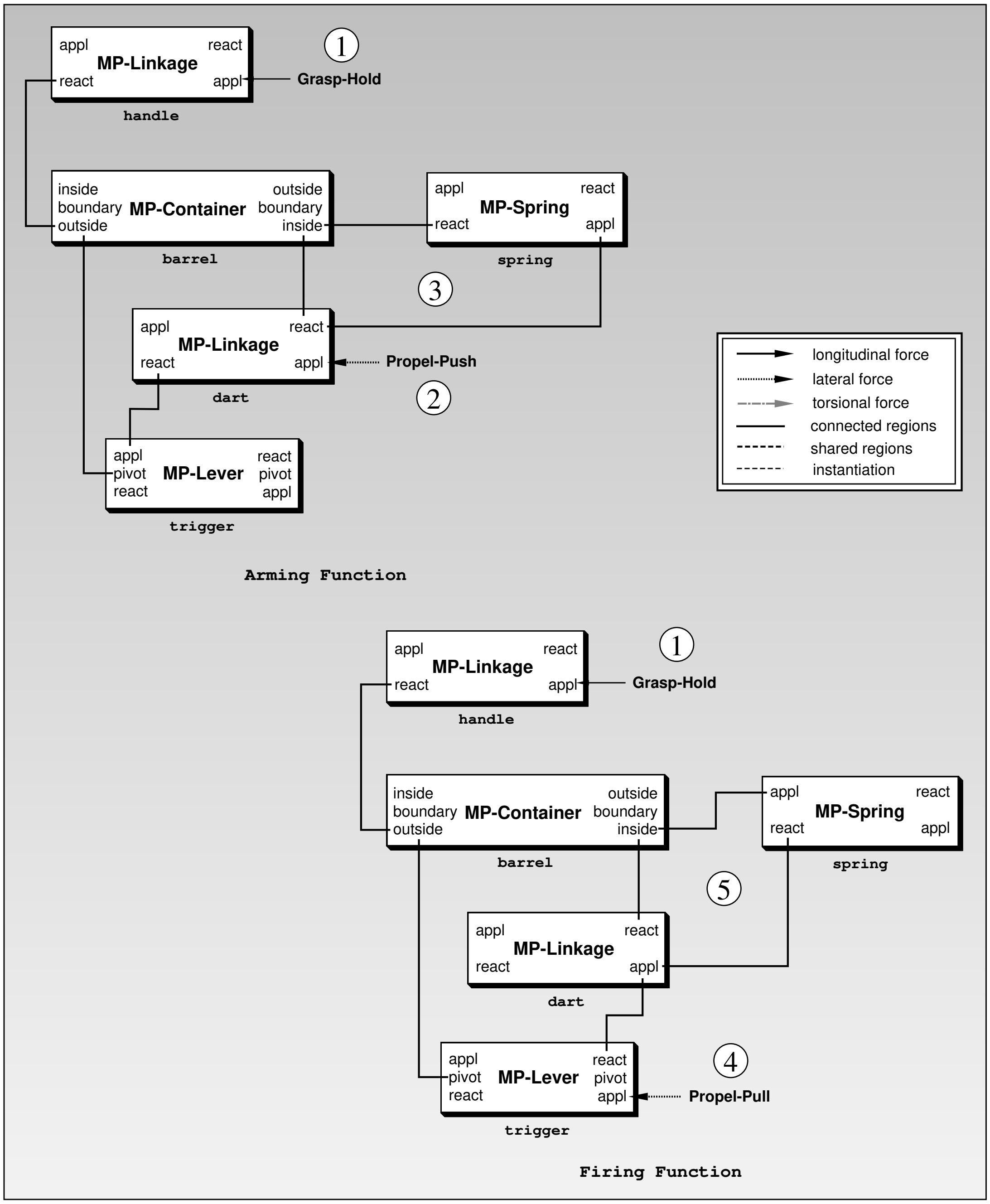
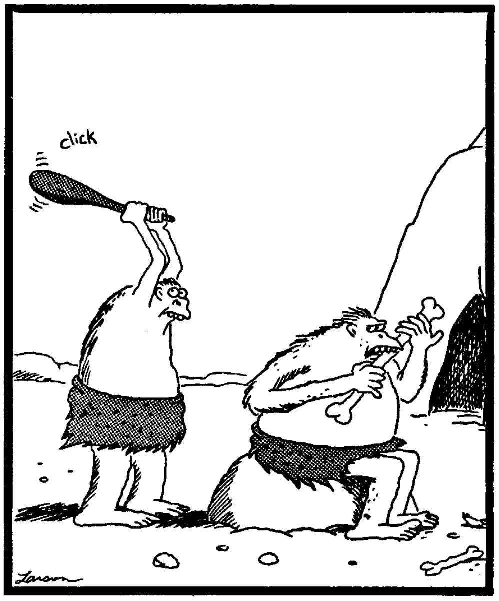


Figure 4.46 MP-Diagrams for toy gun functions.

The role of the spring and how MP-SPRING is instantiated differs in the two functions. In the arming function, the dart compresses the spring until the dart catches the trigger (3). In the firing function, the trigger is pulled (4), disengaging the dart, and the spring is released (5).



At a critical moment, Zak's club jams.

FAR SIDE copyright FARWORKS, INC. Reprinted with permission of UNIVERSAL PRESS SYNDICATE. All rights reserved.

○

○

○

○

Chapter 5

Device Pragmatics

Devices are used to accomplish tasks. *Device pragmatics* represents the use of devices with respect to what the user's intentions and actions are and with respect to the device's function and behavior. Two topics are addressed in this chapter: (1) device use plans, and (2) device related subgoals.

The first section of this chapter introduces the notion of a *device-use* plan, which represents how a device is applied once the decision of what needs to be done has been made. The actions and plans associated with device-use enable a particular device function.

The second section of this chapter introduces the notion of device-related goals, which represent state changes a problem solver must effect that require device behavior and function. Device-use plans are comprised of subgoals, some of which are device-related goals, so these goals help to identify the need for device application, and to represent the causal relationship between the plans in which devices are used and the functions which they are used to produce.

The third section of this chapter presents pragmatic (device-use) representations for four device classes: simple devices, simple compound devices, multiple compound devices, and complex devices.

5.1 Device-Use Plans

A plan in which a device function is applied is called a *device-use* plan. The device-use plan is an adaptation of Schank and Abelson's USE(object) plan schema [Schank and Abelson, 1977]. Device-use plans have five roles: actor, inst, obj, from and to. The *actor* role is filled by the agent who performs the actions.The *inst* role is filled by the instrument used to produce the state. The *obj* role is filled by the object(s) to which the instrument is applied. The *from* and *to* roles are filled by the initial and final states described by the goal. For example, the use of a nutcracker to crack pecans is represented in Fig. 5.1.

```
(plan crack-open
  actor   EDISON
  inst    NUTCRACKER
  obj     PECAN
  from    CLOSED-PECAN
  to      OPEN-PECAN)
```

Figure 5.1 Slot-filler instantiation of device-use plan for nutcracking.

In device-use plans, a goal is achieved by actions, scripts, and plans which result in physical control of the instrument and object. For example, the use of a nutcracker to crack pecans open requires the user to grasp the nutcracker handles (they start out spread apart), insert the pecan between the nutcracker jaws, and press the handles together, as shown in Fig. 5.2.

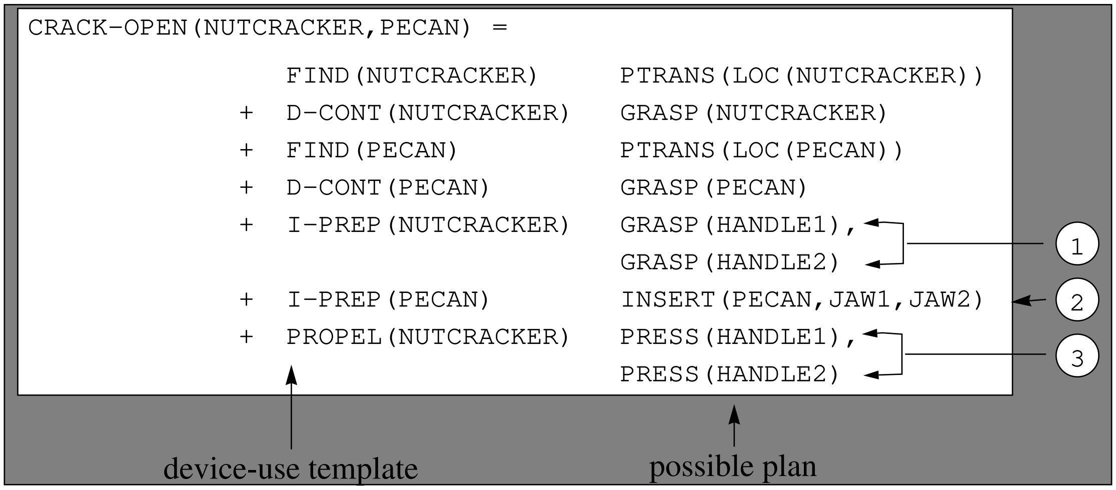


Figure 5.2 Slot-filler instantiation of device-use plan for nutcracking.

The CRACK-OPEN plan of Fig. 5.1, depicted in Fig. 5.2, is now similar to two interacting USE(object) plans, one for the instrument (NUTCRACKER), and the other for the object (PECAN). In Fig. 5.2, the FIND subgoal for both objects has been simplified to going to where the object is located (i.e., PTRANSing). The D-CONT (physical control) subgoal for each object is satisfied with a GRASPing action. The nutcracker preparation goal ensures that the handles are spread apart (i.e., to get the device OPEN), so that the pecan will fit between the jaws. A nutcracker has a spring between the handles, which is uncompressed when the device is open, so the user need not OPEN the nutcracker under normal use. However, the user must grasp the handles (shown at 1 in the figure). These actions provide a means to apply force to the handles and thereby enable the nutcracker CLOSEing plan. The preparation of the pecan involves inserting it between the two nutcracker jaws (2). INSERT is represented with an instrumental script which identifies the location and orientation of the components and results in interference between them. Thus, INSERT(PECAN,JAW1,-JAW2) results in the necessary restraint state on the pecan which enables BPP-DEFORM-CRACK. Once these actions and plans have been successfully completed, the CRACK-OPEN plan is enabled, and pressing the levers together (3) initiates the nutcracker CRACK-APART function (MP-LEVER-TYPE2), resulting in the OPEN pecan state.

5.2 Device Use Goals

The representation for the CRACK-OPEN device-use plan (Fig. 5.2) assumes that there is an explicit causal relationship between the I-PREP(NUTCRACKER) goal of the user and the associated GRASP actions, and between the I-PREP(PECAN) goal and the INSERT(-PECAN,JAW1,JAW2) script. Part of nutcracker use is that it must be open so that the pecan can be inserted between its jaws. In FONM these requirements specialize the preparation goals in terms of identifying the means to resolve them. In this section, the delta goals and instrumental goals which are associated with device-use plans are specialized to relate them with particular device behaviors and functions.

A device behavior or function is applied as a step in a plan to achieve a goal. Likewise, a goal may identify appropriate actions, scripts, and plans which produce specific behaviors and functions. The failure to achieve a goal with a known plan may motivate the user to try another approach rather than to abandon the original goal. Goals which represent changes in behavioral state are associated with blocked delta goals (with a "motivates" link, see Eqn. (1-11) on page 11). For example, suppose the user has an active goal to eat a pickle and that the pickle is in a closed pickle jar. The use(pickle) plan has a subgoal to gain physical control (D-CONT) of the pickle. The subgoal is blocked because the normal method for achieving control is to GRASP the pickle. One alternative is to abandon the goal and decide to eat something else. Another alternative is to open the jar. The latter choice represents a change in the jar's restraint state. In FONM, this change can only be effected by a RESTRAIN process, so the subgoal to change restraint states is directly related to plans which enable the RESTRAIN process. Each behavioral process primitive is similarly associated with a delta goal.

Now suppose that the method applied to open the jar is to GRASP the cap and then TWIST it, but that this plan fails because the lid is on too tight. Again, one option is to abandon the goal. Another is to try an approach which utilizes a device, such as running the jar under hot water, banging on the lid with an object, or finding some means to apply more torsional force to the lid. With these plans, the application of a device function may produce the desired state change indirectly, in which case its use is instrumental to subgoal achievement. In FONM, each machine primitive is associated with an instrumental goal.

5.2.1 Behavioral Delta Goals

In FONM, changes in behavioral state are represented using behavioral primitives, so the need to produce behavioral state changes implies a delta-type goal. There are five delta goal types associated with object behavior: D-POSITION, D-RESTRAINT, D-AFORCE, D-IFORCE, and D-SIZE. The pecan restraint state change needed for the pecan is represented with a delta-restraint (D-RESTRAINT) goal. Each of these delta goals describes necessary changes of a specific behavioral state and identifies the kinds of device-use and instrumental plans which can be applied to produce those state changes.

D-POSITION: A goal for producing object positional changes. Plans which achieve D-POSITION instantiate a MOTION process primitive.

D-RESTRAINT**:** A goal for changing physical restraints on an object. Plans which achieve D-RESTRAINT instantiate a RESTRAIN process primitive. D-RESTRAINT is motivated when a problem solver needs to open a bottle, for example, or to connect two engine components.

D-AFORCE: A goal for producing applied forces on an object when PROPEL is insufficient. Plans which achieve D-AFORCE instantiate a TRANSFORM process primitive. D-AFORCE is motivated when a problem solver needs to dislodge a frisbee from a tree, because an axial force applied at some distance may be sufficient.

D-IFORCE: A goal for temporarily changing object internal forces. Plans which achieve D-IFORCE instantiate a STORE process primitive. D-IFORCE is motivated when a person's watch runs down, or a child's windup toy ceases to run. In both cases, energy stored in a spring is dissipated as the device is used, and the user is motivated to retension the device to achieve a higher goal.

D-SIZE: A goal for changing size or shape of an object. Plans which achieve D-SIZE instantiate a DEFORM process primitive. For instance, D-SIZE is motivated when a problem solver needs to remove metal from objects which do not fit together, by shaving or grinding.

To illustrate the use of behavioral delta goals, consider the situation shown in Fig. 5.3.

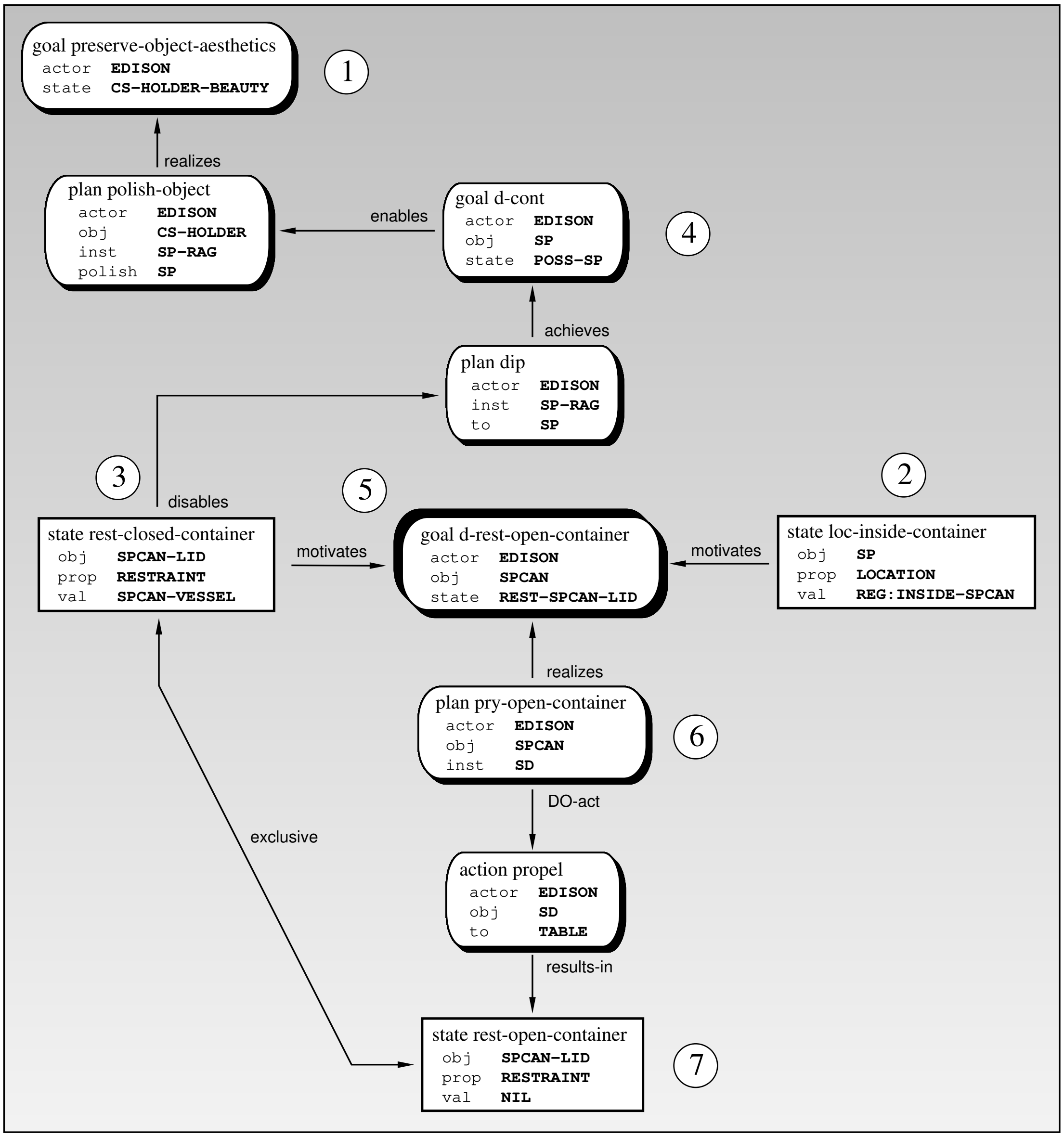
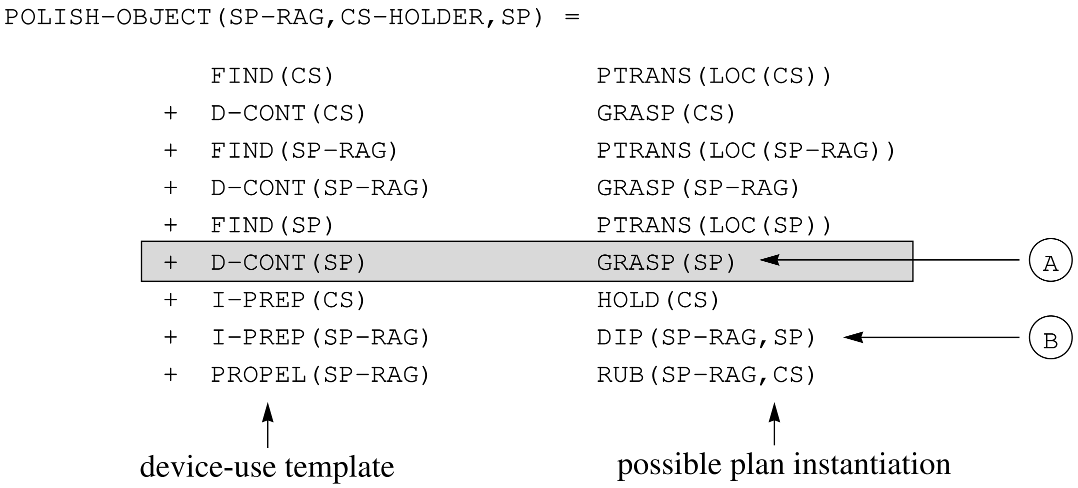


Figure 5.3 The application of behavioral delta goals for resolving goal conflict. The states at items 3 and 7 are named to coincide with the observable state, which is the same state as that mentioned in the d-rest goal at 5.

Fig. 5.3 depicts the goal/plan representation for a situation in which a problem solver wishes to preserve the beauty of a pair of candlesticks (CS-BEAUTY, at 1). One plan associated with such a goal is to polish the candlesticks. The preparation associated with polishing the candlesticks requires the possession of a rag (SP-RAG) and placing some silver polish (SP) on the rag, as illustrated in Fig. 5.4. These subgoals can be represented with the delta goals,



```
POLISH-OBJECT(SP-RAG,CS-HOLDER,SP) =

        FIND(CS)                PTRANS(LOC(CS))
      + D-CONT(CS)              GRASP(CS)
      + FIND(SP-RAG)            PTRANS(LOC(SP-RAG))
      + D-CONT(SP-RAG)          GRASP(SP-RAG)
      + FIND(SP)                PTRANS(LOC(SP))
      + D-CONT(SP)              GRASP(SP)
      + I-PREP(CS)              HOLD(CS)
      + I-PREP(SP-RAG)          DIP(SP-RAG,SP)
      + PROPEL(SP-RAG)          RUB(SP-RAG,CS)
```

Figure 5.4 Device-use plan for polishing an object.

D-KNOW(LOC(SP)), D-PROX(LOC(SP)), and D-CONT(SP), with the action GRASP(SP-RAG), and with the plan DIP(SP-RAG,SP). However, prior to dipping the rag into the polish, the can which contains the polish must be open. Two states are given in the situation illustrated in Fig. 5.3: that the silver polish is located in the silver polish can (at 2), and that the silver polish can is closed (3). The D-CONT(SP) goal (at 4 in Fig. 5.3 and at A in Fig. 5.4) is thwarted because the closed silver polish can disables the DIP(SP-RAG,-SP) plan (B). This motivates a behaviorally related delta goal to open the can by removing the lid (5), D-RESTRAINT(open-container). Lid removal is represented as a plan resulting in restraint removal by enabling a RESTRAIN process. A plan can be used to represent the method for can opening, such as using a screwdriver (SD) to pry the lid open (PRY-OPEN-CONTAINER(SD,SPCAN), at 6), but the motivation to open the can is a D-RESTRAINT goal.

The D-REST goal which is motivated by the inability to directly gain control over a substance in a closed container is associated with many possible plans for opening a container, depending on the container and the substance. Some alternative plans are detailed in Tab. 5.1 The identification of which plan is appropriate for the candlestick polishing sit-

D-RESTRAINT and Container Opening Plans	
Container Description	Open-Container Plan
threaded lid	twist-lid-off
friction-insert lid	pry-lid-off
friction cap	twist-and-pull-cap-off
friction plug	pull-plug-out
crimped cap	pry-cap-off

①

Table 5.1: Container opening plans associated with D-RESTRAINT(remove-restraint).

D-RESTRAINT and Container Opening Plans	
Container Description	Open-Container Plan
release catch	release-catch + lift lid
lip-and-lid, fluid	punch-hole-in-lid
lip-and-lid, non-fluid	cut-lid-off

Table 5.1: Container opening plans associated with D-RESTRAINT(remove-restraint).

uation (at 1) requires a recognition of the container device and its static representation, specifically by identifying the friction interference between the lid and the can, and the insert orientation between the lid and the can.

5.2.2 Mechanical Instrumental Goals

A class of goals for generating mechanical advantage (I-MECH) is satisfied by devices which have functions whose main conceptualizations involve BPP-TRANSFORM (i.e., that can produce the necessary applied force). I-MECH is one of five *mechanical instrumental* goals in FONM which relate a desired function to the devices which can instantiate it: I-REACH, I-MECH, I-CNTN, I-RESIST, and I-WEDGE. Each instrumental goal type is associated with a particular machine primitive and its object type:

I-REACH: A goal for extending control beyond a component boundary in situations requiring mechanical assistance. The object of I-REACH is a linkage-object, which can instantiate MP-LINKAGE. For example, I-REACH is motivated when professor needs to point to something on a blackboard, in which case a ruler might be useful.

I-MECH: A goal for gaining purchase through mechanical force advantage. The object of I-MECH is a lever-object, which can instantiate MP-LEVER. For example, I-MECH is motivated when a problem solver needs to lift a box which is heavier than he can lift. In this case a crow-bar, or block and tackle, will suffice.

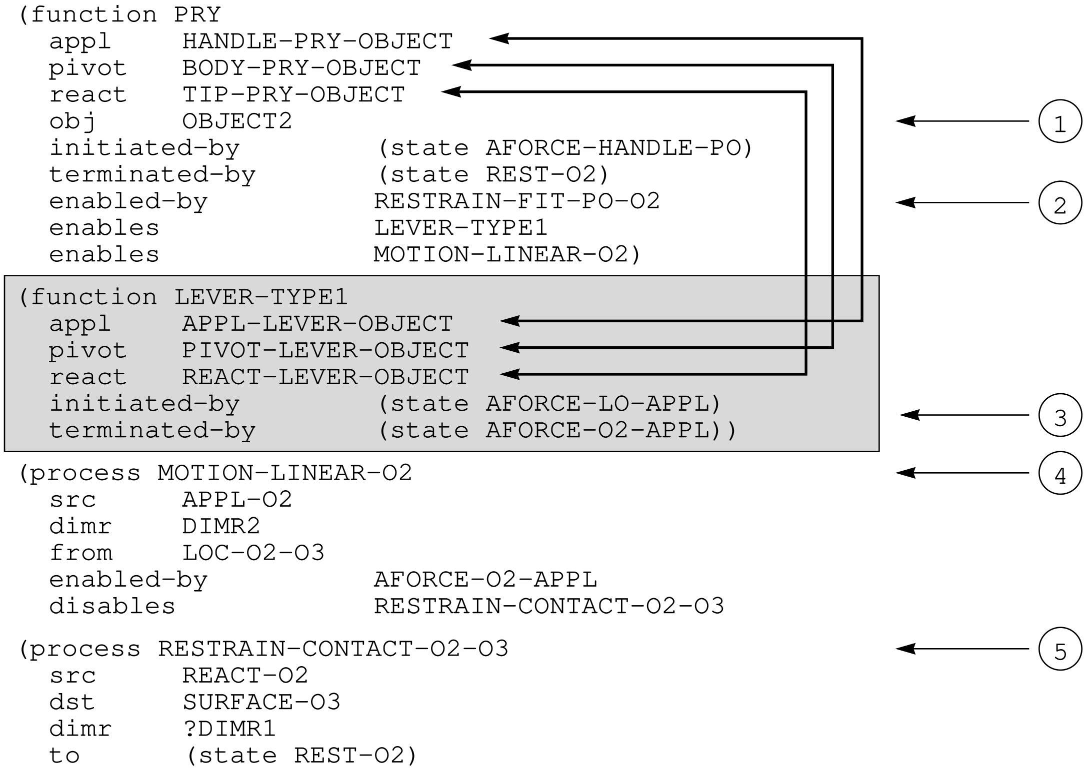
I-CONTN: A goal for gaining mechanical control of an object's range of movement. The object of I-CONTN is a container-object, which can instantiate MP-CONTAINER. For example, I-CONTN is motivated when a problem solver needs to keep tools from rolling down the driveway, or to keep a dog in the yard. In the first instance, the tools might be placed in a hubcap, and in the latter, the dog might be placed on a tether. Each device instantiates MP-CONTAINER.

I-RESIST: A goal for producing passive force that require mechanical assistance. The object of I-RESIST is a spring-object, which can instantiate MP-SPRING. For example, I-RESIST is motivated when a problem solver needs object movement in two directions but can only provide force in one, such as automatic door closing, which is satisfied by objects which instantiate MP-SPRING.

I-WEDGE: A goal for producing changes in object position that require mechanical assistance. The object of I-WEDGE is a plane-object, which can instantiate MP-PLANE. For example, I-WEDGE is motivated when a problem solver wants to change a lightbulb or get a refrigerator onto a moving truck. In the first case, a ladder can be used, while in the second, a ramp can be used, both of which instantiate MP-PLANE.

Five mechanical instrumental goals identify all machine primitives, because they identify the kind of states associated with all device classes. For example, MP-GEAR is used to obtain a specific kind of purchase, one in which position control is required, but it is still used, mechanically, to obtain purchase, so the appropriate instrumental goal is I-MECH.

Mechanical instrumental goals indirectly relate a desired state change to devices whose functions can produce the state change. The candlestick example illustrates the use of mechanical instrumental goals. The delta-restraint goal, which is motivated by a closed silver polish can, cannot be achieved without the use of a mechanical device to produce the necessary force. The PRY-OPEN plan can be used to achieve the D-RESTRAINT goal, but requires a device capable of prying. Consider the device function, PRY, shown in Fig. 5.5, and how it applies to the PRY-OPEN plan.



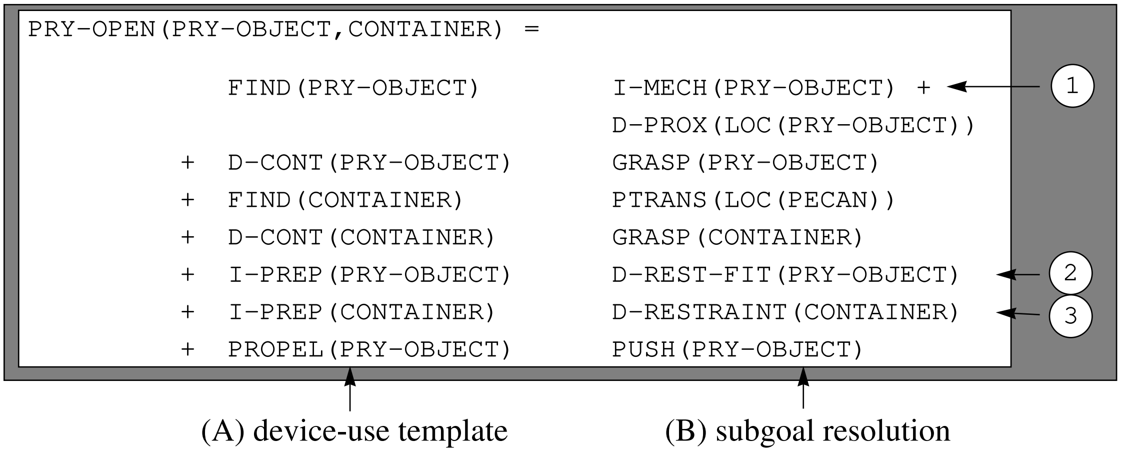
```
(function PRY
  appl     HANDLE-PRY-OBJECT
  pivot    BODY-PRY-OBJECT
  react    TIP-PRY-OBJECT
  obj      OBJECT2
  initiated-by        (state AFORCE-HANDLE-PO)
  terminated-by       (state REST-O2)
  enabled-by          RESTRAIN-FIT-PO-O2
  enables             LEVER-TYPE1
  enables             MOTION-LINEAR-O2)
(function LEVER-TYPE1
  appl     APPL-LEVER-OBJECT
  pivot    PIVOT-LEVER-OBJECT
  react    REACT-LEVER-OBJECT
  initiated-by        (state AFORCE-LO-APPL)
  terminated-by       (state AFORCE-O2-APPL))
(process MOTION-LINEAR-O2
  src      APPL-O2
  dimr     DIMR2
  from     LOC-O2-O3
  enabled-by          AFORCE-O2-APPL
  disables            RESTRAIN-CONTACT-O2-O3

(process RESTRAIN-CONTACT-O2-O3
  src      REACT-O2
  dst      SURFACE-O3
  dimr     ?DIMR1
  to       (state REST-O2)
```

Figure 5.5 Slot-filler instantiation of the PRY function.

Fig. 5.5 illustrates the PRY function and how an object which can be used for prying objects apart enables the function MP-LEVER-TYPE1 (or MP-LEVER-TYPE2, but TYPE1 is represented in the highlighted area of this figure). The function roles instantiated by the pry-object regions also instantiate the roles of the lever-object, as noted by the double-arrowed lines in the figure. The pry-object fits under the object to be pried (RESTRAIN-FIT-PO-O2, at 2). The two contact locations (in this example, different regions/components of OBJECT2) provide a fulcrum and react location for leverage. The direction of applied force determines which MP-LEVER type is instantiated. Motion for the object (OBJECT2, at 1) being pried is enabled by the magnified force produced by the instantiated MP-LEVER.

The interaction between the LEVER machine primitive and the PRY function can be seen in the function bounding states (at 3). The initiating state for PRY is identical to that of MP-LEVER-TYPE1, but the terminating states differ. The terminating state of PRY is the change in restraint on the object being pried, whereas the terminating state of MP-LEVER-TYPE1 is the magnified force. The representations for the MOTION-LINEAR process, and for one of the contact relationships enabling the RESTRAIN-FIT process, are shown in the figure, at (4) and (5), respectively.

To apply the function PRY in a device-use plan, the roles associated with the function and the function preconditions must be instantiated by the objects in the plan. For example, consider the PRY-OPEN plan illustrated in Fig. 5.6.

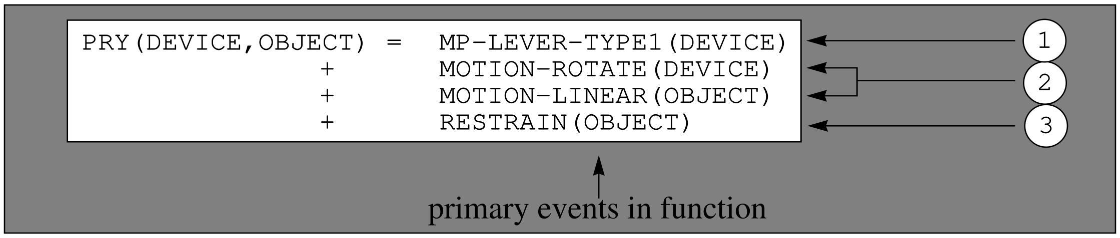


```
PRY-OPEN(PRY-OBJECT,CONTAINER) =

            FIND(PRY-OBJECT)        I-MECH(PRY-OBJECT) +
                                    D-PROX(LOC(PRY-OBJECT))
         +  D-CONT(PRY-OBJECT)      GRASP(PRY-OBJECT)
         +  FIND(CONTAINER)         PTRANS(LOC(PECAN))
         +  D-CONT(CONTAINER)       GRASP(CONTAINER)
         +  I-PREP(PRY-OBJECT)      D-REST-FIT(PRY-OBJECT)
         +  I-PREP(CONTAINER)       D-RESTRAINT(CONTAINER)
         +  PROPEL(PRY-OBJECT)      PUSH(PRY-OBJECT)
```

Figure 5.6 Resolution of subgoals in the prying device-use plan.

Fig. 5.6 illustrates a *resolution* (i.e., an elaboration or motivation) of the subgoals in the PRY-OPEN device-use plan template (at A) into mechanical instrumental and behavioral delta goals, actions, scripts, or plans (at B). In Fig. 5.6, finding a pry-object amounts to locating an object which can instantiate MP-LEVER-TYPE1 or MP-LEVER-TYPE2. The I-MECH goal (at 1) is thus an instrumental enablement to opening the silver polish can by prying. The specialization of the FIND subgoal with I-MECH concurs with the Schank and Abelson use of FIND (Eqn. (8-2), page 308), since D-KNOW(loc(pry-object)) implies knowledge that an object can be used as a pry-object. The mechanical instrumental goals specialize this knowledge to the type of device function needed. The delta-restraint goal D-REST-FIT (at 2) represents the restraint state necessary to achieve in order to instantiate the behavioral precondition for PRY (at 2 in Fig. 5.5). The goal D-RESTRAINT (3) represents the requirement that the container be restrained during the PRY-OPEN plan.

The manner in which the behavior associated with the PRY function is instantiated is illustrated in Fig. 5.7. I-MECH is achieved by the state (AFORCE-O2-APPL), which is



```
PRY(DEVICE,OBJECT) =   MP-LEVER-TYPE1(DEVICE)
               +       MOTION-ROTATE(DEVICE)
               +       MOTION-LINEAR(OBJECT)
               +       RESTRAIN(OBJECT)
```

Figure 5.7 Main conceptualizations of the function PRY.

caused by the function PRY through the instantiation of MP-LEVER-TYPE1. Since the device which instantiates this process is being sought, the goal is to instantiate MP-LEVER-TYPE1 (at 1 in Fig. 5.7). The device, and object motion which result from enabling MP-LEVER (at 2), also enable the RESTRAIN (i.e., a disconnection, or removal of restraint, which is still a change in restraint). The RESTRAIN is the main conceptualization of PRY, because it achieves D-RESTRAINT. The device which instantiates MP-LEVER is an instrumental participant in producing this result.

The interaction between the delta goal which motivates PRY-OPEN, the instrumental goal which is motivated to locate a pry-object, and the PRY function which is applied is shown instantiated for opening a silver polish can (SPCAN) in Fig. 5.8.

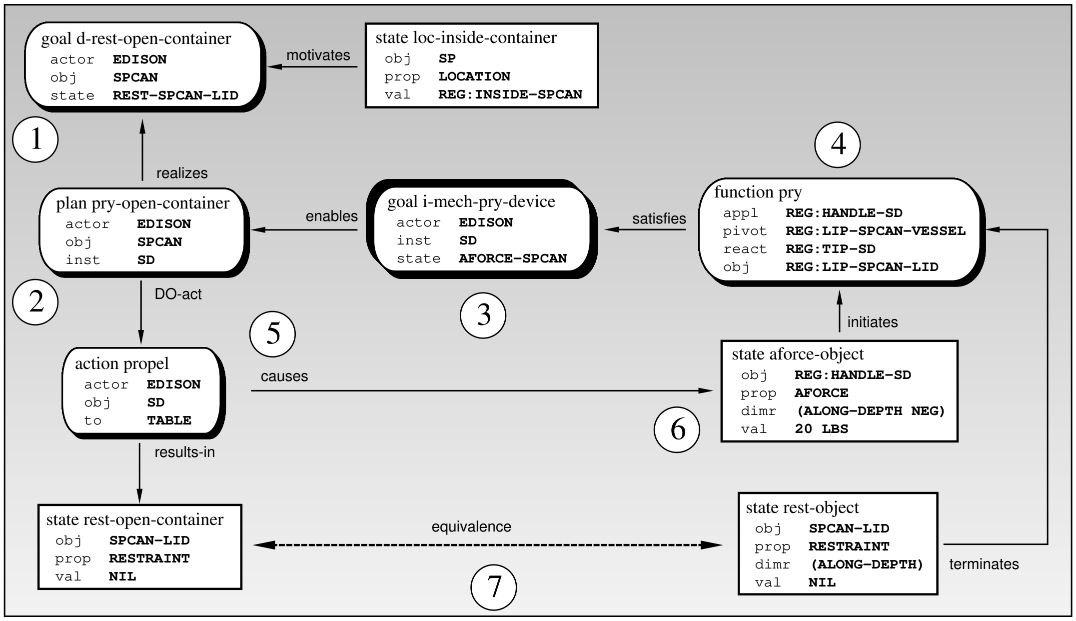


Figure 5.8 A mechanical instrumental goal, I-MECH, and its relationship to device-use plans and device functions.

The delta-restraint goal which was labeled (5) in Fig. 5.3 is labeled (1) in Fig. 5.8. Likewise, the PRY-OPEN-CONTAINER plan which was labeled (6) in Fig. 5.3 is labeled (2) in Fig. 5.8. The instrumental goal which assists in identifying the pry-object is highlighted in the figure (at 3). This goal is associated with devices which can instantiate the PRY function (4) and thus be used in the PRY-OPEN-CONTAINER plan. The object which instantiates the I-MECH goal in this example is a screwdriver (SD). The PROPEL-(PRY-OBJECT) action (5) taken in PRY-OPEN-CONTAINER causes the applied force (6) which initiates the PRY function. The restraint state which terminates the PRY function is identical to the OPEN state caused by the PRY-OPEN-CONTAINER plan (7).

Behavioral delta goals represent the causal interactions between device-use plans and device behavioral states which can be used to resolve plan subgoals. Mechanical instrumental goals represent the causal interactions between device-use plans, behavioral delta goals, and device functions. These goals provide a means to explicitly relate the machine primitives a device might instantiate with the types of goals and plans in which the device might be used.

A taxonomy of plans for resolving delta behavioral goals and instrumental mechanical goals has not been presented. Nevertheless, a class of eight general device-use plan types, called *machine plans*, have been identified which resolve behavioral delta goals and invoke machine primitives through instrumental mechanical goals. The causal relationships between machine plans, device-related goals, and devices are shown in Tab. 5.2. The primary process which is enabled by the machine plan is identified by the behavioral delta (BD) goal in column 2. The machine plan requires a device which satisfies the mechanical instrumental (MI) goal in column 3. The type of machine primitive which satisfies the MI goal is shown in column 4. As an example, consider the class of CONTROL-RANGE plans, highlighted in Tab. 5.2. These plans are invoked when object motion must be controlled, such as containment, confinement, or connection. The delta goal involves object restraint, and plans which resolve the restraint problem create or remove object restraints. The CONTROL-RANGE plan identifies the primary device through an I-CONTN goal (i.e., a device which instantiates MP-CONTAINER). The plan chosen is specific to the type of container involved. The OPEN plans shown in Tab. 5.1 show a broad diversity of methods for removing restraints based on the type of container. In Tab. 5.1, the OPEN plans are associated with known methods associated with the respective container type. For example, the OPEN plan for a container which has a threaded lid involves twisting the lid.

MACHINE PLANS			
Plan Type	**BD Goals**	**MI Goals**	**Machine Primitive**
MAKE-CONTACT	D-RESTRAINT	I-REACH	MP-LINKAGE
TRADEOFF-FORCE	D-AFORCE	I-MECH	MP-LEVER
CHANGE-FORCE	D-AFORC	I-MECH	MP-LEVER
CONTROL-RANGE	D-RESTRAINT	I-CONTN	MP-CONTAINER
RESIST-FORCE	D-IFORCE	I-RESIST	MP-SPRING
TRADEOFF-MOTION	D-MOTION	I-WEDGE	MP-PLANE
SEPARATE-PIECES	D-SIZE	I-WEDGE	MP-BLADE
SHAPE-OBJECT	D-SIZE	I-WEDGE	MP-PLANE

Table 5.2 Machine plans and their goal/function relationships.

5.3 Device Pragmatics Representations

A device's intended application can be represented with device use plans and device-related goals. In the following sections, the pragmatic representations of the design intended uses for several devices are presented and discussed. As in previous chapters, four device classes will be discussed:

(1) Simple devices
(2) Simple compound devices
(3) Multiple compound devices
(4) Complex devices

The device representations shown in the following sections are each specialized to the particular device and use, however, each device represented falls into the machine plan classification presented above. The full device-use template is presented for consistency, and the satisfaction of the instrumental mechanical goals for each device are discussed. Of particular note in these representations are the similarities and differences between device uses, the types of subplans which are involved, and the implications for problem solving which they identify. Device static figures are duplicated for clarity; however, no device dynamics are replicated in this section. For full device representations see the Appendix.

5.3.1 Simple Devices

Simple devices have no moving parts, and all components move as a single object. As a consequence, the use of simple devices is represented and identified by the type of force applied to the object, and the static device representation plays a large role in their application. The pragmatic representations for two simple devices are presented in this section:

(1) Bottle Opener
(2) Screwdriver

Pragmatic representations for four additional devices: (1) carving knife, (2) shovel, (3) spoon, and (4) claw hammer are presented in Appendix A.1.

Bottle Opener

An idealized bottle opener is depicted in Fig. 5.9. The bottle opener has two design-intended uses: (1) can opening, and (2) bottle opening, both of which are TRADEOFF-FORCE type plans in support of the associated container's CONTROL-RANGE (OPEN) plans. In both cases, the bottle-opener is a lever-object used for prying. As a simple device, using a bottle opener is primarily preparatory. For example, once one end of the bottle-opener is grasped, the intended use can be identified, because the use is specialized to the location on the bottle opener. Fig. 5.10 shows can puncturing as a device-use plan instantiated for

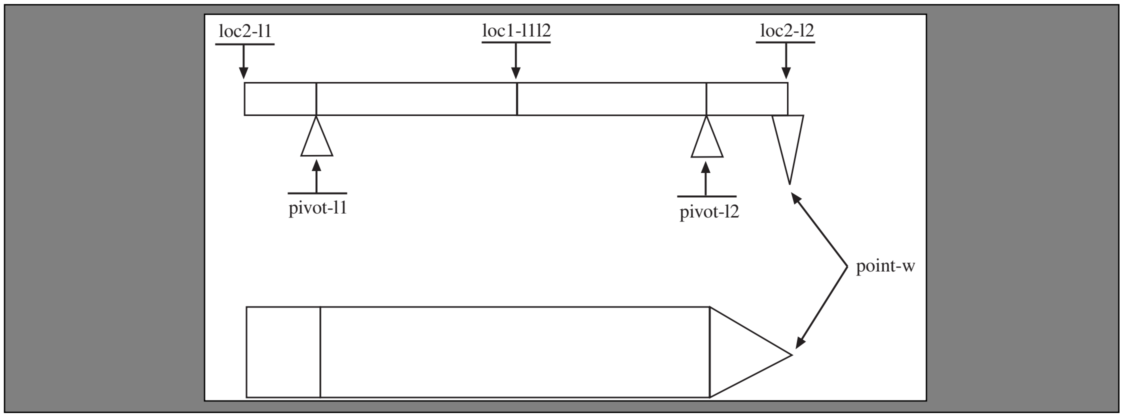


Figure 5.9 Idealized bottle opener.

a bottle opener and can.

```
PUNCTURE-LID(BOTTLE-OPENER,CAN) =
        FIND(BOTTLE-OPENER)           I-MECH(BOTTLE-OPENER) +
                                      I-WEDGE(BOTTLE-OPENER) +
                                      PTRANS(LOC(BOTTLE-OPENER))
    +   D-CONT(BOTTLE-OPENER)         GRASP(LOC2-L1)
    +   FIND(CAN)                     PTRANS(LOC(CAN))
    +   D-CONT(CAN)                   GRASP(CAN)
    +   I-PREP(CAN)                   D-RESTRAINT(CAN,SURFACE)
    +   I-PREP(BOTTLE-OPENER)         D-RESTRAINT(POINT-W,CAN),
                                      D-RESTRAINT(PIVOT-L2,CAN)
    +   PROPEL(BOTTLE-OPENER)         PULL(LOC2-L1,UP)

              ↑                                    ↑
     device-use template                  subgoal resolution
```

Figure 5.10 Pragmatic representation of bottle-opener can puncturing.

Fig. 5.10 is comprised of two components: (a) a device-use template, and (b) a possible resolution of the plan subgoals. A subgoal resolution is a template which satisfies the plan but isn't fully specified. The right hand side of Fig. 5.10 and the left hand side of the Fig. 5.11 are subgoal resolutions for the bottle-opener can puncturing plan. The plan to puncture a can with a bottle opener is enabled by achieving five subtasks. First, the user must locate and take control of a device which can: (a) produce the force necessary to puncture the can (I-MECH), and (b) cut the can (I-WEDGE). These are shown in the upper shaded portion of the figure. The bottle-opener satisfies the requirements of the can's CONTROL-RANGE plan, because it has a pointed end which will cut the can, and because it can generate the force necessary to do so with leverage. Second, the user must locate and take control of a can. Since no function is required of the can, no instrumental goals are involved. Third, the user must stabilize the can so that it cannot move relative to the bottle opener. Fourth, the user must make contact between the bottle opener and the can in the correct orientation to enable the function. Last, the user must initiate the function by pulling the bottle opener upward away from the can. These relationships are identified by the subgoal resolution in the figure. The FIND subgoal in Fig. 5.10 is resolved as a combination of knowing what functions are needed in the device and getting to where such a device is located. The I-MECH goal, when satisfied, will be associated with a device that can produce the mechanical advantage needed to open the can. Likewise, the same device will be capable of cutting. The pragmatic representation does not specify anything more about the physical characteristics of the device which is used in the plan, and all other references to regions are those associated with the devices found. The PROPEL action which is part of the device-use plan template is shown instantiated with a PROPEL specialization PULL. The subgoal resolution in Fig. 5.11 does not repeat actions which have been specified in Fig.

5.10, so the PROPEL-PULL is left out.

	subgoal resolution	possible plan components
	I-MECH(BOTTLE-OPENER)	MP-LEVER(LOC2-L1,PIVOT-L2,LOC2-L2)
+	I-WEDGE(BOTTLE-OPENER)	MP-BLADE(LOC2-L2,POINT-W)
+	PTRANS(LOC(BOTTLE-OPENER))	WALK(LOC(BOTTLE-OPENER))
+	GRASP(LOC2-L1)	HOLD(HAND,LOC2-L1)
+	PTRANS(LOC(CAN))	WALK(LOC(CAN))
+	GRASP(CAN)	HOLD(HAND,CAN)
+	D-RESTRAINT(CAN,SURFACE)	PLACE(BOTTOM-CAN,SURFACE,ABOVE)
+	D-RESTRAINT(POINT-W,CAN)	PLACE(POINT-W,CAN-LID,ABOVE) +
	D-RESTRAINT(PIVOT-L2,CAN)	PLACE(PIVOT-L2,CAN-LIP,BELOW)

Figure 5.11 Subgoal resolution and possible plan components for bottle-opener can puncturing.

Fig. 5.11 is also comprised of two components: (a) the unspecified subgoal resolution in Fig. 5.10, and (b) a possible plan instantiation of the items in (a). The right hand side of Fig. 5.11 shows a possible instantiation of the bottle-opener puncture-lid plan. In the figure, role instantiations are related to the parts and regions in the idealized bottle opener pictured in Fig. 5.9. The upper shaded portion shows the bottle-opener regions which are instantiated to produce leverage, and to puncture, respectively. The same regions are shown in the two placements in the lower shaded portion of the figure. PLACE is an instrumental script (not presented here) which is used to relocate and orient the region of one object to the region of another object and results in contact between the regions of two objects. The main conceptualization of PLACE is the MOVE action. The first instance of placement, PLACE(BOTTOM-CAN,SURFACE,ABOVE), means that the result of this plan is the can being supported by the surface. The second and third instances of PLACE describe the placement of the bottle-opener on the can. The first of these places the pointed end of the bottle opener on top of the can's lid, and the second places the pivot under the can's lip. Each results in a local state of contact, and, combined, they represent the interference needed to produce leverage. As mentioned previously, the addition sign (+) between the two placements is a notation meaning that these plans are conjunctive. In these examples, conjunctive, but temporally sequential, plans will use the addition sign, to distinguish them from conjunctive, but temporally simultaneous, plans, which will use a comma (,).

Figs. 5.12 and 5.13 illustrate the same object being used to remove caps from bottles.

```
REMOVE-CAP(BOTTLE-OPENER,BOTTLE) =

      FIND(BOTTLE-OPENER)          I-MECH(BOTTLE-OPENER) +
                                   PTRANS(LOC(BOTTLE-OPENER))
   +  D-CONT(BOTTLE-OPENER)        GRASP(LOC2-L2)
   +  FIND(BOTTLE)                 PTRANS(LOC(BOTTLE))
   +  D-CONT(BOTTLE)               GRASP(BOTTLE)
   +  I-PREP(BOTTLE)               D-RESTRAINT(BOTTLE,SURFACE)
   +  I-PREP(BOTTLE-OPENER)        D-RESTRAINT(LOC2-L1,BOTTLE-CAP),
                                   D-RESTRAINT(PIVOT-L1,BOTTLE-CAP)
   +  PROPEL(BOTTLE-OPENER)        PULL(LOC2-L2,UP)
```

device-use template

subgoal resolution

Figure 5.12 Pragmatic representation of bottle opener bottle cap removing.

The same plan template is used, and the subgoal resolution is very similar; however, a different target object is used, different regions of the bottle opener are used, and the resulting force location and direction are different. The subgoal resolution is different for the remove-cap plan because no cutting is performed, so only leverage (I-MECH) is needed to enable the plan. In this case, the pivot location is the end of the bottle-opener so the device instantiates MP-LEVER-TYPE2 instead of MP-LEVER-TYPE1.

```
      I-MECH(BOTTLE-OPENER)        MP-LEVER(LOC2-L2,LOC2-L1,PIVOT-L1)
   +  GRASP(LOC2-L2)               HOLD(HAND,LOC2-L2)
   +  PTRANS(LOC(BOTTLE))          WALK(LOC(BOTTLE))
   +  GRASP(BOTTLE)                HOLD(HAND,BOTTLE)
   +  D-REST(BOTTLE,SURFACE)       PLACE(BOTTLE,SURFACE,ABOVE)
   +  D-REST(LOC2-L1,BOTTLE-CAP),  PLACE(LOC2-L1,TOP-BC,ABOVE),
      D-REST(PIVOT-L1,BOTTLE-CAP)  PLACE(PIVOT-L1,LIP-BC,BELOW)
```

subgoal resolution

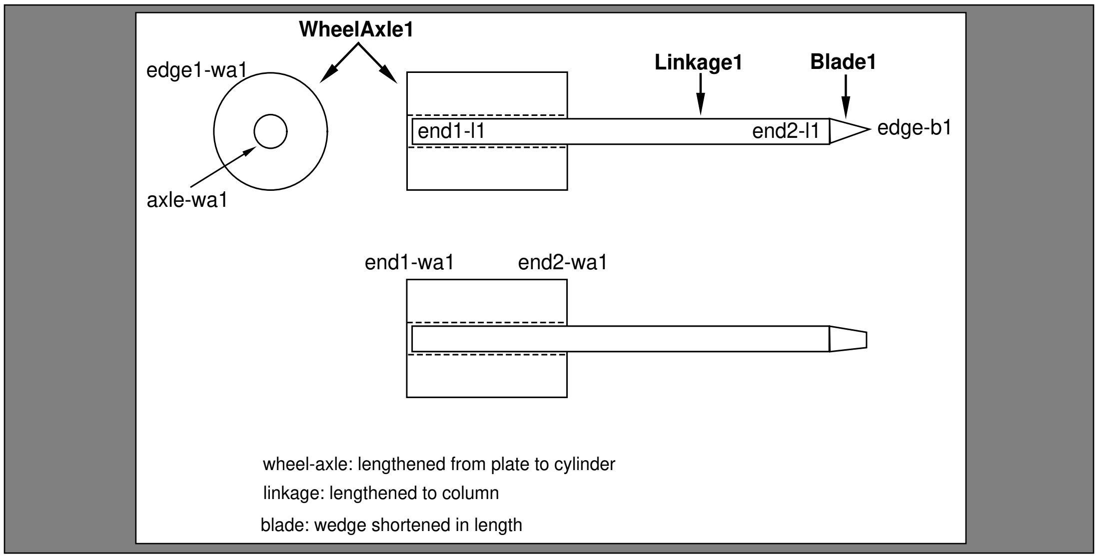
possible plan components

Figure 5.13 Subgoal resolution and possible plan components for bottle-opener cap removal.

In Fig. 5.13 (5), the roles of MP-LEVER are listed in (appl, pivot, react) order, so the bottle-opener regions for this use are (LOC-L2, LOC-L1, PIVOT-L1). Also, for space considerations, D-RESTRAINT is written as D-REST in the figure. As with can-puncturing, the lever regions are placed in their proper locations and orientations with respect to the bottle cap prior to pulling the bottle-opener. Note the different region instantiations of the same I-MECH and D-REST goals in the shaded areas.

Screwdriver

An idealized screwdriver is depicted in Fig. 5.14. The screwdriver has one design-intended use: (1) driving, which is shown in Fig. 5.15. Driving is used to create torque (i.e., a TRADEOFF-FORCE plan) in support of connecting objects (i.e., a CONTROL-RANGE plan). The driving plan is applied by (a) placing the screw point in contact with the proper substance, (b) inserting the screwdriver tip into the screw slot, and (c) turning the screwdriver handle. The driving plan makes use of a torque-generating device (MP-WHEEL-AXLE) function. When the handle is grasped and the tip inserted into the screw slot, the screwdriver instantiates the device function. When the screwdriver handle is turned, the function is initiated.



wheel-axle: lengthened from plate to cylinder
linkage: lengthened to column
blade: wedge shortened in length

Figure 5.14 Idealized screwdriver.

Fig. 5.15 illustrates a PROPEL specialization, TURN, which causes the production of torque on the screwdriver handle. The delta-restraint goal is also slightly different than that in the bottle opener use, since the screwdriver must be kept from turning so that it can transmit torque to the screw. This is accomplished by inserting it into the screw slot. INSERT, like PLACE, is an instrumental script which results in restraint states. In the case of INSERT, the result is confinement rather than contact.

```
DRIVING(SCREWDRIVER,SCREW) =

     FIND(SCREWDRIVER)             I-MECH(SCREWDRIVER) +
                                   PTRANS(LOC(SCREWDRIVER))
   + D-CONT(SCREWDRIVER)           GRASP(EDGE-W)
   + FIND(SCREW)                   PTRANS(LOC(SCREW))
   + D-CONT(SCREW)                 GRASP(SCREW)
   + I-PREP(SCREW)                 D-REST(SCREW,HOLE)
   + I-PREP(SCREWDRIVER)           D-REST(SCREWDRIVER,SCREW)
   + PROPEL(SCREWDRIVER)           TURN(EDGE-W)
```

device-use template

subgoal resolution

Figure 5.15 Pragmatic representation of screwdriver driving.

```
     I-MECH(SCREWDRIVER)           MP-WHEEL-AXLE(EDGE-W,EDGE-B)
   + D-REST(SCREW,HOLE)            PLACE(TIP-SCREW,HOLE,ABOVE)
   + D-REST(SCREWDRIVER,SCREW)     INSERT(EDGE-B,SLOT-SCREW,ABOVE)
```

subgoal resolution

possible plan components

Figure 5.16 Subgoal resolution and possible plan components for screwdriver driving use.

5.3.2 Simple Compound Devices

Simple compound devices have two relatively moving components. As a result, the use of this class of devices requires control of both components, and the device has more than one recognizable state. The functions are thus identified by behavioral state as well as input type and static representation, generally resulting in subplans which apply component functions within the overall device-use plan. The pragmatic representations for two simple compound devices: (1) a nutcracker, and (2) a clothes pin are presented in this section. Three additional devices: (3) scissors, (4) pliers, and (5) door are presented in Appendix A.2.

As devices increase in complexity, their use becomes more specialized, and their application is more strongly associated with their intended function. For example, a pair of scissor is designed to cut a straight line in some material, which is a combination of the functions associated with MP-LEVER and MP-BLADE. The I-MECH and I-WEDGE subgoals combine to identify devices like scissors which have been used to perform such tasks, rather than to search for devices which can resolve each goal independently and collectively.

Nutcracker

An idealized nutcracker is depicted in Fig. 5.17. The nutcracker has a single design-intended use: (1) remove nut meats from nut shells (a CONTROL-RANGE plan), as illustrated in Fig. 5.18. The nutcracker is itself a confinement device, which both restrains the object and produces the force necessary to open it. The spring used in this device automatically prepares the device for use, so the use plan is simpler than that of a pair of scissors or pliers even though they are very similar. The only plan of import is the insertion/placement of the nut between the handles. In other examples of insertion, the object is associated with a containment volume. The nutcracker is associated with a space bounded by two objects and is a confinement space. The device is used to remove nuts by cracking their shells open. The procedure is initiated by locating an object which can: (a) produce enough force to overcome the strength of the shell which, (b) confine the nut during use, and (c) passively return the device to its starting state so that the device can be operated with one hand. In the case of a real nutcracker, this step degenerates to a PTRANS. The user must then gain control of the nutcracker and nut. Preparation of the nutcracker involves only holding the handles,

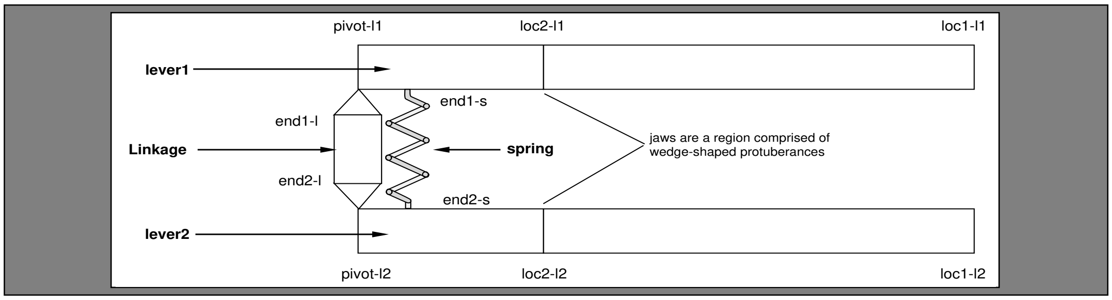


Figure 5.17 Idealized nutcracker.

```
CRACK-OPEN(NUTCRACKER,PECAN) =

    FIND(NUTCRACKER)              I-MECH(NUTCRACKER) +
                                  I-CONTN(NUTCRACKER) +
                                  I-RESIST(NUTCRACKER) +
                                  PTRANS(LOC(NUTCRACKER))
 +  D-CONT(NUTCRACKER)            GRASP(NUTCRACKER)
 +  FIND(PECAN)                   PTRANS(LOC(PECAN))
 +  D-CONT(PECAN)                 GRASP(PECAN)
 +  I-PREP(NUTCRACKER)            GRASP(LOC1-L1),
                                  GRASP(LOC1-L2)
 +  I-PREP(PECAN)                 D-RESTRAINT(PECAN,NUTCRACKER)
 +  PROPEL(NUTCRACKER)            PRESS(LOC1-L1),
                                  PRESS(LOC1-L2)
```

device-use template

subgoal resolution

Figure 5.18 Pragmatic representation of nutcracker nut cracking.

but preparation of the nut requires that it be confined by the nutcracker. Finally, the handles are pressed together. The subgoal resolution of the nutcracker device-use plan illustrates (a - c) as a combination of I-MECH, I-CONTN, and I-RESIST goals, each of which must be achieved using the same device. This plan identifies a spring with the device (I-RESIST), though the device would degenerate to a pliers-like device if the spring were to break. This would require manual opening and the device-use plan would then be identical to that for pliers or scissors.

Fig. 5.13 illustrates a possible specification for the items highlighted in Fig. 5.18. The nutcracker components and regions instantiate two instances of MP-LEVER (type2), MP-CONTAINER, and MP-SPRING. The INSERT script is instantiated by the nut and the nutcracker regions associated with the react MP-LEVER-TYPE2 role.

subgoal resolution	possible plan components
`I-MECH(NUTCRACKER)`	`LEVER-TYPE2(LOC1-L1,PIVOT-L1L2,LOC2-L1),` `LEVER-TYPE2(LOC1-L2,PIVOT-L1L2,LOC2-L2)`
`I-CONTN(NUTCRACKER)`	`CONTAINER(LOC2-L1,LOC2-L2)`
`I-RESIST(NUTCRACKER)`	`SPRING(LOC2-L1,LOC2-L2)`
`D-REST(PECAN,NUTCRACKER)`	`INSERT(PECAN,LOC2-L1,LOC2-L2)`

Figure 5.19 Subgoal resolution and possible plan components for nutcracker use.

Clothes Pin

An idealized clothes pin is depicted in Fig. 5.20. The clothes pin has a single design-intended use: (1) holding two objects together, which is a confinement (CONTROL-RANGE) plan. The clothes pin has both opening and closing functions embedded in its application, as shown in Fig. 5.21. As with the nutcracker, a clothes pin's spring eliminates some of the preparatory work the user has to perform (1). Unlike the nutcracker, which returns the device to its initial (open) state, the clothes pin is found in and returns to its final (closed) state. The clothes pin has identical functional prerequisites to the nutcracker: (a) develop

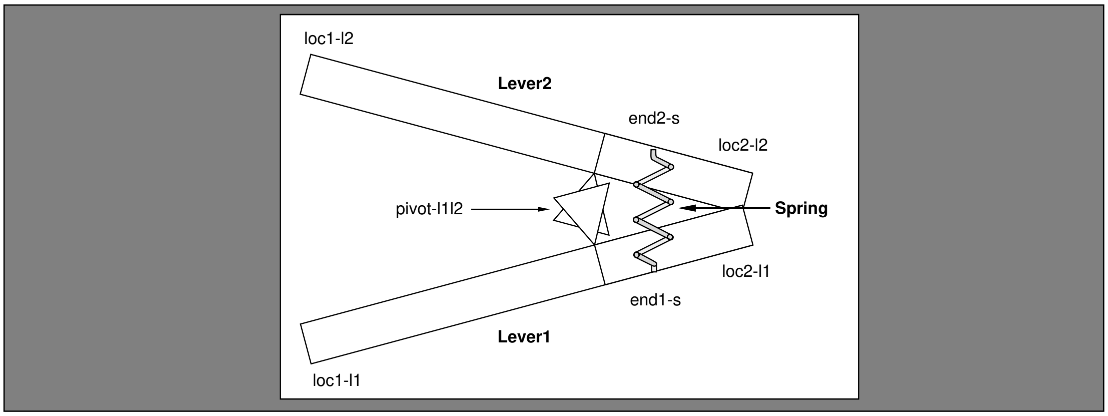


Figure 5.20 Idealized clothes pin.

force, (b) confine an object, and (c) use passive resistance to support one-handed operation. As with the nutcracker, the clothes pin components and regions achieve the I-MECH, I-CONTN, and I-RESIST subgoals; however, the clothes pin instantiates MP-LEVER (type1) instead of type2. Preparation of the clothes is illustrated in Fig. 5.21 with a HANG instrumental script.

```
HOLDING(CLOTHES-PIN,CLOTHES,CLOTHES-LINE) =

      FIND(CLOTHES-PIN)          I-MECH(CLOTHES-PIN) +
                                 I-CONTN(CLOTHES-PIN) +            ①
                                 I-RESIST(CLOTHES-PIN) +
                                 PTRANS(LOC(CLOTHES-PIN))
   +  D-CONT(CLOTHES-PIN)        GRASP(CLOTHES-PIN)
   +  FIND(CLOTHES)              PTRANS(LOC(CLOTHES))
   +  D-CONT(CLOTHES)            GRASP(CLOTHES)
   +  D-FIND(CLOTHES-LINE)       PTRANS(LOC(CLOTHES-LINE))
   +  I-PREP(CLOTHES)            HANG(CLOTHES,CLOTHES-LINE)
   +  I-PREP(CLOTHES-PIN)        OPEN(CLOTHES-PIN) =
                                   GRASP(LOC1-L1),
                                   GRASP(LOC1-L2) +                ②
                                   PRESS(LOC1-L1),
                                   PRESS(LOC1-L2)
                                   PLACE(CLOTHES-PIN,CLOTHES,ABOUT)
   +  PROPEL(CLOTHES-PIN)        RELEASE(LOC1-L1),                 ③
                                 RELEASE(LOC1-L2)

              ↑                              ↑
      device-use template        possible plan components
```

Figure 5.21 Pragmatic representation of clothes pin holding.

As with other such scripts mentioned, HANG is not represented here. Preparation of the clothes pin is represented as a separate OPEN plan (2), and is very similar to pliers (and scissors) devices. The ends are held in the hand and pressed together, and the clothes are placed between the ends. The other difference, which is seen at (3) in the figure, is that the DO action for the clothes pin, because of the spring and geometry, is simply a unGRASP (release) of the handles, instead of a PROPEL.

5.3.3 Multiple Compound Devices

Multiple compound devices may have multiple independently moving components. This increase in complexity has two consequences on the representation of device use for this class of devices. First, as the number of components which move independently increases, the number of design-intended uses tends to decrease toward 1. Second, as the number of components which move independently increases, knowledge of the functions affecting individual components, and how they interact, must be captured by the plan representation. The pragmatic representation for the finger nail clipper is presented in this section, and the representations for three other devices: (1) bottle lid loosener, (2) press, and (3) crank can opener are presented in Appendix A.3.

Finger Nail Clipper

An idealized finger nail clipper is depicted in Fig. 5.22. The finger nail clipper has a single design-intended use: removing finger nails, which is a separation (i.e., SEPARATE-PIECES) plan. Like the nutcracker and clothes pin, the finger nail clipper must constrain the motion of the finger nail prior to clipping it, so the components of a CONTROL-RANGE plan will be embedded in its use as well. The device-use plan for the finger nail clipper is shown

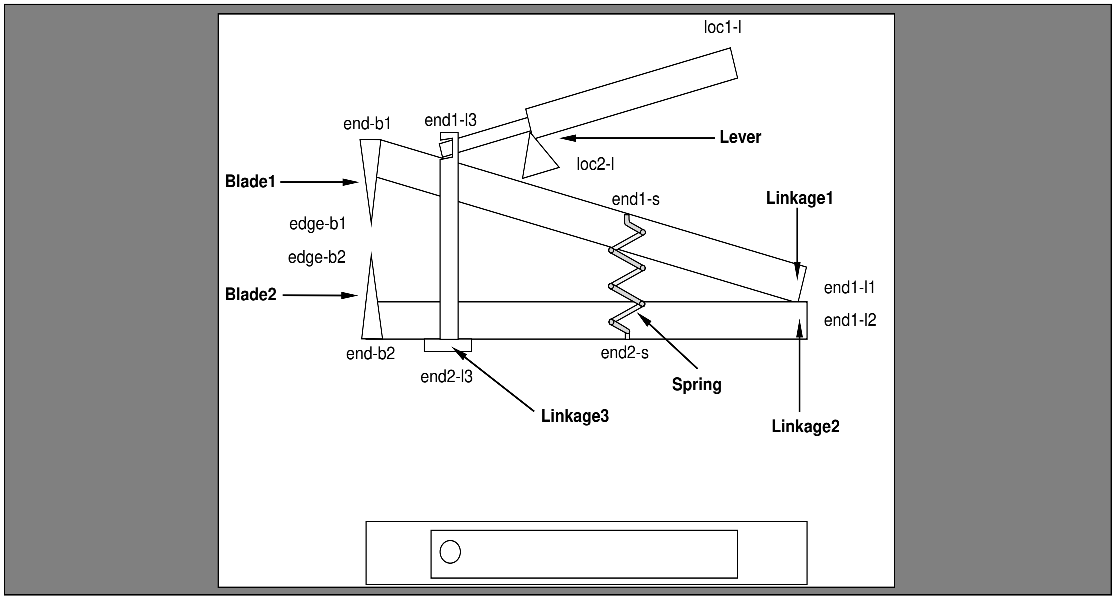


Figure 5.22 Idealized finger nail clipper.

in Fig. 5.23.

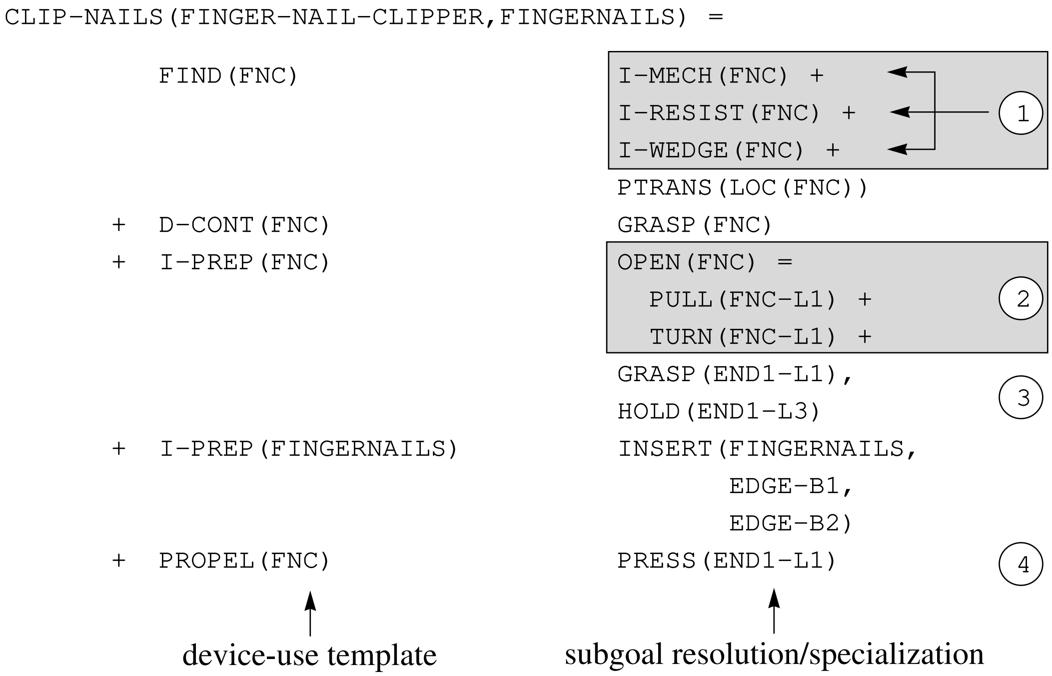


Figure 5.23 Pragmatic representation of finger nail removal.

The device needs three capabilities: (a) to provide enough force to cut, (b) to cut, and (c) to return to its initial state so that it can be operated with one hand. These requirements are represented with the three mechanical instrumental goals I-MECH, I-RESIST, and I-WEDGE, (at 1 in Fig. 5.23). The lever which is used to initiate the cutting function can move independently with respect to the pin linkage (FNC-PIN), and with respect to the linkage which is the upper blade (FNC-L2). The blades (FNC-B1 and FNC-B2) can move independently with respect to each other. The device is prepared in two stages. The first stage is to open the lever (at 2) and then rotate it into position. These actions enable the spring to relax, which enables the blades to move to their initial (OPEN) positions. The preparatory actions of grasping the lever and holding the lower linkage (at 3) enable: (a) the initiating action for the lever, and (b) the restraint of the device which will keep it from moving when the force is applied. By pressing on the lever (4), the device function is enabled and initiated. The finger nail clipper use is similar to the nutcracker use in that the spring returns the device to its initial position after use. This similarity is captured by the representations even though the devices have different functions.

5.3.4 Complex Devices

Complex devices have components which may themselves be multiple compound devices. As a result, the use of this class of devices requires integration of the functions of its components, each of which has more than one recognizable state. The use of one complex device, a toy gun, is discussed in this section. Appendix A.4 presents pragmatic representations for three other complex devices: (1) cork screw, (2) egg beater, and (3) mouse trap.

Toy Gun

A toy gun is a complex device which is used to propel a projectile into contact with a distant object (a MAKE-CONTACT plan). The device has arming and firing mechanisms, a housing, and a projectile, as shown in its idealized form in Fig. 5.24. The toy gun is used to make-contact-at-distance using a shooting plan, as shown in Fig. 5.25. The projectile is the object which makes contact (at 1 in Fig. 5.25), and the rest of the device is designed to produce the necessary force (2), to control the projectile (4), and to control its firing (3). The housing serves to confine the projectile and to determine its path of motion. The arming and firing mechanism utilizes leverage to restrain the stored energy until the chosen time for release. The projectile is reusable and can be disconnected from the device during nor-

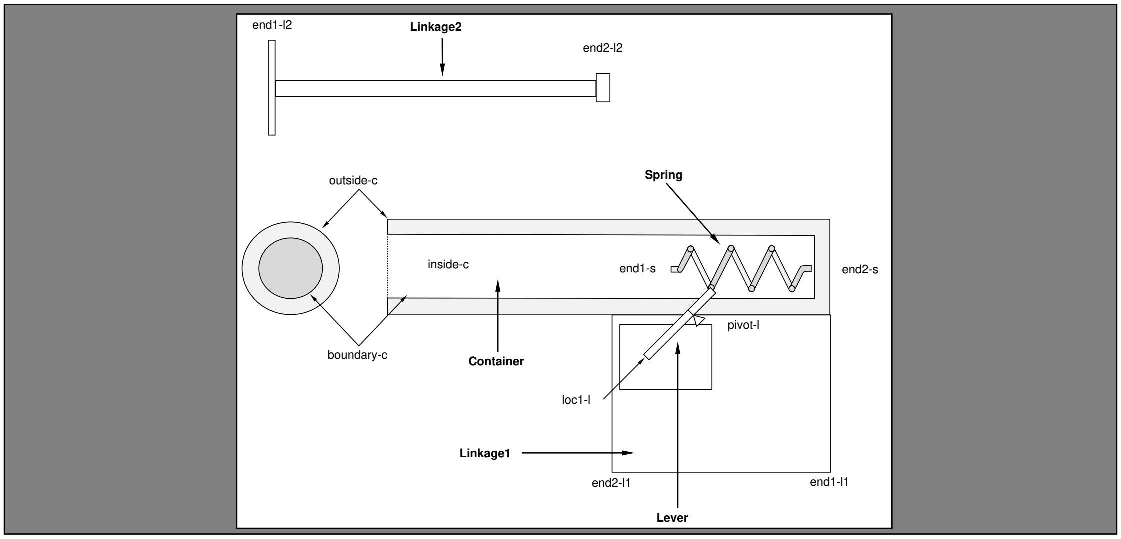


Figure 5.24 Idealized toy gun.

mal operation. The device is used by inserting the dart into the barrel. This action enables

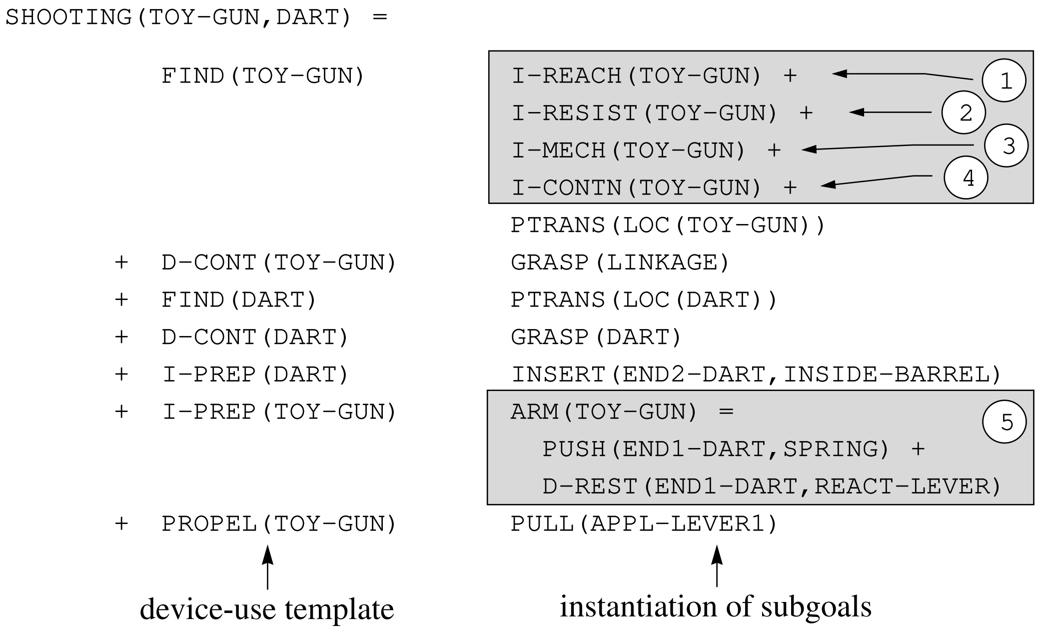


Figure 5.25 Pragmatic representation of toy gun shooting.

the arming by providing a vehicle for applying force that will compress the spring inside of the gun. When the gun is armed, it can be fired by pulling the trigger.