

Crossing the Bridge over Norman's Gulf of Execution: Revealing Feedforward's True Identity

Jo Vermeulen¹ Kris Luyten¹ Elise van den Hoven^{2,3} Karin Coninx¹



¹Hasselt University – tUL – iMinds,
Expertise Centre for Digital Media, Diepenbeek, Belgium

²University of Technology, Sydney, Australia,
Design, Architecture & Building Faculty

³Eindhoven University of Technology, Eindhoven, The Netherlands,
Industrial Design Department

¹{jo.vermeulen, kris.luyten, karin.coninx}@uhasselt.be

²elise.vandenhoven@uts.edu.au

ABSTRACT

Feedback and affordances are two of the most well-known principles in interaction design. Unfortunately, the related and equally important notion of *feedforward* has not been given as much consideration. Nevertheless, feedforward is a powerful design principle for bridging Norman's Gulf of Execution. We reframe feedforward by disambiguating it from related design principles such as feedback and perceived affordances, and identify new classes of feedforward. In addition, we present a reference framework that provides a means for designers to explore and recognize different opportunities for feedforward.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces

Author Keywords

feedforward; affordances; feedback; design; theory

INTRODUCTION

The notions of *feedback* and (*perceived*) *affordances* [22]¹ have emerged over the past few decades as core concepts in interaction design. While these principles are generally recognized and well-known among designers, the same is not true for the related notion of *feedforward*. The term was first

¹Norman later argued for replacing the term “perceived affordances” with “signifiers” to avoid confusion [24]. However, for historical relevance and to accurately reflect what others have written about the relation between feedforward and perceived affordances, we will continue to use “perceived affordance” throughout this paper. All mentions of this term can, however, be replaced with “signifier”.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

Copyright 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

introduced in this context in the HCI community by Djajaningrat et al. in 2002 [6]. Well-designed feedforward is an effective tool for bridging Norman's Gulf of Execution [22] – the difference between the user's intentions and the allowable actions – as it *tells users what the result of their action will be*. This is illustrated in Figure 1. A simple example of feedforward is a label on a button: the label tells users what happens if they push the button. For instance, the design of the iPhone's lock screen combines perceived affordances and feedforward to tell users how they can unlock the phone, as shown in Figure 2. However, feedforward also exists in many other forms and guises.

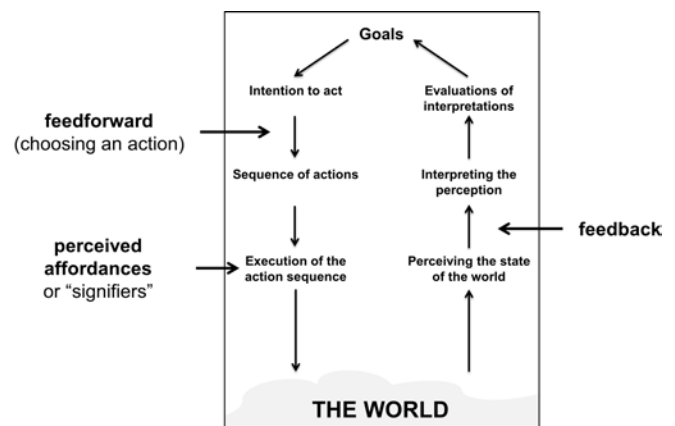


Figure 1. The position of perceived affordances (or signifiers [24]), feedforward and feedback in Norman's Stages of Action model (image based on [22]).

Even though few designers are familiar with the term feedforward, almost every designer has – in one way or another – already used feedforward in their designs. Yet, opportunities for feedforward are often left unexplored because designers are not fully aware of this design aspect. An important factor causing this limited awareness is the lack of a well-defined and generally accepted meaning of feedforward. Additionally, there is no existing library of examples and proven so-



Figure 2. A well-designed combination of feedforward and perceived affordances in the iPhone lock screen. The “slide to unlock” label, the slider and its button with an arrow icon, and the subtle animation of light that moves over the slider indicates what users can do to unlock the phone. Feedforward tells users what they can expect when they execute an action – in this case, that the phone will be unlocked when they move the slider to the right. Given their desired goal (unlocking the phone so that it can be used), they can select the appropriate action corresponding to that goal.

lutions for applying feedforward when designing new interfaces. With this paper, we want to define feedforward more clearly. In addition, we provide a reference framework for designers that differentiates feedforward from other design principles such as affordances and feedback.

As our everyday environments and devices become smarter and more sophisticated, it will become more difficult to convey to users what tasks are supported, what can be accomplished and how users can achieve their goals. Bellotti et al. [2] argue that, unlike designers of GUI interfaces, designers of “sensing systems” – which include context-aware systems, gestural interfaces, tangible interaction and post-WIMP user interfaces [26] in general – have few pre-packaged answers available to solve basic interaction design challenges. For example, it has been suggested that context-aware systems should provide support for *intelligibility* [3] – presenting to users what the systems know, how they know it, and what they are doing. Similarly, while one of the strengths of tangible interaction is the fact that physical objects can communicate their purpose through their form and might therefore be easier for users to understand, Hornecker and Buur [14] warn against these simple, direct associations. They argue that if tangible interaction is to become useful for complex domains and to scale up to real-world size examples, balancing legibility and computational power is one of the grand challenges in the field. We believe that feedforward is an important design concept to consider for such systems which will help users in knowing what they can expect and thereby bridge Norman’s Gulf of Execution. As recently argued by Norman [21], we should not try to avoid complexity, but rather tame complexity through good design.

In summary, the contributions we present in this paper are twofold:

- We use Hartson’s four types of affordances [12] to distinguish feedforward from related design principles such as feedback and affordances and clarify their relationship. By doing so, we raise awareness of feedforward and clear up some of the confusion surrounding the term.
- We describe a reference framework for designers that allows them to explore and recognize different opportunities for feedforward.

BACKGROUND

Although the term feedforward has been used in other domains such as control theory (*e.g.*, [20]), this paper focuses on the usage of the term feedforward in the context of interaction design. The first definition of feedforward in this context was given by Djajadiningrat et al. [6, p. 285]. They have defined feedforward by disambiguating it from related concepts such as feedback and perceived affordances [emphasis ours]:

We distinguish between information before the user carries out the action (pre-action), and after the user carries out the action (post-action). These phases correspond with feedforward and feedback. *Feedforward informs the user about what the result of his action will be.* Inviting the appropriate action is a prerequisite for feedforward but it is not sufficient. The product also needs to communicate what the user can expect. Feedback informs the user about the action that is carried out, shows that the product is responding, indicates progress, confirms navigation, etc.

Note that, unlike feedforward, perceived affordances do not communicate the purpose of an action.

We can situate perceived affordances, feedforward and feedback within Norman’s Stages of Action model [22], as shown in Figure 1. Feedback bridges Norman’s Gulf of Evaluation [22] – the amount of effort users must exert to interpret the state of the system and to determine how well the expectations and intentions have been met – by helping users evaluate the state of the system. When evaluating the state of the world, users go through the Stages of Evaluation shown on the right side of Figure 1. Feedforward, on the other hand, bridges Norman’s Gulf of Execution [22] – the difference between the user’s intentions and the allowable actions – by helping users decide what action to take based on that action’s expected outcome. When users act on a certain goal, they go through the Stages of Execution seen on the left side of Figure 1. Perceived affordances are also used for bridging Norman’s Gulf of Execution, but serve a different purpose: they suggest a particular action to users, such as pressing a button, or turning a knob.

It has been argued that Norman’s perceived affordances can be viewed in the context of semiotics [4]: an object’s appearance (*e.g.*, the shape of a button) *signifies* to the user that this object can be pressed. A similar argument could be made for feedforward in the sense that it signifies to the user what they can expect when performing a certain action.

In the remainder of this paper we will explore feedforward in depth. We start with current use of the term in practice which is at times inconsistent and thus illustrates the need for a clear and generally accepted definition. Next, we discuss other definitions of feedforward to develop our own reframing of feedforward in which we further detail its relation with perceived affordances and feedback. Finally, we give an overview of what the different definitions cover in terms of feedforward and analyze a few notable examples of feedforward.

USE OF FEEDFORWARD

Djajadiningrat et al. [6] mostly focus on the importance of feedforward for tangible interaction. However, there have also been other application domains in which feedforward has been successfully applied, such as gestural interaction. A common problem of gestural interfaces is their lack of visibility: users lack awareness of the available gestures that are recognized by the system, and what these gestures do. Feedforward can help users in performing the correct gesture by telling them what will happen when a certain gesture is invoked. An early example of the use of feedforward in gestural interaction is Kurtenbach et al.'s concept of marking menus [17]. Marking menus are pie menus – circular menu's that support gestural interaction as shown in Figure 3(a) – intended to accommodate both novice and expert users. They allow a user to perform a menu selection by either popping-up a pie menu, or by making a straight mark in the direction of the desired menu item without popping-up the menu. The pie menu serves as a feedforward display that helps novice users who hesitate when they are unsure of a gesture or command, as shown in Figure 3(a). When users become more experienced, they tend to use marks more, although they still look at the feedforward display once in a while to refresh their memory [16]. Note that marking menus show both the available gestures (or actions) and what users can expect when they perform one of these gestures.

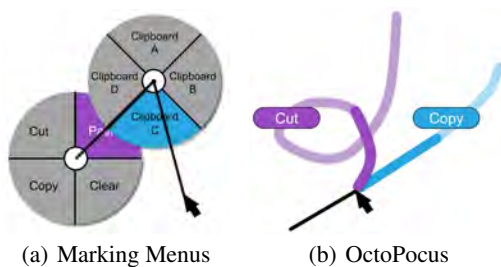


Figure 3. Examples of feedforward in gestural interaction (images based on [17] and [1]).

Bau and Mackay developed OctoPocus [1], a dynamic guide that combines on-screen feedforward and feedback to help users learn, remember and execute gestures. They state that “feedforward mechanisms provide information about a gesture's shape and its association with a particular command, prior to the execution or completion of the gesture.” [1]. Like marking menus, OctoPocus takes advantage of possible hesitation by appearing after a “press and wait gesture”. The

set of possible gestures and associated commands are continuously updated while the user is performing a certain gesture (Figure 3(b)). Another example of feedforward for gestural interaction is ShadowGuides [8], which extends Bau and Mackay's concept of dynamic guides to multi-touch and whole-hand gestures.

Additionally, feedforward has also been deemed important for context-aware computing. Context-aware systems [5] typically act based on implicit input collected from the environment. Additionally, system actions are usually a result of complex reasoning about context data which might be hard for users to understand and makes the system unpredictable [3]. Bellotti and Edwards [3] argue that context-aware systems should be *intelligible* and inform end-users of their capabilities and current understanding. One of the proposed design principles for realizing this is feedback which includes *feedforward* – an answer to the question “What will happen if I do this?” [3]. Although Bellotti and Edwards adhere to Djajadiningrat et al.'s [6] basic idea of feedforward – communicating the *purpose* or *result* of an action – they view it as a form of feedback, which is not in line with Djajadiningrat et al.'s ideas. They also provide examples of feedforward in WIMP GUIs that are often taken for granted, but as they argue, are necessary components of the interface that help users know what will happen when a certain action is performed: flashing insertion points; cursors, pointers and handles; window highlighting; and rubberbanding. Lim and Dey have developed a toolkit [18] to support “What if?” questions in context-aware systems. This can be seen as an example of the kind of feedforward that Bellotti and Edwards refer to.

By comparing Djajadiningrat et al.'s [6] original definition of feedforward to the interpretations by Bellotti and Edwards [3] and by Bau and Mackay [1], it is evident that the term feedforward has not always been used consistently. Different communities seem to interpret the concept in a different way. There are only a handful of HCI textbooks that talk explicitly about feedforward. One of them is “Designing for Interaction” [25] by Dan Saffer, which refers to the definition by Djajadiningrat. Saffer argues that designers should look out for opportunities to use feedforward – even though he states that it is harder to design into products and services than feedback [25, p. 133].

FEEDFORWARD DEFINITIONS

In this section, we outline the differences between existing definitions of feedforward. This overview will provide a thorough review on the notion of feedforward and its relation with affordances. The table in Figure 4 lists different definitions and examples of feedforward alongside a number of important dimensions. We will discuss this table in more detail later.

Djajadiningrat – Going Beyond Affordances

As previously mentioned, Djajadiningrat et al. [6] have defined feedforward by contrasting it with feedback and affordances. Feedback is one of the most well-known design principles in interaction design, along with affordances, visibil-

Modality			Detail			Hierarchy	Time		
Visual	Tactile	Aural	High	Average	Low	Nested	Static	Sequential	
								Discrete	Continuous
Definitions									
Djajadiningrat	✓	✓			✓		✓		
Wensveen	✓	✓	✓	✓	✓	✓	✓		
Gaver / McGrenere and Ho	✓	✓	✓			✓	✓	✓	
Hartson	✓					✓	✓		
Norman	✓				✓		✓		
Bau and Mackay	✓			✓					✓
Examples									
OctoPocus / ShadowGuides	✓			✓					✓
SpeakCup	✓				✓			✓	
Disney Pixar Cars 2 AppMATes	✓				✓	✓			
Tooltips	✓			✓			✓		
Tangible Video Editor	✓			✓		✓	✓	✓	
TempSticks	✓	✓			✓	✓	✓		

Figure 4. Summary of the coverage of the feedforward definitions, their differences and an analysis of several feedforward examples in practice.

ity, constraints and mappings. Feedback is a message about whether or not a goal was achieved or maintained [25] and is typically used to inform the user that the system is responding, to indicate progress or to confirm navigation [3]. Djajadiningrat et al. [6] state that feedforward, like feedback, returns information about the *result* of a process or activity. However, while feedback is communicated *during* or *after* the action, feedforward is information that is offered *before* the action takes place. Whereas feedback informs the user about the action that is carried out, feedforward informs the user about *what the result of their action will be*.

Next to feedforward, affordances also provide information to users before they carry out an action. Gibson [10, 11] defined affordances as:

All “action possibilities” latent in the environment, objectively measurable and independent of the individual’s ability to recognize them, but always in relation to the actor and therefore dependent on their capabilities.

As introduced in the HCI literature by Norman [22], perceived affordances (affordances that the user is made aware of through good design) essentially *invite the user to a particular action*. Affordances therefore “suggest” how one can interact with a product or system. Typical examples in HCI are buttons which “afford” pushing, knobs which “afford” turning, or sliders which “afford” moving up and down (or left and right, depending on the orientation). Affording the right actions has been widely regarded as a crucial aspect of usability. Even though perceived affordances are very useful, Djajadiningrat et al. [6] argue that inviting the appropriate action is a prerequisite for feedforward, but it is not sufficient. They state that the essence of usability lies not in communicating the necessary action, but the *purpose* of an action.

Wensveen – Inherent, Augmented & Functional Feedforward

Wensveen, Djajadiningrat and Overbeeke [29] further elaborated on their previous definition and distinguish between three different types of feedforward, based on the “form of information” that the user receives about their action: **inherent**, **augmented** and **functional** feedforward.

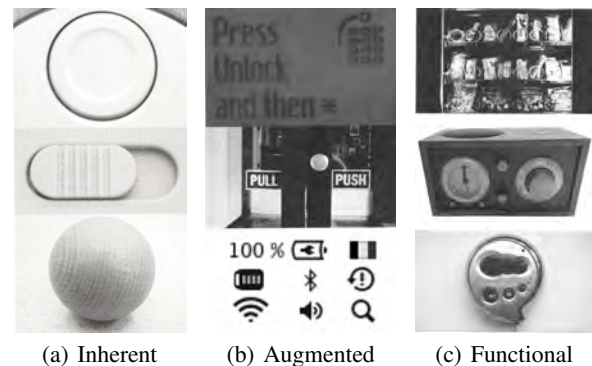


Figure 5. Wensveen’s three types of feedforward. Images from [27] reused with permission (Copyright © 2005 Stephan Wensveen).

Inherent feedforward offers information related to the action possibilities of the product and appeals primarily to the *perceptual motor skills* of the person. Inherent feedforward communicates what kind of action is possible (*e.g.*, pushing, sliding, rolling; see Figure 5(a)) and how this action can be carried out (the amount of force required, body parts, etc.). Wensveen [27] states that inherent feedforward can be viewed as a limited interpretation of Gibson’s affordances [10]. Note that Wensveen does not contradict Djajadiningrat et al. [6],

who did not yet differentiate between different types of feedforward. Feedforward as a whole – the combination of inherent, augmented and functional feedforward – still goes beyond affordances, according to Wensveen [28]. The differentiation between the three different types of feedforward was mainly made for analysis purposes [28].

Augmented feedforward is information *from an additional source* about the *action possibilities* of a product or system, or the *purpose* of these action possibilities [27]. This type of feedforward appeals primarily to the *cognitive skills* of users. Figure 5(b) shows examples of augmented feedforward, such as on-screen messages indicating what to do (*i.e.*, conveying the action possibilities) and lexical or graphical *labels* communicating the purpose of the action possibility.

Functional feedforward goes beyond the action possibilities and their specific purpose and instead informs the user about the *more general purpose of a product and its functional features* [27]. A possible strategy for functional feedforward is making the functional parts visible (*i.e.*, corresponding to Norman’s notion of *visibility* [22]), as illustrated for example by the candy vending machine in Figure 5(c) (top), where the available types of candy and the mechanism which delivers the products to the user are clearly visible.

Even though the above-mentioned definitions distinguish between feedforward and affordances, there have been a number of frameworks for affordances (*e.g.*, [9, 12, 15, 19]) that included aspects of feedforward without explicitly mentioning the term, further adding to the confusion. In the next sections, we give an overview of these frameworks and explain how they relate to feedforward.

Gaver – Technology Affordances

In his paper titled “Technology Affordances”, Gaver [9] argues that affordances are not always single, independent entities, but can be related to one another. He describes two different relationships between affordances: *nesting* and *sequence*. Nested affordances are grouped in space, while sequential affordances are sequential in time (*i.e.*, acting on an affordance leads to information indicating new affordances). Gaver argues that affordances are not passively perceived, but explored [9]. He also hints at the possibility of conveying affordances through different modalities (*e.g.*, sound, tactile information).

Nested affordances, in particular, bear resemblance to feedforward. McGrenere and Ho [19] have analyzed Gaver’s work, and clarify *nested* affordances with the example of a button. They state that users are not interested in clicking on a button for its own sake, but are interested in *invoking a certain function*. The function that will be invoked by a button is usually specified through its label or icon. They explain that here the affordance of “button clickability” is nested within the affordance of “function invocability”. McGrenere and Ho stress that it is important to acknowledge that each of the levels of the affordance hierarchy may or may not *map to system functions*. They further argue that affordances are not limited to physical aspects of the system (*e.g.*, physical interaction with a mouse, keyboard or screen), as implied by Norman in

his clarification of the use of the term affordances [23]. They state that application software also provides possible actions. For example, a word processor affords writing and editing at a high level, but also actions such as clicking and scrolling.

Given Gaver’s extension of nested affordances, Wensveen’s functional feedforward [27] could be seen as a perceptible affordance [9] which conveys the general (top-level) function of a system – or what the system affords the user. This top-level affordance can be seen as the root of a hierarchy of affordances. As an example, in Figure 5(c) (bottom), the shape of the voice recorder conveys its general function to the user. However, to actually record speech, users will also have to be aware of the nested functions and affordances for these functions (*e.g.*, the record button). One could argue that through the concept of functional feedforward, Wensveen implicitly suggests that feedforward might also be nested, just like Gaver’s affordances.

Gaver also introduced *sequential* affordances which are only available at certain points in time. This is common in graphical user interfaces since these can be updated during usage. In contrast, physical objects usually have a static physical appearance and cannot update their form over time. The information that specifies an affordance (*e.g.*, a button on the screen), can be quickly updated as new affordances become available (*e.g.*, adding a drop down menu when the button is clicked to allow the user to make a selection) [9]. Similarly, feedforward could only be made available at certain points in time or be updated during the user’s action to provide new information. The examples of feedforward in gestural interaction that were discussed previously (marking menus [17] and OctoPocus [1]), can be seen as examples of sequential feedforward. Sequential feedforward in combination with feedback could further blur the difference between the two concepts. Feedback provided after performing an action might afterwards serve as feedforward for the action that logically follows the previous one.

Kaptelinin and Nardi – Mediated Action & Affordances

Kaptelinin and Nardi [15] call for adopting a mediated-action perspective on technology affordances as an alternative for Gibson’s ecological psychology perspective. They differentiate between two types of affordances: *instrumental technology affordances* which are comprised of a *handling affordance* and an *effector affordance*; and a set of *auxiliary technological affordances* such as maintenance, aggregation and learning affordances. We mainly focus on instrumental technology affordances – and in particular **effector affordances** – here, as these appear to be quite similar to feedforward. The difference between handling and effector affordances is explained with the example of a knife. A knife consists of two distinct parts: the handle and the blade. The knife handle is used for holding the knife, while the blade is used to manipulate objects (*e.g.*, an apple). Kaptelinin and Nardi [15] argue that this distinction also applies to digital technologies, and, more specifically, to user interface widgets. For example, the ability to drag the scroll box of a scroll bar is the handling affordance, while the ability to display a certain portion of the document in the window is the effector affordance.

Like Djajadiningrat [6] and Wensveen [27], Kaptelinin and Nardi [15] distinguish between the purpose of an action (the effector affordance) and the action possibility (the handling affordance). Indeed, they provide an example of a dialog where: “handling affordances are clear but the outcomes of user actions (effector affordances) are not immediately obvious. [...] The user can see that they can select the checkboxes and click the buttons, but the effects of these actions are not directly apparent.” They state that users will be confused when handling and effector affordances are not coupled tightly enough. Kaptelinin and Nardi’s effector affordances appear to be closely related to feedforward since they convey the outcome of a certain action. The idea of tightly integrated handling and effector affordances seems to be similar to how perceived affordances and feedforward can be combined to communicate both the action possibilities and the expected outcomes of those actions.

Hartson – Feedforward as a Cognitive Affordance

Hartson [12] further clarified the concept of affordances based on Gaver’s [9] and McGrenere and Ho’s [19] work. He distinguishes between four types of affordances based on the role they play in supporting users during interaction:

Cognitive affordances are considered to be an extension of Norman’s perceived affordances [22], helping users with their cognitive actions. Hartson [12] defines cognitive affordances as “a design feature that supports, facilitates, or enables *thinking and/or knowing about something*”. *Example: A button label that helps users know what will happen if they click on it.*

Physical affordances help users with their physical actions, and match with Norman’s real affordances [22] (or Gibson’s affordances [10, 11]). According to Hartson [12], a physical affordance is “a design feature that helps, aids, supports, facilitates, or enables *physically doing something*”. *Example: A button that is large enough so that users can click on it accurately.*

Sensory affordances help users with their sensory actions (perceiving information). Hartson [12] defines a sensory affordance as “a design feature that helps, aids, supports, facilitates, or enables the user in *sensing (e.g., seeing, hearing, feeling) something*”. Sensory affordances play a supporting role for cognitive and physical affordances. Hartson thus explicitly separates sensing from understanding. *Example: A label font size large enough to read easily.*

Functional affordances are a design feature that help users *accomplish work*. It ties usage to usefulness, and is similar to McGrenere and Ho’s idea of “affordances in software” [19]. Functional affordances add *purpose* to a physical affordance. *Example: The internal system ability to sort a series of numbers (invoked by users clicking on the Sort button).*

These four types of affordances are tightly coupled and work together to help users in their interaction. Physical affordances are associated with the “operability” characteristics

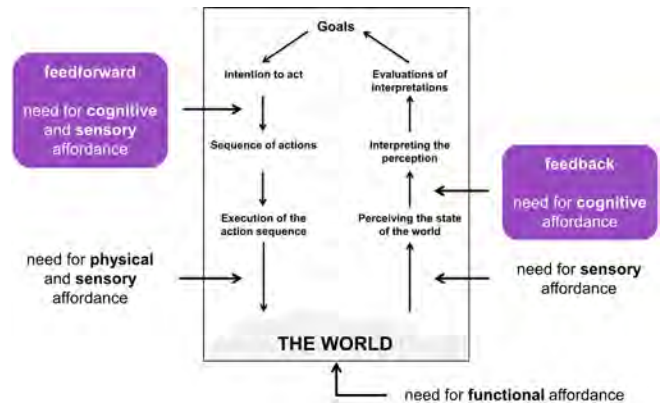


Figure 6. The need for physical, cognitive, sensory and functional affordances, and the position of feedback and feedforward in Norman’s Stages of Action model (image based on [12], adapted from [22]).

of user interface artefacts. Cognitive affordances are associated with the semantics or meaning of user interface artefacts. Sensory affordances have a supporting role, and are associated with the “sense-ability” characteristics of user interface artefacts, especially of physical affordances and cognitive affordances. According to Hartson [12], it is *design* that connects sensory affordances to physical and cognitive affordances, so that they can be seen, heard or felt to be used. Moreover, physical affordances carry a mandatory component of *utility* or *purpose* – the functional affordance – to which statements about physical affordances should refer.

Hartson’s framework significantly broadens the scope of affordances, so that they also include both the notions of feedback and feedforward. Hartson explains a cognitive affordance with the example of “a button label that helps users know what will happen when they click on it”, which essentially explains the purpose of this button, and can thus be seen as feedforward. Hartson explains how the four types of affordances helps user achieve their goals by plugging them into Norman’s Stages of Action Model [22], as seen in Figure 6. Earlier, we positioned feedforward and feedback in Norman’s Stages of Action Model (see Figure 1). It is interesting to note that Hartson identified the need for cognitive (and sensory) affordances exactly where we situate feedforward and feedback in Norman’s Stages of Action model (Figure 6), which suggests that both *feedback and feedforward are cognitive affordances*.

Furthermore, Wensveen’s augmented, inherent and functional feedforward [27, 29] can be explained in terms of Hartson’s four types of affordances:

Inherent feedforward : Wensveen [27] clearly sees inherent feedforward as a limited interpretation of the concept of Gibson’s affordance [10, 11]. It communicates what kind of action is possible and how it can be carried out. This is similar to Hartson’s *physical affordance*, as a physical design feature that helps users to physically do something [12].

Augmented feedforward : this is clearly a *cognitive affordance*, where lexical or graphical labels (e.g., words, pic-

tograms, spoken words) act as a sensory affordance to communicate the result of the user's action in an understandable way.

Functional feedforward : informs the user about the more general purpose of a product and its functional features [27]. As noted above, Wensveen's functional feedforward can be seen as a high-level nested affordance in Gaver's terminology. Even though Hartson does not explicitly state that his four kinds of affordances can be nested, we feel this is implied by several examples in his paper [12]. Wensveen's functional feedforward might be categorized as a *cognitive affordance* that makes the high-level functionality of the system – or its functional affordance – visible. If this is achieved through physical design, the product's form can be seen as a *sensory affordance* which allows users to recognize its functionality.

Note that Hartson's framework unites both Gaver's [9] and McGrenere and Ho's [19] work on affordances, and can be used to explain feedforward according to both Djajadiningrat's [6] and Wensveen's [27, 29] definitions. We will later use Hartson's framework to reframe feedforward and disambiguate it from perceived affordances and feedback. We conclude this section by giving an overview of aspects related to feedforward in Norman's work.

Norman – Feedforward, Symbols & Constraints

While this section started with the assumption that feedforward goes beyond affordances, Hartson [12] argued that feedforward is just a special kind of affordance, namely a cognitive affordance. Norman implies a purpose for a physical affordance (*e.g.*, a doorknob that can be turned in order to open the door), thereby eliminating the need for explicit feedforward. This is confirmed by Hartson [emphasis ours]:

In Norman's Design of Everyday Things world of non-computer devices, a purpose for a physical affordance is always implied. The doorknob is a cognitive and physical affordance for operating the door. The physical affordance offered by a doorknob does not mean merely that the doorknob can be grasped and turned. It means that the doorknob can be grasped and turned *in order to operate* (*e.g.*, invoke the function or mechanism of opening) the door; the user is enabled to operate the door. In turn, the door itself is a functional affordance that, when invoked, allows passage. In this interaction design view, a physical affordance gives access to functionality, the *purpose* of the physical affordance used to access it.

Hartson [12] notes that even though the addition of purpose to the description of a physical affordance is an obvious extension, it should be made explicit to avoid ambiguities in terminology.

There are several situations in which it is necessary to explicitly communicate the purpose of an action, especially in electronic products, graphical user interfaces (GUIs) and tangible user interfaces. There is often a great deal of visual similarity present and the same physical affordances are used multiple times (*e.g.*, physical and virtual buttons or sliders in products

or GUIs; and the typical cubes or blocks in tangible interaction [7]). Moreover, with the integration of sensing into everyday artefacts, it is often even harder for users to reason about the result of actions they undertake. It is important to note that complex systems that use labels and/or icons are not examples of bad design. It is often overlooked that Norman did not disapprove of labels and icons altogether, he only said: "When *simple* things need pictures, labels, or instructions, the design has failed." [22].

When Norman talks about more complex systems or devices, he provides two ways to help users determine the purpose of a user interface artefact: *mappings* and a good *conceptual model* [22]. As these mechanisms allow users to know what will happen when they perform an action, they could be seen as examples of feedforward. Mappings allow users to determine the relationships between actions and results, between the controls and their effects and between the system state and what is visible by spatial coupling. The controls are laid out in the same order as the artefacts in the physical world that they control. However, according to Djajadiningrat [7], mappings fall short when dealing with abstract data that has no physical counterpart. A good conceptual model allows users to predict what will happen when they perform an action by exploiting consistency. Consistency in the presentation of operations and results helps users to form a coherent, consistent system image [22].

Two other important design principles proposed by Norman are *symbols* and *constraints* [22]. Norman argues that these are not affordances and that wording in the label on a button, for example, is symbolic communication. Hartson [12] agrees, but states that under his own definition, communication is exactly what makes good wording effective as a cognitive affordance. It helps the user in knowing (*e.g.*, knowing what to click on). In other words, Hartson [12] sees symbols, constraints, and conventions as essential underlying mechanisms that make cognitive affordances – and therefore also feedforward and feedback – work. Hartson [12] argues that Norman would agree that cognitive affordances play an enormously important role in interaction design. According to Hartson [12], they are key to answering Norman's question: "How do you know what to do?". Hartson mentions that the design of cognitive affordances can indeed depend greatly on cultural conventions (or constraints [22]) as a common base for communicating the meaning of cues from designer to user.

REFRAMING FEEDFORWARD

In this section, we reframe feedforward informed by the above discussion of feedforward and related design principles such as affordances and feedback. We further clarify the differences between feedforward, (perceived) affordances and feedback based on Hartson's four types of affordances [12]. As Hartson not only subsumes Gibson's affordances [11] (or Norman's real affordances [22]) and Norman's perceived affordances [22], but also significantly broadened the scope of affordances to include the notions of feedback and feedforward, we feel his framework is useful to reason about the differences and interrelationships between these design principles. As discussed earlier, Hartson states that physical af-

fordances carry a mandatory component of *utility* or *purpose* – the so-called *functional affordance* – to which statements about physical affordances should refer. Hartson’s notion of conveying the *purpose* of an action is, indeed, nothing else than feedforward.

Disambiguation: Affordances, Feedforward & Feedback

As discussed before, Hartson situated his four types of affordances into Norman’s Stages of Action Model [22]. Remember that Hartson identified the need for cognitive (and sensory) affordances exactly where we positioned feedforward and feedback in Norman’s Stages of Action model (see Figure 6), which suggests that both *feedback* and *feedforward* are cognitive affordances. Hartson later confirmed this in personal email communication [13]. Even though it is a cognitive affordance, feedforward is also connected to the three other types of affordances.

Our new view on feedforward, feedback and perceived affordances is as follows:

Perceived affordances (Figure 7(a)) are cognitive affordances that are understandable through well-defined sensory affordances (e.g. a door’s handle) and reveal a physical affordance (an action possibility), which is coupled to a functional affordance (the action’s purpose). Perceived affordances occur before the user’s action and *invite them to an appropriate action*.

Feedforward (Figure 7(b)) is a cognitive affordance that is understandable through a well-defined sensory affordance (such as a readable, descriptive label or the object’s physical shape) and reveals the functional affordance (the system function) coupled to a physical affordance (the action possibility). Feedforward occurs before the user’s action and *tells users what the result of their action will be*.

Feedback (Figure 7(c)) is a cognitive affordance that is understandable through a well-defined sensory affordance (e.g. an informative message), and provides information about the result of a user’s action, which is offered to the user through the combination of physical and functional affordances. Feedback is provided during or after a user’s action and *informs them about the result of performing their action*. Feedback can later turn into feedforward again (in combination with a perceived affordance) for another action that logically follows the previous one.

Figure 7 illustrates these definitions and shows how perceived affordances, feedforward and feedback relate to each other and are linked to Hartson’s four types of affordances [12].

Both perceived affordances and feedforward tell users something about a particular action through a combination of a physical and functional affordances. Perceived affordances and feedforward essentially provide different information about the action that users have to perform to achieve their goals. While *perceived affordances reveal the physical affordance*, which tells users that there is an physical action available and how to perform it, *feedforward reveals the functional affordance*, which tells users what will happen when they perform that action. We agree with Hartson [12] that

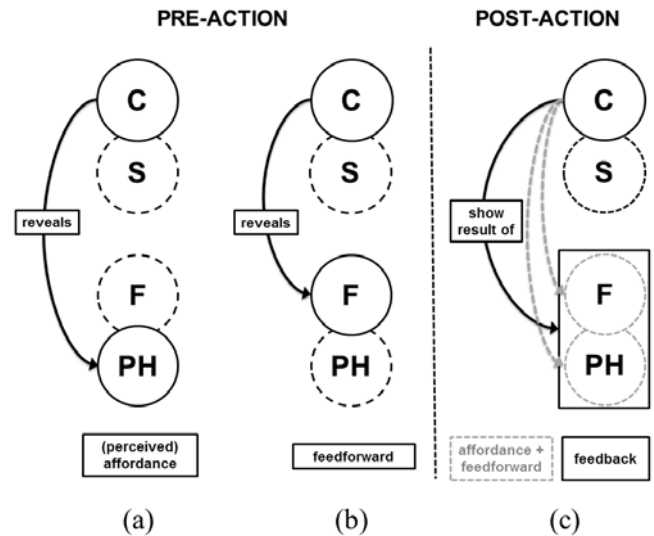


Figure 7. An overview of how perceived affordances, feedforward and feedback can be explained using Hartson’s four types of affordances. *C*, *S*, *F* and *PH* refer to Hartson’s Cognitive, Sensory, Functional and Physical affordances respectively. The functional and physical affordances together constitute a possibility for a purposeful action. While perceived affordances and feedforward provide information before the user’s action (*pre-action*), feedback occurs after the user’s action.

Norman always implied a purpose (or functional affordance) for a physical affordance (e.g., a doorknob that can be turned in order to open the door). As the purpose was implied, there was no need for explicit feedforward. Similarly, Norman always implied that perceived affordances were provided with a well-defined sensory affordance. As mentioned before, Hartson states that the addition of purpose should be made explicit to avoid ambiguities in terminology [12]. Additionally, we argue that the more complex a system or interaction context gets, the larger the need will be for elaborate feedforward in order to aid users in achieving their goals.

Note that feedback provided after performing an action might afterwards again serve as feedforward for the action that logically follows the previous one (Figure 7(c)). Wensveen refers to feedback that turns into feedforward as *inherent traces of action* [29]. In its simplest form, it is “nothing more than evidence for the user that he has acted on the action possibilities, as if it were a trace of the bygone action.” An example of feedback turning into feedforward is a physical light switch. When users flick the switch, feedback consists of the changed position of the switch, and, of course, also the light bulb that produces light (Wensveen’s notion of functional *feedback* [27]). However, the user’s action also changed the possibilities for action, as the light cannot be turned on again, it can only be turned off. In essence, the feedback of the glowing light bulb and the state of the switch, becomes feedforward indicating that flicking the switch again will reverse the state of the light bulb and thereby turn it off. Another example of feedback that turns into feedforward can be found in marking menus [17]. Once a function in the marking menu is invoked, its label is changed to the corresponding inverse function. This inverse function label, at first, serves as feedback to indicate that the previous function has been invoked,

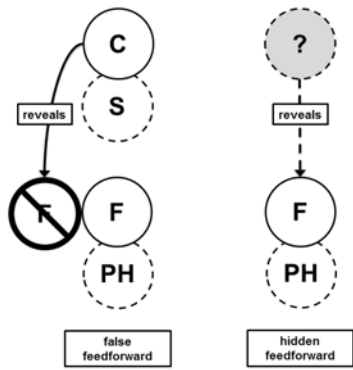


Figure 8. False and hidden feedforward.

and secondly, as feedforward for invoking the reverse function (thereby undoing the earlier action again).

Hidden and False Feedforward

Gaver also discerns between affordances – as in Gibson’s original definition [10, 11] – and the perceptual information available about them (so-called apparent affordances), which corresponds to what Norman defined as *perceived* affordances [22]. Based on this distinction, Gaver introduces the concept of *false* and *hidden* affordances, where the apparent information about the affordance is incorrect or missing respectively. A similar reasoning could be applied to feedforward and how an action is coupled to a system function, as shown in Figure 8. Feedforward can be *false* when it conveys incorrect information about what system function the action performs. When feedforward is missing, it *hides* how the action is related to the system function (*e.g.*, a button without a label).

Although undesirable, false and hidden feedforward might be useful notions to consider in interaction design. A simple example of a false feedforward in a graphical user interface is a button with an incorrect label (an effective technique which is often employed by malicious software to trick the user in invoking certain destructive actions).

Nested and Sequential Feedforward

Gaver [9] proposed the idea of *sequential* affordances and *nested* affordances for complex actions. We argue that feedforward can also be nested or sequential. In his discussion of sequential affordances, Gaver also mentions that affordances can be conveyed through multiple modalities (*e.g.*, visual, tactile, auditory information). Similarly, we think that feedforward could also be provided using different modalities, as confirmed by Wensveen [27] and Djajadiningrat et al. [6]. However, some modalities (*e.g.*, tactile) will be better suited to exploratory actions as they cannot be perceived through what Gaver calls “relatively passive perception” [9].

As previously discussed, Wensveen’s functional feedforward [27] conveys the general (top-level) function of a system (as seen in *e.g.*, the voice recorder in Figure 5(c)), and can be seen as the root of a nested feedforward hierarchy. It can be combined with feedforward that is provided for lower-level (or *nested*) functions. Functional feedforward typically

uses the shape of an object to inform users about its general purpose.

Gaver’s idea of *sequential* affordances, also applies to feedforward. Notable examples of sequential feedforward are systems that use feedforward to make gestural interfaces easier to use, such as Bau et al.’s OctoPocus [1] and Freeman et al.’s Shadowguides [8]. OctoPocus and ShadowGuides *continuously issue dynamic feedforward and gradual feedback during input*. While performing a gesture, users are provided with information about their current set of possible gestures (or action possibilities) and the expected result of those gestures (using a simple label), together with feedback about how well the current gesture has been recognized. Feedforward could also be made available at *discrete points* in time, instead of being updated continuously. Dynamically updating feedforward is probably easiest to achieve in software. Sequential feedforward in combination with feedback could further blur the difference between the two concepts since feedback might afterwards serve as feedforward for the user’s next actions.

Retrospect: Definitions and Examples

The table in Figure 4 shows which aspects of feedforward are covered by different definitions and also analyzes how feedforward is used in a number of notable examples.

As pointed out before, feedforward can be provided using multiple *modalities*. Unfortunately, designers mostly rely on visual information to convey feedforward, apart from a few exceptions, such as Djajadiningrat et al.’s TempSticks [6].

Bau and Mackay have introduced the *level of detail* as a useful criterium for classifying feedforward mechanisms. Usually, feedforward is provided in a low to average amount of detail. However, there might be situations in which feedforward can be provided with lots of details, for example to reassure the user when they have to trust the system (*e.g.*, for an e-commerce application or a smart home).

Feedforward can be nested in a *hierarchy*. There are a number of examples that use nested feedforward, such as Disney AppMATes and the Tangible Video Editor (TVE) [31]. Nested feedforward tends to rely on the object’s shape to convey its general function (*i.e.*, functional feedforward), combined with lower-level types of feedforward information. Disney’s AppMATes is a children’s toy which uses tangible toy cars which can be used on a tablet. In this case, the shape of the toy car serves as high-level feedforward that explains the general purpose of the object. When children place the toy car on the screen, the car’s lights will be shown on the display. This acts as additional feedforward information indicating that the car and display are linked, after which children can try to move the car on the display. In case of the TVE, the shape of the different building blocks indicate their function.

Finally, feedforward can be either be static or updated over *time* (*sequential feedforward*). An example of static feedforward is a fixed label in a tooltip. Examples of sequential feedforward are the OctoPocus [1] and ShadowGuides [8] dynamic guides, the TVE and SpeakCup [30]. When users combine different building blocks with the TVE, this arrangement

of blocks indicates new information that serves as feedforward for knowing how the movie clips will be combined together. SpeakCup is a digital voice recorder that is shaped like a small rubber disc with holes in its center. When the holes are pressed in, forming a small cup, SpeakCup absorbs sound. When the holes are pressed out, the stored sounds are released. SpeakCup uses its shape to communicate to users what they can expect. As this shape changes over time when the disc is pressed in or out, this is another example of sequential feedforward.

CONCLUSIONS

With this work, we reaffirm the importance of feedforward as a powerful tool for bridging Norman's Gulf of Execution. We strongly believe that, like affordances, feedforward is not a concept that can be easily defined. Although designers do use feedforward in many cases, they do so unknowingly and based on their experiences and skills. We have reframed feedforward in terms of Hartson's four types of affordances: (1) we disambiguated feedforward from related design principles such as feedback and perceived affordances; and (2) we identified four new classes of feedforward: hidden, false, sequential and nested feedforward. Our reframing of feedforward, together with the table in Figure 4 provides a means for designers to explore and recognize different opportunities for feedforward.

Acknowledgements

We would like to thank Stephan Wensveen for the interesting discussions we had on feedforward. We also want to express our gratitude to H. Rex Hartson for answering our questions about his work and sharing his reasoning on feedforward versus affordances. Many thanks to Johannes Schöning and Steven Houben for their suggestions and feedback on early drafts of this paper. We also thank Don Norman for his insightful suggestions. Finally, we are indebted to the anonymous reviewers whose comments further improved this paper.

REFERENCES

- Bau, O., and Mackay, W. E. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, ACM (Monterey, CA, USA, 2008), 37–46.
- Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., and Lopes, C. Making sense of sensing systems: five questions for designers and researchers. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, ACM (Minneapolis, Minnesota, USA, 2002), 415–422.
- Bellotti, V., and Edwards, K. Intelligibility and accountability: human considerations in context-aware systems. *Hum.-Comput. Interact.* 16, 2 (2001), 193–212.
- De Souza, C. *The Semiotic Engineering of Human-Computer Interaction*. Acting with Technology Series. MIT Press, 2005.
- Dey, A. K. Understanding and using context. *Personal Ubiquitous Comput.* 5, 1 (2001), 4–7.
- Djajadiningrat, T., Overbeeke, K., and Wensveen, S. But how, donald, tell us how?: on the creation of meaning in interaction design through feedforward and inherent feedback. In *Proc. of Designing Interactive Systems*, DIS '02, ACM (London, England, 2002), 285–291.
- Djajadiningrat, T., Wensveen, S., Frans, J., and Overbeeke, K. Tangible products: redressing the balance between appearance and action. *Personal Ubiquitous Comput.* 8, 5 (2004), 294–309.
- Freeman, D., Benko, H., Morris, M. R., and Wigdor, D. Shadowguides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, ACM (New York, NY, USA, 2009), 165–172.
- Gaver, W. W. Technology affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, ACM (New Orleans, Louisiana, United States, 1991), 79–84.
- Gibson, J. J. The theory of affordances. *Perceiving, acting and knowing: toward an ecological psychology* (1977), 6782.
- Gibson, J. J. *The Ecological Approach to Visual Perception*, new edition ed. Lawrence Erlbaum, Sept. 1986.
- Hartson, R. Cognitive, physical, sensory, and functional affordances in interaction design. *Behaviour & Information Technology* 22, 5 (2003), 315.
- Hartson, R. personal communication, 2010.
- Hornecker, E., and Buur, J. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM (Montral, Qubec, Canada, 2006), 437–446.
- Kaptelinin, V., and Nardi, B. Affordances in hci: toward a mediated action perspective. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, NY, USA, 2012), 967–976.
- Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, ACM (New York, NY, USA, 1994), 258–264.
- Kurtenbach, G. P., Sellen, A. J., and Buxton, W. A. S. An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Hum.-Comput. Interact.* 8, 1 (1993), 1–23.
- Lim, B. Y., and Dey, A. K. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, ACM (New York, NY, USA, 2010), 13–22.
- McGrenere, J., and Ho, W. Affordances: Clarifying and evolving a concept. In *Proceedings of Graphics Interface 2000* (2000), 179–186.
- Meckl, P. H., and Seering, W. P. Feedforward control techniques to achieve fast settling time in robots. In *American Control Conference, 1986* (june 1986), 1913–1918.
- Norman, D. *Living With Complexity*. Mit Press, 2011.
- Norman, D. A. *The Psychology Of Everyday Things*. Basic Books, New York, USA, June 1988.
- Norman, D. A. Affordance, conventions, and design. *interactions* 6, 3 (1999), 38–43.
- Norman, D. A. The way I see it: Signifiers, not affordances. *interactions* 15, 6 (Nov. 2008), 18–19.
- Saffer, D. *Designing for Interaction: Creating Innovative Applications and Devices (2nd Edition)*, 2 ed. New Riders Press, Aug. 2009.
- van Dam, A. Post-WIMP user interfaces. *Commun. ACM* 40, 2 (1997), 63–67.
- Wensveen, S. A tangibility approach to affective interaction. *Doctoral dissertation, TU Delft, Delft, The Netherlands* (2005).
- Wensveen, S. personal communication, 2010.
- Wensveen, S. A. G., Djajadiningrat, J. P., and Overbeeke, C. J. Interaction frogger: a design framework to couple action and function through feedback and feedforward. In *Proc. of Designing Interactive Systems*, DIS '04, ACM (Cambridge, MA, USA, 2004), 177–184.
- Zigelbaum, J., Chang, A., Gouldstone, J., Monzen, J. J., and Ishii, H. SpeakCup: simplicity, BABL, and shape change. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, TEI '08 (2008), 145–146.
- Zigelbaum, J., Horn, M. S., Shaer, O., and Jacob, R. J. K. The tangible video editor: collaborative video editing with active tokens. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, ACM (New York, NY, USA, 2007), 43–46.