Chapter 9
Physigrams: Modelling Physical Device Characteristics Interaction

Alan Dix and Masitah Ghazali

Abstract In industrial control rooms, in our living rooms, and in our pockets, the devices that surround us combine physical controls with digital functionality. The use of a device, including its safety, usability and user experience, is a product of the conjoint behaviour of the physical and digital aspects of the device. However, this is often complex; there are multiple feedback pathways, from the look, sound and feel of the physical controls themselves, to digital displays or the effect of computation on physical actuators such as a washing machine or nuclear power station. Physigrams allow us to focus on the first of these, the very direct interaction potential of the controls themselves, initially divorced from any further electronic or digital effects—that is studying the device 'unplugged'. This modelling uses a variant of state transition networks, but customised to deal with physical rather than logical actions. This physical-level model can then be connected to underlying logical action models as are commonly found in formal user interface modelling. This chapter describes the multiple feedback loops between users and systems, highlighting the physical and digital channels and the different effects on the user. It then demonstrates physigrams using a small number of increasingly complex examples. The techniques developed are then applied to the control panel of a wind turbine. Finally, it discusses a number of the open problems in using this kind of framework. This will include practical issues such as level of detail and times when it feels natural to let some of the digital state 'bleed back' into a physigram. It will also include theoretical issues, notably the problem of having a sufficiently rich semantic model to incorporate analogue input/output such as variable finger pressure. The latter connects back to earlier streams of work on status–event analysis.

A. Dix (✉)
School of Computer Science, University of Birmingham, Birmingham, UK
e-mail: alan@hcibook.com

A. Dix
Talis Ltd. Birmingham, Birmingham, UK

M. Ghazali
University of Technology Malaysia, Johor, Malaysia
e-mail: masitah@utm.my

B. Weyers et al. (eds.), *The Handbook of Formal Methods in Human-Computer Interaction*, Human-Computer Interaction Series,
DOI 10.1007/978-3-319-51838-1_9

9.1 Introduction

In industrial control rooms, in our living rooms, and in our pockets, the devices that surround us combine physical controls with digital functionality. The use of any device, including its safety, usability and user experience, involves the conjoint behaviour of its physical and digital aspects; however, this is often complex.

The digital side of this has been the subject of extensive research in the formal user interface modelling community. However, there has been very little work that addresses the physical interaction, and this chapter addresses that gap. We present physigrams, a semi-formal modelling notation for describing the physical interactions of devices and how this links to underlying digital behaviour.

The chapter begins by describing the multiple feedback loops between users and systems, highlighting the physical and digital channels and the different effects on the user with reference to the case studies and other examples. It will then demonstrate physigrams using a small number of increasingly complex examples, largely of domestic appliances for which the methods were first developed.

Having laid the groundwork, we then look at a more practical use where product designers used the notation to describe three potential options for a media-player. This exposes some of the advantages of a semi-formal notation, as the designers were able to extend the notation to deal with emerging issues in the design.

Finally, physigrams are used in a case study of a wind-turbine control panel. This explores the potential for use in real-world industrial control panels and some of the practical problems in studying detailed physical phenomena of critical equipment while in use.

Finally, the chapter will describe some of the open problems in using this kind of framework. This will include practical issues such as level of detail and times when it feels natural to let some of the digital state ‘bleed back’ into a physigram. It will also include theoretical issues, notably the problem of having a sufficiently rich semantic model to incorporate analogue input/output such as variable finger pressure. The latter will connect back to earlier streams of work on status–event analysis.

9.2 Physical and Digital Feedback Loops

The provision of feedback is a core feature in most lists of interaction design heuristics and guidelines. For example, in Shneiderman’s *Eight Golden Rules* number three is “offer informative feedback” (Shneiderman and Plaisant 2010), and Norman’s *Execution–Evaluation Cycle* is as much about interpreting feedback as it is about formulating and performing actions (Norman 1988).

For PC-based systems, the main feedback is visual, the effects of keystrokes and mouse/pointer movements on virtual screen objects. However, when we look at richer digital-physical systems, we find that there are multiple feedback pathways, from the look, sound and feel of the physical controls themselves, to digital displays or the effect of computation on physical actuators such as a washing machine or nuclear power station.

Figure 9.1 looks at a number of different kinds of feedback using the paths of action and communication between the user's body, an input device (e.g. a control knob, or button) and the underlying digital system. The various paths are described in detail elsewhere (Dix et al. 2009), so here we will just work through a few examples.

First consider a public display in a café. The café owner is about to open and so presses the power switch for the display (physical manipulation (a)). The switch moves and the café owner can both feel and see the switch is now on (a directly perceived state of the control switch (b) leading to feedback loop A). In addition, after a few seconds of boot-up, the public display system shows a message (electronic feedback (d), loop C).

As another example, the operator of a nuclear power station wishes to reduce the reactor power output to tick-over level ready for maintenance. The operator twists a 'soft' dial (a). The dial is fitted with a haptic feedback system: when the system detects the dial movement, it produces a small resistance to any movement (physical effects in the device (ii)), but has much stiffer resistance at the point at which the reactor would shut down completely. Because the operator is able to feel this (digital feedback similar to physical feedback (c), feedback loop B), it is easy to move the dial to the minimum point without danger of over-shooting the critical level. In addition, actuators within the reactor begin to lower the control rods (physical effects on the environment (iv), feedback loop D).

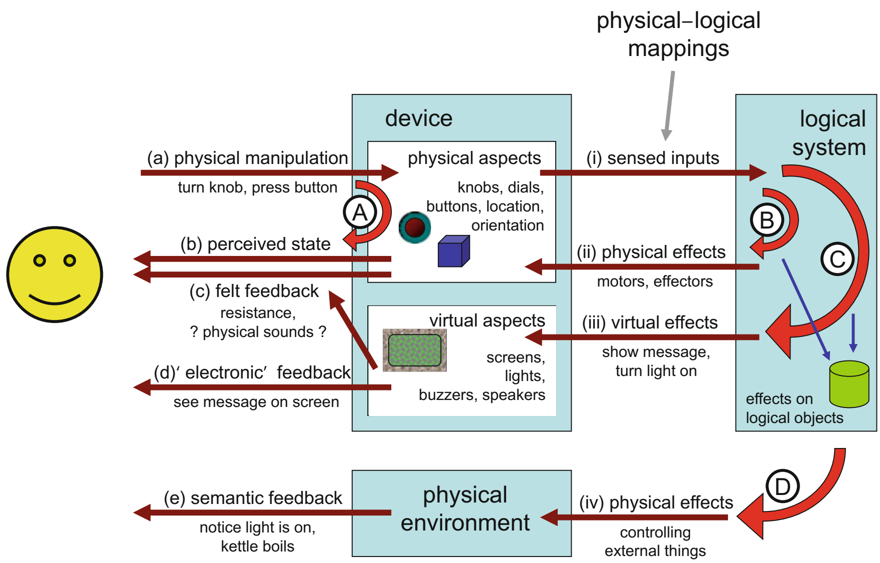


Fig. 9.1 Multiple feedback loops (from Dix et al. 2009)

While loop D is very clearly of a different nature, the distinctions between loops A, B and C are a little more subtle. From a *system* point of view, loops B and C are effectively identical and would appear so in most formal modelling notations. However, *for the user* the opposite is the case, very well designed and executed haptic feedback (loop B) might seem just as if it were direct physical resistance in a device (loop A). Indeed, if the implementation really is perfect, then there is no difference for the user. However, any slight or occasional imperfections can be critical; for example, we describe elsewhere how delays of only a fraction of a second in simulated key-clicks made it almost impossible to triple tap on a mobile phone (Dix et al. 2009).

All of these loops are important; however in this chapter we will focus principally on loop A, the immediately perceived effects of the physical manipulation of the device. That is the things you feel whether or not the digital aspect of the system is actually working. We have called this analysing the 'device unplugged' (Dix et al. 2009).

9.3 The Device Unplugged

Imagine visiting a nuclear power plant that is being decommissioned. All of the wiring and electronics have been removed, but the shell of the control room is still there. You can twist dials, press buttons, pull levers, but nothing does anything. To take a more day-to-day example, it is common to see a small child playing with their parent's phone. If the power has run out (maybe because of the child's playing), would there be things the child could still play with, such as buttons to press?

The idea of considering the device unplugged is to foreground the interaction potential of the physical controls themselves.

Affordances are an important concept in HCI. In Gibson's original definition, "*the affordances of the environment are what it offers the animal, what it provides and furnishes, either for good or for ill*" (Gibson 1979, p. 127). Following adoption in the HCI community, led by Norman and Gaver (Gaver 1991; Norman 1988, 1999), we often think about the apparent affordances of on-screen controls; however, Gibson was focused on more basic physical interactions: a rock affords picking up and throwing if it is of a suitable weight and size for the hand.

Looking at a light-switch, we can consider the whole lighting system of the house and say that the switch affords turning on the light, but before that the fact that the switch looks and feels switch-like means it affords pressing. While later writers have considered other aspects, such as cultural knowledge, when considering simple physical actions Gibson claimed that affordances were immediately perceived, in the sense that our whole perceptual systems are tuned to see the interaction potential of the world. The argument for this is that as a species we are fitted for our environment and optimised for action. Of course, for complex digital systems there is no guarantee that this will be the case.

Many of the problems with pervasive technologies happen where this more immediate affordance is missing. A common example is the water for handbasins in public toilets. The water is often turned on by a proximity sensor, but there is little to indicate this theoretical affordance except the knowledge that something must make the water go on (cultural knowledge). If you imagine the bathroom 'un-plumbed', with the water turned off and the pipes empty, a traditional water tap (faucet) can still be twisted, but the sensor-based one has no physical interaction potential at all.

Of course, the complete system is not just the physical controls, we also have to think about the mapping between physical actions and digital effects; even if we know that a switch can be pressed, what happens when it is pressed? However, by initially focusing on the device unplugged we both redress the more abstract action- and functionality-oriented descriptions that are common, and also lay proper emphasis on the first point of contact between human and machine.

9.4 Modelling the Device Unplugged

The vast majority of more formal modelling of interaction is focused on the dialogue level of the Seeheim model (Pfaff and ten Hagen 1985). Many techniques are used, from state transition networks (STNs) to Petri nets, but almost all assume user inputs at the level of abstracted actions, such as 'quit button pressed', rather than the actual physical action of pressing the button.

There are occasional descriptions of issues closer to the physical interface. For example, in even the earliest editions of the Human–Computer Interaction textbook (Dix et al. 2004), a case study is described where a change in the keyboard layout led to errors as it became possible to accidentally press adjacent keys forming an important and critical key sequence (Fig. 9.2).

Thimbleby has also looked at physical placement of buttons on a fascia and analysed how this might affect relevant time costs of button press sequences due to Fitts' Law delays for longer movements (Thimbleby 2007). Also, safety cases and accident reports include all levels of interaction: for example, the placement of a label at Three Mile Island hid an important indicator.

As a way to model the physical aspects of the device unplugged, we use *physigrams*, a variation of *state-transition networks* (STNs) with additional features

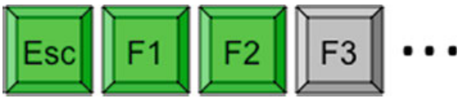


Fig. 9.2 Layout matters (from Dix et al. 2004). The key sequence F1–F2 meant "exit and save if modified", whereas F1–esc–F2 meant "exit without saving". This was fine on keyboards where the function and escape keys were separated, but on the above layout led to accidentally pressing the escape when trying to hit F1–F2

to deal with particular kinds of physical interactions (for example, buttons that 'bounce back' after being pressed). The name 'physigram' was coined by product designers who used the notation to model prototype devices. Crucially the STN-based notation is precise enough to give some analytic traction, whilst readable enough for product designers to use, even though they would normally avoid more formal descriptions.

9.5 Physigrams—Modelling Physical States

Figure 9.3 shows the simplest example of a physigram, the states of a simple on/off switch. It has two states, and the transition between the two is controlled solely by the user.

Of course, just looking at the switch 'unplugged', you cannot tell what it does electrically:

Is it a light switch, or controlling a cooling fan?

If it is a light switch, which light?

However, when you press the switch you can immediately see and feel that you have changed something, even if the ultimate effect is not clear (it may be a fluorescent light that takes a few seconds to warm up, or even an outside light that is not visible from where you are).

Not all switches are like this; Fig. 9.4 shows an example of a *bounce-back button*. This is the kind of button you quite frequently find on computers; you press it and, as soon as you release it, the button immediately bounces back to its original position. Although, like a click switch, it has two main states, there is only one *stable state*, the other is a transient *tension state* that requires constant pressure to be applied.

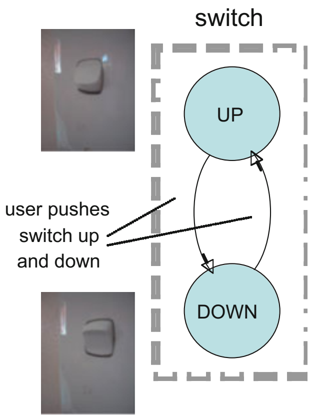


Fig. 9.3 Physigram states of a switch

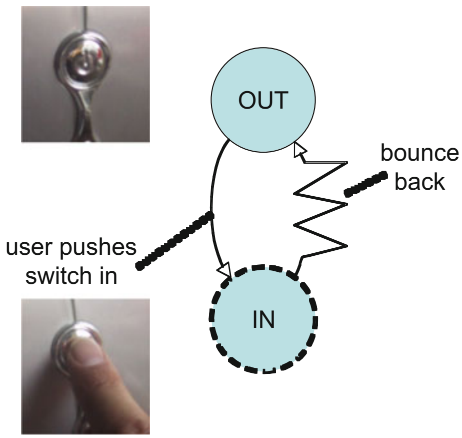


Fig. 9.4 Physigram of a bounce-back button

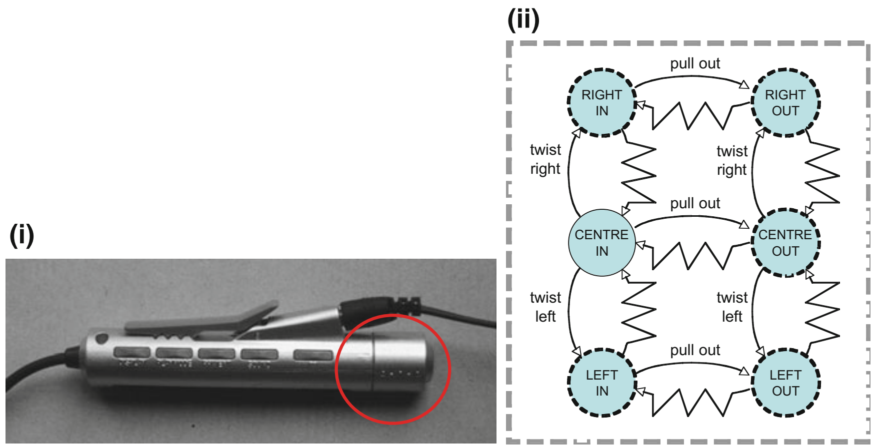


Fig. 9.5 **i** Minidisk controller, **ii** physigram

The physigram denotes this bounce-back using the jagged spring-like arc; whilst the solid transition is triggered by the user's action, the bounce-back is physically autonomous, usually just a spring inside.

These simple transitions can be used to represent more complex controls. Figure 9.5 shows the physigram of an old minidisk controller. The knob at the end can be pulled outwards, but once released it snaps back to the 'IN' state (a *bounce-back*). This is shown in the middle of the physigram (CENTRE IN <-> CENTRE OUT); the pull-out transition is controlled by the user, but it has a bounce-back transition back in. This centre part (turned on its side) is exactly the same as Fig. 9.4, except it is a pull rather than push action for the user.

However, the minidisk controller uses this pull not to trigger some action, but effectively as a mode control and the knob can be twisted as well. Twisting the knob when 'IN' changes the volume, but while pulled out skips forward and back through tracks.

While the effect of the twisting is not part of the physical 'unplugged' state, the twists themselves are modelled in the full physigram (Fig. 9.5ii). Note that the left and right twists are themselves bounce-backs; the only stable state is when the knob is in the 'IN' position and centred; all other states are *tension states* requiring some continued physical force.

As with many formal notations, there is some flexibility in terms of the level of detail you wish to capture with a physigram. Figure 9.3 showed a simple switch, but if you press a switch just lightly it 'gives' a little, yet does not snap to the other position until you are more than half way. That is, as well as the up and down states, there are many part-up and part-down states in between.

Figure 9.6i shows this more detailed view of the physical states and transitions including the spring-like bounce-back and the 'give' as a lightning-like transition. Note the give is similar to bounce-back as the physical properties create the transition, but whilst the bounce-back only happens when tension is released, the 'give' works with the user's physical force.

Whilst this clearly provides more details about the physical interaction potential than Fig. 9.3, arguably it could be far more detailed still as there is not a single 'part down' state, but one where it is down just a little, then down a little more. The choice of level will depend on the issues being analysed. In our previous work on this issue, we suggested that a more detailed status–status mapping view of this

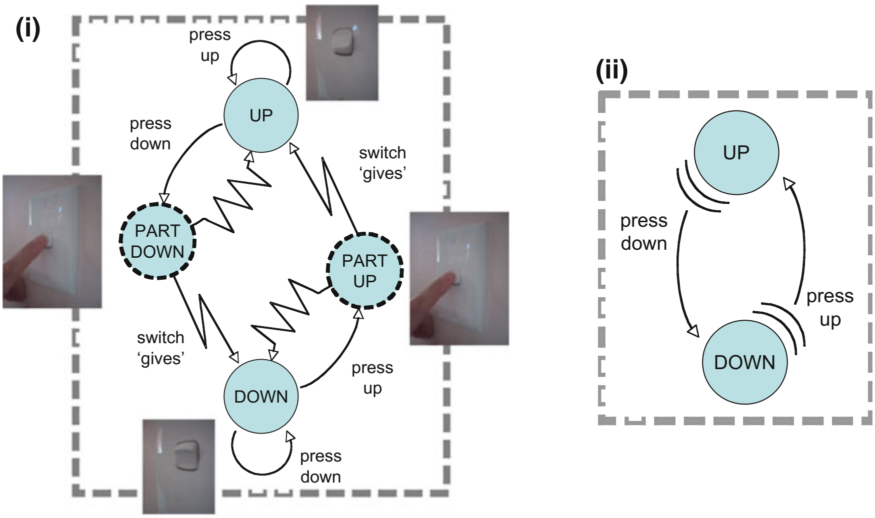


Fig. 9.6 Switch with 'give' **i** detailed physigram, **ii** 'shorthand' physigram with decorated transition

would be valuable, and since then Zhou et al. (2014) have produced analyses of button pressing using force-displacement graphs and taking into account dynamics such as weight and inertia. One could imagine being able to 'zoom' into the transitions in Fig. 9.6ii and see such a graph for the particular choice of button, or even annotating the transitions in some way to distinguish different levels of resistance or dynamics where this is relevant.

As this slight 'bounce-back then give' is a common phenomenon, a shorthand version is introduced in Fig. 9.6ii. This has much of the clarity of Fig. 9.3, but the transition is annotated to show that it has the initial resistance.

9.6 Plugging in—Mappings to Digital State

While we have emphasised the importance of describing the device unplugged, this is of course only part of the story. In the end we are interested in the whole system, physical and digital. To do this we also model the logical states of the digital or electronic side of the system, and the relationship between these and the physigram. We will use STNs to model the logical state also, but one of the many dialogue or system modelling notations could be used for this.

The idea of natural 'mappings' between controls and effects has been a core concept in interaction design since very early days (Norman 1988). There are many different kinds of mapping; in some cases if the controlled thing is spatial and the controls are laid out spatially we might look for spatial correlation: the light on the right is controlled by the switch on the right; or proximity: the switch to turn on the light is the one closest to it.

The relationship between digital states and physigram states is more about the dynamics of interaction.

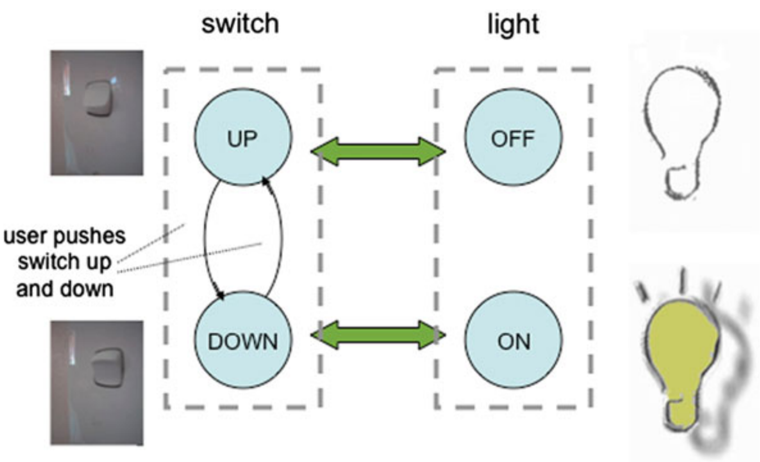


Fig. 9.7 Logical states of an electric light map 1–1 with physigram states

Figure 9.7 shows the physigram of a light switch on the left, and on the right the logical states of the light, whether it is on or off. Strictly there are probably additional states, when the light bulb is defective, there is a power cut, etc., but we are simply modelling the common illumination states.

In this case, the relationship between the physigram states and logical states is 1–1, but of course this is not always the case. Figure 9.8 shows a bounce-back switch. Except maybe for a signalling flashlight, it would be rare to find a bounce-back switch in 1–1 correspondence with the logical state. Most often, a user-determined transition in the physigram triggers a transition in the logical system.

In this example, we have labelled the physigram transition as an abstract action (a) and then shown on the logical system where this abstract action causes transitions. While state–state mappings are easy to represent diagrammatically, transition–transition mappings are harder to portray, especially as they may be many-to-many (as in Fig. 9.8). The use of named actions to link STNs is similar to the technique used in statecharts (Harel 1987), which, due to their adoption in UML, are likely to be reasonably familiar to many in computing.

In the case in Fig. 9.8, we have shown the bounce-back switch as a press on/press off button. In fact this is more likely for a light control, since powering a computer down may cause damage to the system. Quite commonly, the bounce-back switch is used to turn on a computer, but the 'off' is soft, determined by the computer after a software shutdown.

Figure 9.9 shows this alternative. Note how the physigram of the physical control is the same—you cannot tell by playing with the switch alone, when unplugged from the power supply, what action it will have. However, the mapping between physical and logical states allows us to model the different behaviour.

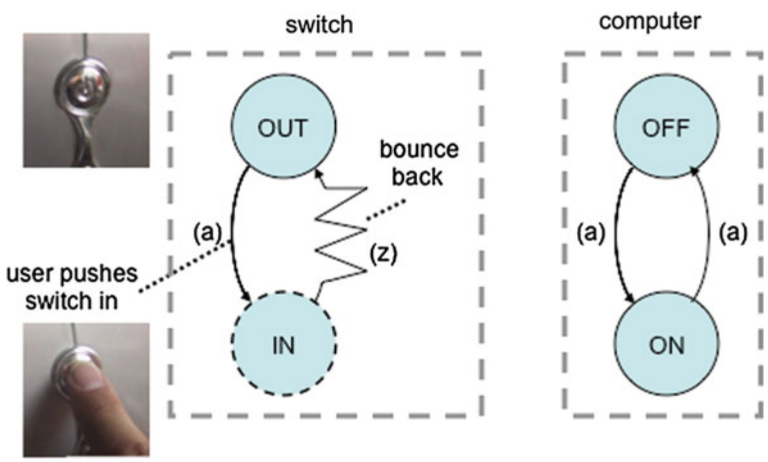


Fig. 9.8 Physical and logical states of a bounce-back switch

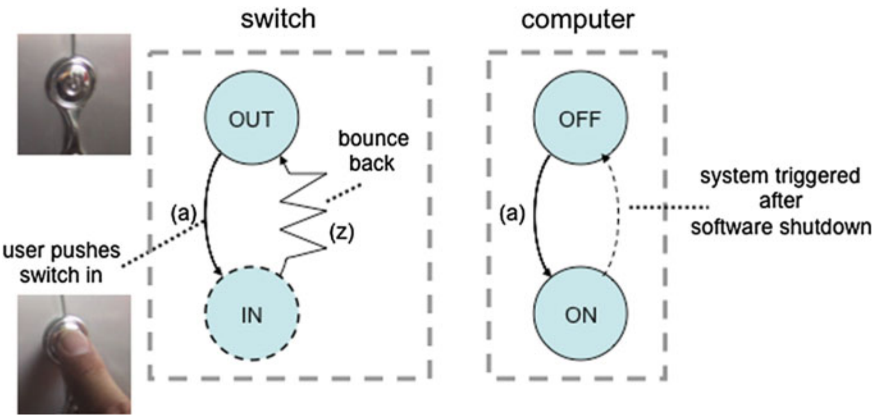


Fig. 9.9 Bounce-back switch only controlling one transition direction

9.7 Properties of Physical Interactions

We are now in a position to talk about some of the design properties of physical–digital devices.

First of all, consider again the *1–1 mapping* between device state and logical state for the light switch in Fig. 9.7. This is an example of *exposed state*; that is where the logical system state can be apprehended directly from some aspect (visible, tactile) of the device. Note this does not necessarily mean that the user can know what that logical state is (for example, whether the switch controls a light out of sight or is in fact turning on the nuclear emergency alarm). However, once this is known, it means that the system state is instantly observable.

The opposite of exposed state is *hidden state*: for example, in Fig. 9.8, the switch does not expose the state of the computer system. Of course, you might see the screen flicker or hear the disk start to spin, that is, in terms of Fig. 9.1, feedback loops C or D. However, there is the danger that if there is a small delay you might think you have not pressed the button properly and press again, accidentally turning the computer back off in mid-boot-up.

Clearly exposed state is a very powerful property when it is possible, but it is not always achievable.

Part of the power of digital systems is that vast amounts of system state can be packaged into very small footprints; if every state of a typical phone were to be made physically accessible in terms of dials, knobs, or switches, the control panel would probably stretch over an area the size of a city.

Bounce-back buttons are most useful where there is a large or unbounded number of logical states, for example the number of tracks on a music player. However, here, as the physical device feedback loop (loop A) is weaker, it is important that there is additional logical feedback. In the case of the minidisk player, increasing/decreasing the volume will be apparent—except when it happens to be during a quiet point. For example, "Funeral for a Friend", the first track of Elton John's album "Goodbye Yellow Brick Road", starts very quietly, with the distant sound of an owl hoot; if you adjust the volume at this point you are likely to be deafened moments later. Some sound controls deliberately make a small sound immediately after the level is set, to help deal with this problem.

Another property, which is evident in the minidisk player, is the *natural inverse.* The volume up is achieved by twisting the knob in one direction and the opposite twist turns it down again.

This property is very important in two ways:

- *Automatic correction*—if you are trying to do something and 'overshoot', you automatically do the physically 'opposite' action. If this does not map to the logically opposite action you are likely to have problems.
- *Discoverability*—if you know that a pair of buttons always behave as natural inverses, then even if you do not know precisely what they do, you can feel free to experiment knowing you can correct the effect.

One of the authors used to have a phone with an up/down slider button. Sometimes it changed menu selections, and sometimes it controlled the volume. The author never learnt the precise rules for how this changed in each mode, but was still able to use it knowing that it was 'safe' to experiment.

Indeed in experiments (Ghazali 2007; Ghazali et al. 2015), we found that if we had completely arbitrary cognitive mappings (that is the user had no idea which controls affected which system property), but a good natural inverse, this performed much better than when there was a good cognitive map (the user knew precisely the effects of any action), but a poor natural inverse. In the first situation, users had to experiment slightly to work out which control to use, but once they did were able to complete tasks. In the second situation, things were fine initially as they knew which control to use, but if they overshot everything fell to pieces!

Give is also very important for discoverability, even to work out what physical manipulation is possible. Most lever-style door handles push down to open. However, when they do the opposite, that is you have to lift the handle to open, few people have serious problems. This is because having pushed slightly down and felt no 'give', the user is likely to automatically try to pull up slightly. The slight give tells you which directions are possible manipulations. This is a form of low-level and instinctive epistemic action (things you do in order to discover information).

In Fig. 9.9, we saw an example of the logical state changing autonomously. This is of course normal for computer systems! Less common in computers, but more so in domestic appliances, is when the system makes some change to the state of the controls themselves.

Electric kettles are a good example of this: when the water boils, the switch flicks to the off position.

Figure 9.10 shows a typical electric kettle. On the left is the physigram (ignore for a moment the dotted line). The user can press the switch down or up. There is an

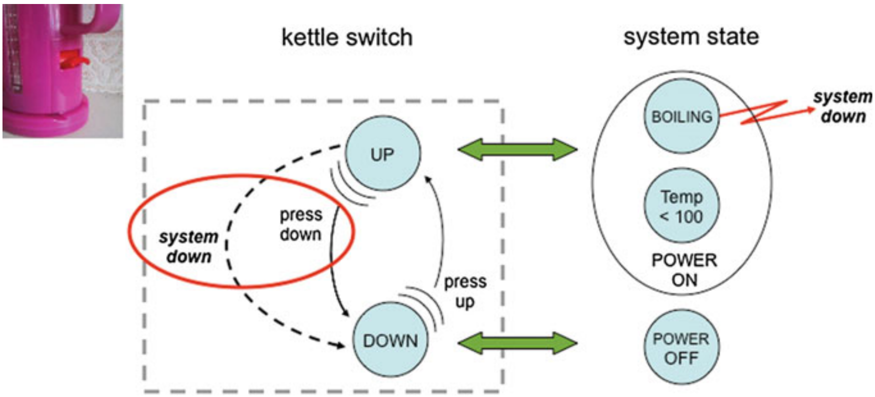


Fig. 9.10 Compliant interaction—matching system and user control

Fig. 9.11 Compliant interaction—the washing machine dial

exposed state relationship between the kettle switch and the power going to the kettle. In this case when the switch is up the kettle power is on, when it is down the power is off.

However, when the power is on and the water is boiling the sensor triggers the switch to flick back down into the power-off state. That is the kettle achieves both system control of the power and *exposed state*.

Not only that, but the way this is achieved means that if you, as the user, want to turn the kettle off, you do exactly the same action as the system does: a property we call *compliant interaction*.

Typically, when system and user control are aligned like this, it is easier to learn how to manipulate potentially complex system states. A good example of this is the traditional washing machine dial (Fig. 9.11). The user sets the desired programme using the dial, and then as the washing machine moves through its cycle it moves the dial. The dial therefore becomes an indicator of progress and also a means for the user to make tweaks, perhaps cutting a stage short. Sadly, this fluid expert behaviour has become more difficult and less common as washing machine control panels have become more digital.

On the other hand, where these properties fail we often see problems. Many electric toasters allow you to push down the handle to let the bread into the heat, but pop the toast up automatically. If you want to stop the toast early, there is often a special button. However, if in a hurry or flustered (maybe when the smoke comes out the top!), it is easy to forget and people often try to lift the toaster handle manually, leading to a few moments' struggle before the frantic search for the stop button. Other toasters effectively make lifting the handle serve this purpose, both a *natural inverse* to pushing it down and also *compliant interaction*: the automatic way to stop toasting and the manual way both involve the same control change.

9.8 Flexibility and Formality

Physigrams are a semi-formal notation. The STN core is amenable to analysis. However, in practice this is usually trivial; it is an STN of the physical device controls and manipulations: physical interactions between controls, and the combinatorial explosion this often causes, are rare. The real power of the physigram lies less in these aspects than in the differing styles of transitions, which enable subtle distinctions in behaviour to be expressed, for example, Figs. 9.3 versus 9.4 and 9.8 versus 9.9.

This more communicative power of physigrams was demonstrated very clearly when product designers used them as part of a design exercise for a media controller. They compared three different designs for a dial: two were physical movable dials, each with slightly different tactile properties, and one was a smooth touchpad-style dial.

Figure 9.12 shows the three physigrams produced by the designers, who had been shown examples of physigrams, but were not aided by the formal team. In some ways, rather like the comparison between Figs. 9.3 and 9.4, these are virtually identical, simply a number of control states that the control cycles between. However, when examined in detail, there are subtle differences: some controls allow full 360° movement (ii and iii), while one has a 'stop' (i), so that it is logically a linear control.

+Crucially, the designers also augmented the notation. One of the dials (Fig. 9.12i, close-up Fig. 9.13i) had hard selections, it clicked from one to another with a small amount of bounce-back when it was in one of them; another (Figs. 9.12ii and 9.13ii) moved fluidly but with tangible feel as the user went over the critical transitions; whilst for the third (Figs. 9.12iii and 9.13iii) there was only virtual feedback.

Strictly, the numbered states in (iii) are logical states only; the physigram of the physical interactions alone would only include the ability to move one's finger

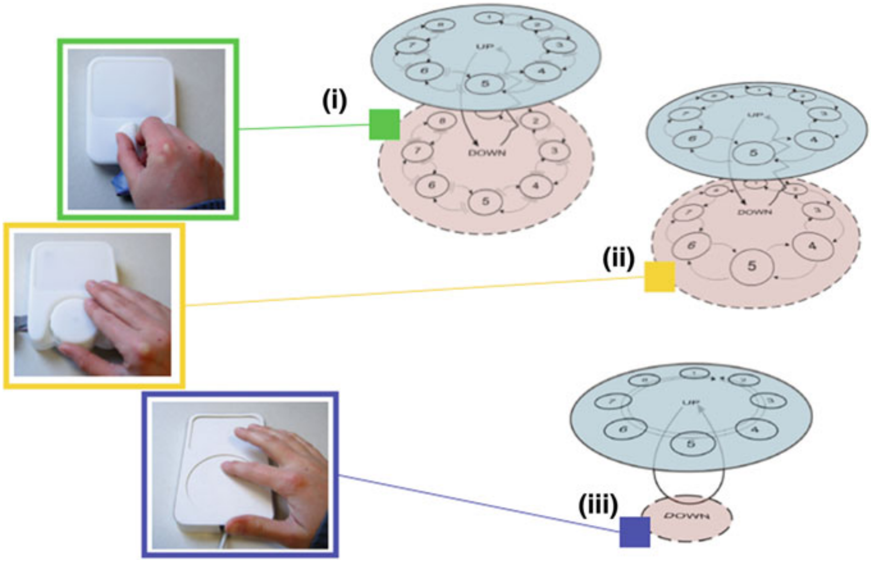


Fig. 9.12 Product designers' use of physigrams

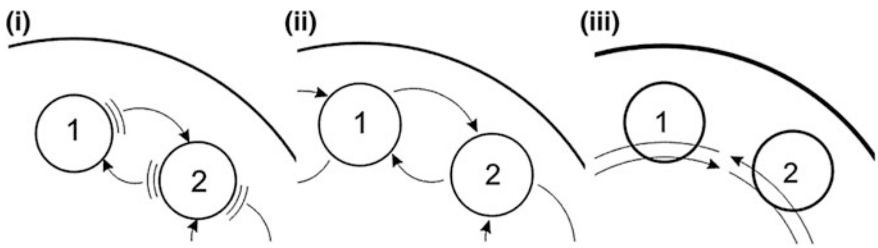


Fig. 9.13 Detail of transitions in Fig. 9.12

smoothly round the pad and press it down. However, neither did the numbered states denote the full controlled digital state, which might represent different menu selections depending on mode; instead, they were somewhere in between, capturing an early design decision that the device would always allow precisely 8 selections, but not the exact binding of these. Note that the line is doubled in Fig. 9.13iii, as the finger can slide over the touchpad in either direction.

This sort of touch-only device is common now with trackpads and smartphone displays. For these devices it is not that there is no tactile feedback, you can feel that you are in contact with the pad, and you can feel the movement through both touch and proprioceptive senses. The control in Fig. 9.12iii is a three-state device in Buxton's three-state model (Buxton 1990):

- State 0 when the finger is not in contact with the touchpad

- State 1 when the finger is in contact and dragging across the surface
- State 2 when the touchpad is pressed down

Figure 9.12iii shows a state 1–2 transition between the UP and DOWN states. In Fig. 9.12i, ii this is a bounce-back transition as the knobs noticeably press down and then click back into place when released (strictly this is a shorthand for lots of bounce-back transitions between the corresponding states as the dial stays in the same orientation when it bounces back). In contrast, the touchpad has a 'press to select' action, but it is not tangible except for the internal feeling of pressing something. The designers denoted this by showing the DOWN state as transient (there is a device 'down'), but with a loop transition drawn going from the UP state *through* the DOWN state and back to the UP state. This denotes, as in Fig. 9.13iii, that the user is not perceptually aware of the device state change.

Although there is tactile feedback through the internal sense of pressure for the touch device, the relation between the felt feedback and the system detection of a touch or movement is less clear than with the physical buttons.

It may be that felt touches are not registered by the system. For example, one of the authors has a TV with on-screen touch buttons; it can take many touches before it registers and turns on. This may be because he misses the 'hot spot' of the button or because the system 'de-bounces' the finger press, treating it as electrical noise.

Alternatively, it may be that the system registers events without the user being aware of it. For example, one of the authors cannot use a laptop with 'tap to select' enabled; when he types, his thumb occasionally makes contact with the touchpad, and although this is too light to feel it registers as a 'select' and the typing appears in apparently random parts of the screen.

For some purposes, the discrete version of the three-state model might be sufficient, but a real physical model to deal with touch would be even more difficult than those discussed for 'give' in Sect. 9.6. We will not attempt to deal with these issues here, but the physigram does act as a communication point with the potential, on the one hand, to drill down into the minutiae of body contact and pressure, and, on the other, to connect to system behaviour.

Note also that the designers drew the states in a form that visually matched the physical shape of the controller. The fact that the physical location of states in an STN does not have a formal interpretation left it open for the designers to create a meaning for this. This *openness to interpretation* is one of the design principles for appropriation identified elsewhere (Dix 2007) and effectively allows secondary notation (annotations that can be added which do not have formal meaning within the notation), which has been identified as an important feature in the study of cognitive dimensions of notations (Green and Petri 1996).

This openness of the notation is essential for physical devices because the range of possible interactions is wider than for screen-based controls. When using semi-formal notation for communication within a design team, it is more important that the meaning is clear to those involved than that they shoe-horn the behaviour into pre-defined but inappropriate categories. This also feeds back into more formal versions of the notation, as it highlights gaps (e.g. the 'half way logical' states).

On the other hand, if we wish to perform more formal analyses, some aspects have to be more strictly defined. In previous work, we have used a more precise meta-model to define the semantics of the various physigram primitives (Dix et al. 2009). A discrete meta-model was possible when describing the physical state physigrams (effectively an STN with coloured states and transitions). However, unsurprisingly, we found limitations when describing the more continuous interactions; a point encountered by the first author previously in studying status–event analysis (Dix 1991; Dix and Abowd 1996) and by others (Massink et al. 1999; Wüthrich 1999; Willans and Harrison 2000; Smith 2006). Ideally, we would also like to be able to model human aspects such as the level of pressure applied, and felt, as a finger presses a switch. We should be able to describe Fig. 9.6i, ii in such a way that we can verifiably say that one is syntactic sugar for the other. The only work that comes close to this is Eslambolchilar's (2006) work on cybernetic modelling of human–device interactions.

9.9 Case Study—Tilley, a Community Wind Turbine

We will look at the control panels described in the community wind turbine case study in Chap. 4, although the other book case studies would also include similar panels. The particular wind turbine, Tilley, is a land-based one, on the island of Tiree (TREL 2016), where, due to the windswept environment, it is one of the most efficient turbines of its type in the world.

Recall there are two control panels described in detail in Chap. 4 and reproduced in Fig. 9.14.

The 'digital' panel in Fig. 9.14a is mostly dedicated to outputs except for the generic function keys and numeric keypad in the middle. This will be used for the most complex, but not the most time critical, interactions. The individual membrane buttons have some *give*, and are each *bounce-back* as in Fig. 9.6. As they are for

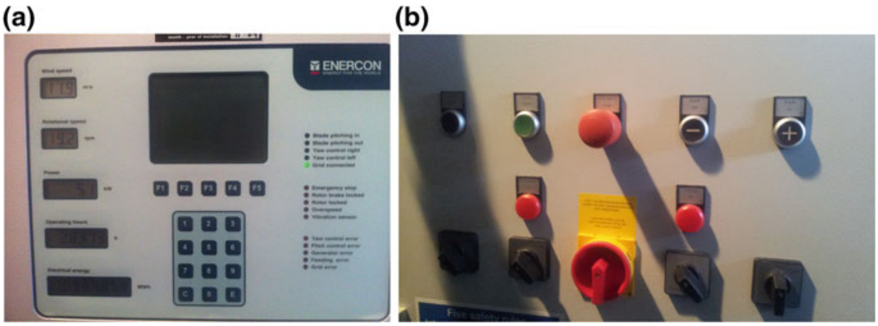


Fig. 9.14 **a** Digital display and control panel in Tilley (photo © William Simm), **b** physical control panel in Tilley (photo © Maria Angela Ferrario)

generic input this is a reasonable choice, so there is little more that physigrams can say about them. However, alternative analyses would be useful, for example the techniques applied in the CHI-MED project to medical numeric input devices (Cauchi et al. 2014). We should note also that the numeric keypad somewhat oddly adopts a phone-style key order (with 1 at the top) rather than the order found in calculators or computer numeric keypads (with 1 at the bottom).

The panel in Fig. 9.14b is more interesting, with a combination of press switches and twist knobs of various kinds. Of course it is not possible to 'unplug' Tilley to experiment in the way described for domestic devices earlier in this chapter; neither is it possible to experiment with the live panel to get the 'feel' of the buttons, so the analysis here is inevitably limited. However, during the design of an interface such as this, detailed physical examination and specification would be possible.

Note that this is an example that could benefit from the kinds of pressure annotations discussed towards the end of Sect. 9.6. The emergency stop button (large red button, top centre in Fig. 9.14b) needs to be firm enough not to be pressed accidentally, but also responsive enough that it can be pressed quickly and that you know you have pressed it successfully from its physical response. Note too that this is large, as it, especially, needs to be operated easily even if the operator is wearing gloves. This button is a *bounce-back button* and this is appropriate from a system control point of view as the restart is expected to be a complex process, not simply pulling the button out again (see Fig. 9.15). However, as a *hidden state* control it does not have feedback of type A (Fig. 9.1). Instead, the operator would either need to look at numeric outputs on the digital display panel or screen (feedback loop C), or more likely simply be aware of the physical effect (loop D) as the system shuts down.

The reset button (green to the left of the emergency stop button in Fig. 9.14b) is also a bounce-back, although of a slightly different design as it does not need to be

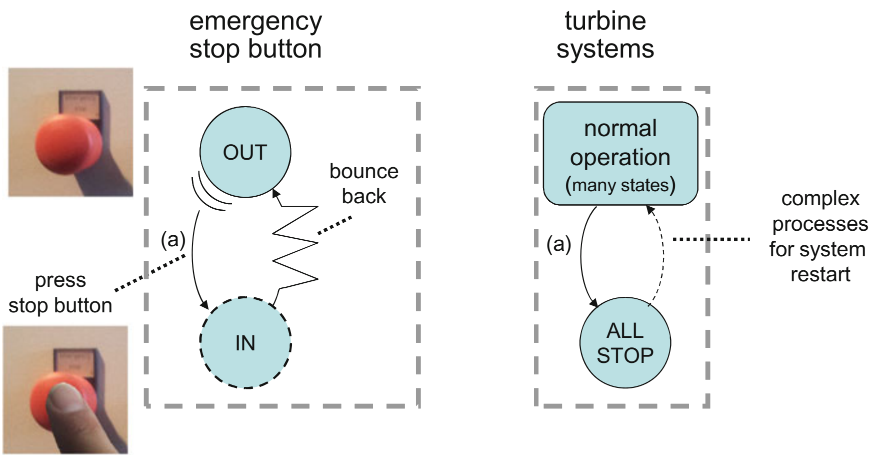


Fig. 9.15 Emergency stop button (*left*) physigram (*right*) system state

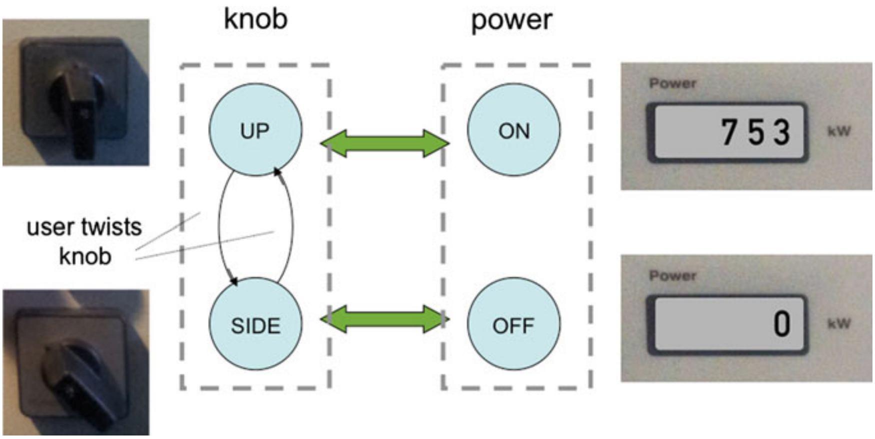


Fig. 9.16 Power isolation knobs (*left*) physigram (*right*) system state

hit quickly like the emergency stop button. It too has hidden state, presumably resetting various electrical systems. The effects of this may be less obvious than the emergency stop and might require explicit checking of the digital display. However, as it is a reset, it is *idempotent*, meaning that if the operator is at all uncertain whether it was pressed, they can simply press it again.

Below the emergency stop button is a large power control knob flanked by several small black knobs to control specific circuits. These are all visible state controls with one-to-one mappings between physical state and controlled system state (see Fig. 9.16). The settings of these are critical: if the engineer wrongly thinks a circuit is off and touches it, they may be injured or killed. From the image we can immediately see that the main control is switched on, as are three of the sub-circuits, but the one on the right is off.

At the top right are two buttons labelled '+' and '–'. These control the angle of attack of the turbine blades. These are each *bounce-back buttons* and serve to increase or decrease the current blade angle (see Fig. 9.17). These are *hidden state* controls and the engineer would either need to go outside to look at the blades (loop D), or more likely simply observe the impact of the change in angle on power output.

One could imagine an alternative control that used a dial to adjust the blade angle. This would lead to a visible state and also be faster to set a particular angle. However, the speed is probably immaterial; the engineer has taken a three-day trip to come to the island, a few seconds pressing ± buttons is not going to make much difference! Also, CHI-MED has shown that often this form of increment/decrement setting is safer as it makes it harder to perform gross errors (Oladimeji et al. 2011).

However, the positioning of the buttons is not optimal. They are clearly placed in a neat grid, but this means there is nothing to suggest physically or visibly that the buttons are linked. Unlike the minidisk volume controls, this does not form a *natural inverse*.

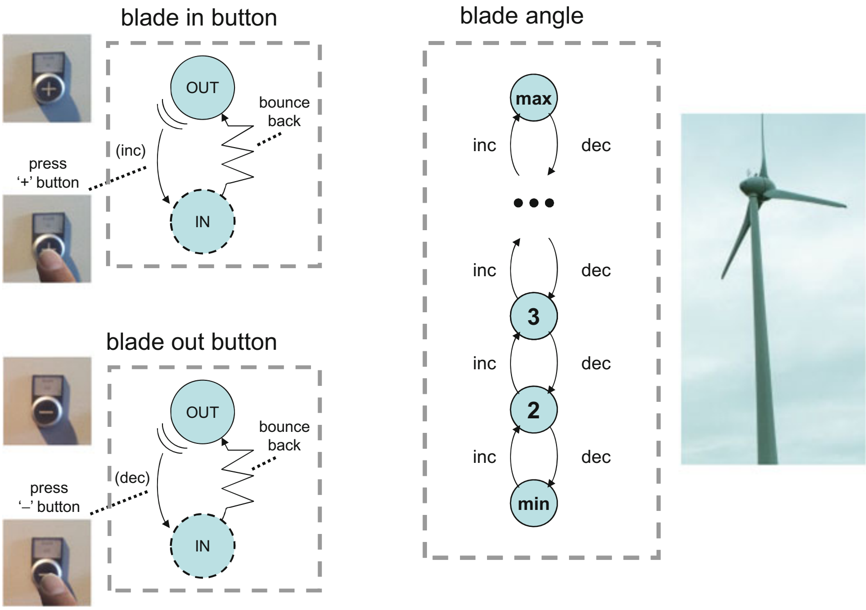


Fig. 9.17 Blade angle control (*left*) physigram (*right*) system state

9.10 Conclusions

We have seen how physigrams allow us:

(i) to describe the behaviour of the device 'unplugged', the purely physical interaction potential of a device, enabling the exploration of sensory-motor affordances of the physical device independent of how it is connected into the wider system.

(ii) to link these subsequently to digital states, exposing a variety of properties of physical–digital hybrid systems, related to but going beyond conventional discussions of representational 'mapping'.

The first of these often exposes subtle differences between controls that otherwise appear superficially similar. The second both allows specific issues to be identified and also offers the potential to create generic advice on matching physical controls and digital state.

In addition to exposing these generic properties, we have seen how physigrams can be used by product designers to describe aspects of their physical designs that would otherwise only be apparent during physical exploration. This can both help their own design activity and also allow clearer discussions with developers or more formal analysis.

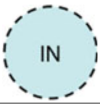
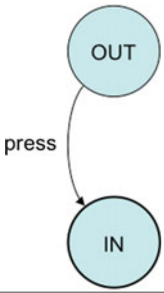
The semi-formal nature of physigrams means that we can specify precise formal semantics for a core set of primitives, whilst still allowing an openness to extend and augment the notation for new circumstances. Marrying these two, flexibility for extension and formal tractability, is still an open issue, not just for physigrams, but for any formal notation.

Both the informal and formal uses of physigrams raise again the still open issue of how to create comprehensible and tractable models of ‘hybrid’ interactive systems that combine both continuous (status) and discrete (event) behaviours. In particular, these would ideally also be able to talk about the physical actions and sensations of the user as well as more common mental states.

Finally, physigrams were used to explore the specification of the control panel of a medium sized wind turbine. There were limitations to this as the system could not be brought out of operation simply for the purposes of the study; so some aspects of the behaviour had to be guessed. However, the case study gives some idea of how the methods could be applied during the design stages of this kind of industrial control panel.

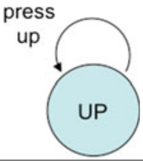
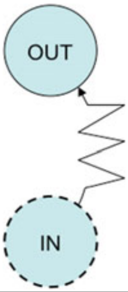
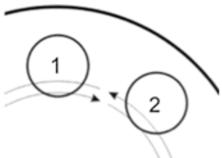
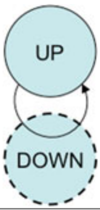
9.11 Key to Notation

Physigrams

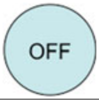
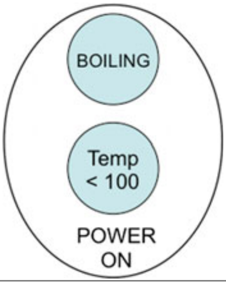
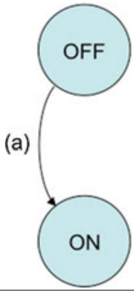
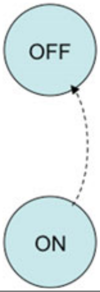
Symbol	Meaning	Used in Figs
	**state**—physical state of the device	3–10, 12, 13, 15–17
	**transient tension state**—physical state which can only be maintained by some sort of continuous user pressure or exertion	4–6, 9, 12, 15, 17
	**transition**—this may be labelled by the user action that causes this, or this may be implicit. It may also have a label to connect it with logical state (see linkage)	3–9, 12, 13, 16

(continued)

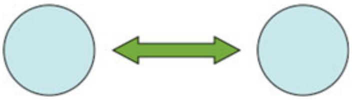
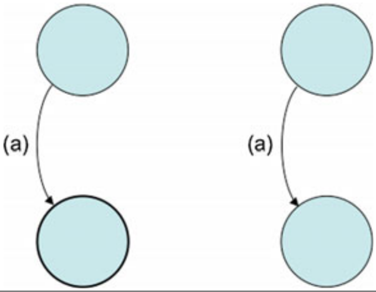
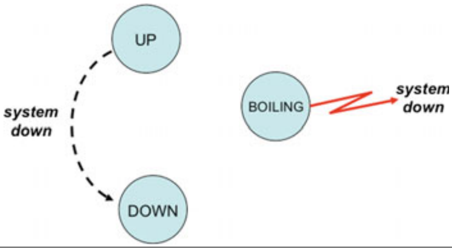
(continued)

Symbol	Meaning	Used in Figs
	**self transition**—special case of the above where a user action has no effect on the system (e.g. attempting to twist a dial beyond its limits)	6
	**bounce-back**—when the device spontaneously returns from a transient tension state to a stable state when user pressure is removed	4–6, 8, 9, 12, 15, 17
	**give**—where a button or other control moves slightly but with resistance before 'giving' and causing state change. If the user stops exerting pressure before the 'give' point, it will return to the initial state	6, 10, 12, 13, 15, 17
	**slide transition**—designer extension to denote situation when there are device changes that are not perceptible to the user	6, 10, 12, 13
	**unfelt bounce-back**—this is basically a press and bounce-back, but where there is no perceptual feedback	12

Logical State

Symbol	Meaning	Used in figs
	**state**—state of the logical system	7–10, 16, 17
	**group**—where several lower-level states can be thought of as sub-states of a more abstract state. In the example, 'POWER ON' is a higher-level state, but within this water may be below or at boiling point	10
	**user–initiated state transition**—logical state transition initiated by external user activity (see also linkage)	8, 9, 17
	**system-initiated transition**—where some internal computation, or environmental event not connected with the interacting user, causes a transition in the logical system	9

Linkage

Symbols physigram logical state	Meaning	Used in figs
	**state–state mapping** –where the physically visible or tangible states in an *exposed state* device correspond precisely to states in the underlying logical system	7, 10, 16
	**user-initiated transition**—where a state transition in the physical device caused by a user action gives rise to a transition in the logical state. The connection is denoted by the label, '(a)' in the example	8, 9
	**system–initiated transition**—where a spontaneous change or event in the system (denoted by the lightning symbol) triggers a physical state change in the device	10

Acknowledgements Early parts of this work were supported by the AHRC/EPSRC funded project DEPtH "Designing for Physicality" (http://www.physicality.org/). We are very grateful for insightful comments from the reviewers.

References

Buxton W (1990) A three-state model of graphical input. In: Proceedings of human–computer interaction—INTERACT'90. Elsevier, Amsterdam, pp 449–456

Cauchi A, Oladimeji P, Niezen G, Thimbleby H (2014) Triangulating empirical and analytic techniques for improving number entry user interfaces. In: Proceedings of the 2014 ACM SIGCHI symposium on engineering interactive computing systems (EICS '14), ACM, NY, USA, 243–252. doi:10.1145/2607023.2607025

Dix A (1991) Formal methods for interactive systems. Academic Press, New York. http://www.hiraeth.com/books/formal/

Dix A, Abowd G (1996) Modelling status and event behaviour of interactive systems. Softw Eng J 11(6):334–346. http://www.hcibook.com/alan/papers/SEJ96-s+e/

Dix A, Finlay J, Abowd G, Beale R (2004) Human–computer interaction, 3rd edn. Prentice Hall, Englewood Cliffs. http://www.hcibook.com/e3/

Dix A (2007) Designing for appropriation. In: Proceedings of BCS HCI 2007, People and computers XXI, vol 2. BCS eWiC. http://www.bcs.org/server.php?show=ConWebDoc.13347

Dix A, Ghazali M, Gill S, Hare J, Ramduny-Ellis S (2009) Physigrams: modelling devices for natural interaction. Formal Aspects Comput, Springer 21(6):613–641

Eslambolchilar P (2006) Making sense of interaction using a model-based approach. PhD thesis, Hamilton Institute, National University of Ireland, NUIM, Ireland

Gaver W (1991) Technology affordances. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI'91). ACM Press, New York, pp 79–84

Ghazali M (2007) Discovering physical visceral qualities for natural interaction. PhD thesis, Lancaster University, England, UK

Ghazali M, Dix A, Gilleade K (2015) The relationship of physicality and its underlying mapping. ARPN J Eng Appl Sci 10(23):18095–18103

Gibson J (1979) The ecological approach to visual perception. Houghton Mifflin Company, USA

Green T, Petri M (1996) Usability analysis of visual programming environments: a 'cognitive dimensions' framework. J Vis Languages Comput 7:131–174

Harel D (1987) Statecharts: a visual formalism for complex systems. Sci Comput Program 8 (3):231–274. doi:10.1016/0167-6423(87)90035-9

Massink M, Duke D, Smith S (1999) Towards hybrid interface specification for virtual environments. In: DSV-IS 1999 design, specification and verification of interactive systems, Springer, Berlin, pp 30–51

Norman D (1988) The psychology of everyday things. Basic Books, New York

Norman D (1999) Affordance, conventions, and design. Interactions, 6(3):38–43. ACM Press: NY

Oladimeji P, Thimbleby H, Cox A (2011) Number entry interfaces and their effects on error detection. In: IFIP conference on human-computer interaction, Springer, Heidelberg

Pfaff G, ten Hagen P (eds) (1985) Seeheim workshop on user interface management systems. Springer, Berlin

Shneiderman B, Plaisant C (2010) Designing the user interface: strategies for effective human-computer interaction, 5th edn. Addison-Wesley, MA

Smith S (2006) Exploring the specification of haptic interaction. In Interactive systems: design, specification and verification (DSVIS 2006). Lecture notes in computer science, vol 4323. Springer, Berlin, pp 171–184

Thimbleby H (2007) Using the fitts law with state transition systems to find optimal task timings. In: Proceedings of second international workshop on formal methods for interactive systems, FMIS2007. http://www.dcs.qmul.ac.uk/research/imc/hum/fmis2007/preproceedings/FMIS2007preproceedings.pdf

TREL (2016) Tilley, Our turbine, Tiree Renewable Energy Limited. http://www.tireerenewableenergy.co.uk/index.php/tilley-our-turbine/. Accessed 24 Mar 2016

Willans J, Harrison M (2000) Verifying the behaviour of virtual world objects. In: Proceedings of DSV-IS'2000. Springer, Berlin, pp 65–77

Wüthrich C (1999) An analysis and model of 3D interaction methods and devices for virtual reality. In: Proceedings of DSV-IS'99. Springer, Berlin, pp 18–29

Zhou W, Reisinger J, Peer A, Hirche S (2014) Interaction-based dynamic measurement of haptic characteristics of control elements. In: Auvray M, Duriez C (eds) Haptics: neuroscience, devices, modeling, and applications: 9th international conference, EuroHaptics 2014, Versailles, France, June 24–26, 2014, Proceedings, Part I, pp 177–184. Springer, Berlin. doi:10.1007/978-3-662-44193-0_23