Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-84525

AFFORDANCES IN PRODUCT ARCHITECTURE: LINKING TECHNICAL FUNCTIONS AND USERS' TASKS

Adriano B Galvao
Institute of Design
Illinois Institute of Technology
350 N LaSalle Street, 4th floor
Chicago, Illinois, 60610, USA
Tel: (312) 595-4900, *galvao@id.iit.edu*

Keiichi Sato
Institute of Design
Illinois Institute of Technology
350 N LaSalle Street, 4th floor
Chicago, Illinois, 60610, USA
Tel: (312) 595-4912, *sato@id.iit.edu*

ABSTRACT

Developing usable and desirable products requires an understanding of how users build close relationships with objects and how these relationships can be controlled by developers. This paper discusses the importance of the concept of affordances as an instrument useful for understanding the relationships between technical functions and user tasks. The approach introduces a Function-Task Design Matrix to link technical functions with user tasks and to capture relevant affordance-level requirements throughout the product architecture generation. Functional and Operational Affordance levels are introduced to help determine the product attributes necessary to optimize the ease with which users can undertake technical functions. The paper uses functional language, focusing attention towards the use of the product, rather then merely its workings. The tools for describing affordances are described first, followed by a step-by-step description of how they can be used to improve decisions during product architecture generation. The mechanism is illustrated in a case study on a kitchen appliance.

Keywords: *affordance levels, product architecture, design structure matrix, interaction groupings and user requirements.*

1. INTRODUCTION

Developing usable and desirable products takes an understanding of how users build close relationships with objects and how these relationships can be controlled by developers. Efforts to understand these relationships in design research have recently been directed toward the concept of affordance. This movement is supported by the body of knowledge developed in the field of psychology and human-computer interaction and promoted by the need to incorporate user-centered requirements into product architecture generation. Considering the impact of user requirements on downstream issues of product architecture generation, the need for structured methods that address affordances at the function level is significant. Because of this, the research problem addressed here is the development of a method to link technical functionality and user requirements, using affordances as a conceptual instrument.

Recent work in engineering design distinguishes the affordance concept from the functional way of thinking. While function-based modeling methods, such as Quality Function Deployment, functional decomposition [1], and input-output flow heuristics [2], assist developers generating product architectures based on the gathering of customer needs, they do not take into account the product relationships with users during product use. Affordance-based methods [3, 4], on the other hand, assist developers in discovering these relationships, but provide only a surface-level understanding where in the product architecture these relationships are established and how they could leverage product architecture decisions. This paper posits that improvement on product architecture decisions can be achieved by merging the functional and the affordance ways of thinking.

To fully comprehend the interplay between functions and users' requirements, a shift from the traditional product-centered approach to the user-centered approach has been suggested [5, 6]. Advantages for incorporating the user-centered approach include the expansion of need analysis by looking into the relationships between users' activity and available technology. This analysis, when combined with use of the concept of affordance, becomes the basis of the Product-Affordance-User model (Fig. 1). In this model, affordances are simply the relationships established between the product and the user during product use.

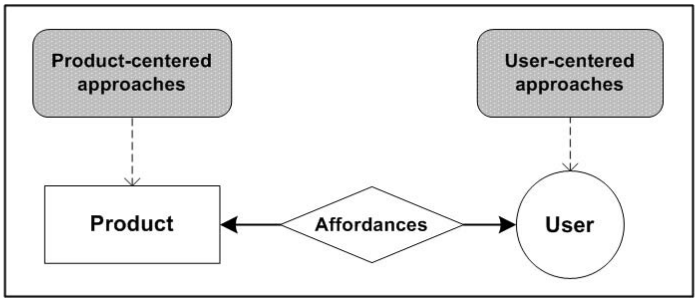


Figure 1. Product-Affordance-User model.

This paper describes a methodology that associates technical functions with user tasks to provide product architecture guidelines based on the affordance concept. The hypothesis here is that better product architecture decisions can be made by decomposing technical functions and users' tasks and combining them with affordance requirements to generate alternative product configurations. In order to test this proposition, we initially revisit some of the previous research on the affordance concept, including the original definition in psychology and its subsequent use in the field of engineering design. We then extend the affordance definition by adding two levels of affordances (functional and operational) and propose a method that results in product architecture representations based on a Function-Task Design Matrix. Finally, we introduce a case study to test the method with analysis and observations.

2. BACKGROUND

In the affordance concept, product properties offer a direct link between perception and action. In order to understand how these concepts have been adopted and appropriated, past and current work was reviewed in the following sections.

2.1 Perspectives on Affordances

The concept of affordances has been the focus of attention in psychology for some time, but only recently has it been applied by the design community. The term "affordance" was first introduced by the psychologist Gibson and was described as being a specific combination of the properties of the environment and its surfaces taken with reference to an animal [7]. Gibson had been influenced by Gestalt theorists, such as Koffka [8], who recognized that the meaning of things seems to be perceived as readily as the things themselves, telling us what to do with them. These theories lead Gibson to believe that meaning could be inherent in the artifacts of the environment we live in. He stated that

> *...our own experience of the visual world can be described as extended in distance and modeled in depth; as upright, motion-less as a whole, and unbounded; as colored, textured, shadowed, and illuminated; as filled with surfaces, edges, shapes, and inter spaces. But this description leaves out the fact that the surfaces are familiar and the shapes are useful. No less than our primitive ancestor, we apprehend their uses and dangers, their satisfying or annoying possibilities, and the consequences of action centering on them (J. J. Gibson, 1950, p. 198, emphasis added).* [7]

Gibson suggested that we perceive what the environment and its characteristics offer in terms of mediums, surfaces and substances rather than molecules and atoms. Essential to this conception is the idea that animal and environment make a pair, and, as a result, a relationship exists between them. Parts of this relationship are the persistent (invariant) features of the environment that permit the animal to do things. These permitted actions are "affordances". According to his view, an affordance has three features:

- It is independent from our perception;
- It exists relative to our actions;
- It doesn't change as our needs and goals do.

More specifically, the first assertion implies that an affordance is a stable property of the artifact and its existence does not depend on our interpretation or experience. This means affordances do not need to be visible, known or desirable to exist in a particular context. The second assertion, perhaps the most radical contribution of Gibson's work to the notion of affordances, implies mutual relationships between our actions and the artifacts around us. These relationships are only conceivable when paired with actualizing agents (e.g. users) that can make use of these relationships or simply acknowledge their existence. The third assertion implies that an affordance is a constant that may not vary at the same rate as our motivations and actions do. This is especially true for products that have physical moving parts with which we interact and less true for information products that rely on operational instructions.

Gibson further asserts that affordances "explain the sense in which values and meanings are external to the perceiver" [7, p.127], thus reducing meaning to ecological relationships. He emphasizes, "the niche (defined as a set of affordances) should not be confused with what some animal psychologists have called the phenomenal environment of the species, the 'subjective world,' or the world of 'consciousness'" [7, p.129]. These claims place affordances beyond the traditional concerns familiar to academics.

In 1984, William H. Warren's experiment [9] assigned a metric to the concept of affordances by using ratios between properties of the environment and properties of humans. For example, an edge would not be discerned as a step or a cliff by its absolute size or shape but rather how it relates to a particular user, including that user's size, agility, and style of locomotion. Ratios, defined as π numbers, would represent the affordance of stair climb-ability between the stair (H) and the climber's leg length (L); ($\pi = H/L$). For climbers of different heights, a lower ratio (π_0) determines the minimum energy expenditure required to climb a given vertical distance, and a maximum ratio (π_{max}) determines when a stair becomes impossible to climb bipedally. Warren concludes that the "perception for the control of action reflects the underlying dynamics of the animal-environment system [9]. In design terms, this set limits for target users.

In contrast to Warren, Norman [10] investigated affordances of everyday things, such as doors, telephones, and radios, and argued that their embodiment provides strong clues to their operation. More important, he rekindled the notion of affordances as the result of the mental interpretation of things, which is based on people's past knowledge and experiences. The term affordance, in this case, refers to "the perceived properties of the thing…that determine just how the thing could possibly be used" [10, p.9]. More recently, Maier & Fadel introduced positive and negative bounds for affordances to emphasize what the artifact should afford while safeguarding what the artifact should not afford. They positioned the affordance concept "away from the specific ecological context in which Gibson originally defined" [3, p.786], suggesting two ways to talk about affordances: the User-Artifact Affordances for interactions between a living agent and an inanimate agent in which perception is of utmost importance and the Artifact-Artifact Affordances for interactions between two inanimate agents. The major difference here is the existence of perception and the artifact's usefulness [3, p. 789]. The approach adopts Gibson's assertion that affordances exist independent of our perception and can be described as "-ables" as in "catch-able", "walk-able", "climb-able", and "type-able." Table 1 summarizes the meanings assigned to affordances by Gibson, Warren, Norman, and Maier.

Table 1. Comparison of meanings assigned to affordance concept.

	Perception dependency	**Action dependency**	**Susceptibility to change**
Gibson 1979	*Independent of perception*	*Dependent on actions' possibilities*	*Immutable, it exists in the environment*
Warren 1984	*Dependent on perception and physical limitations*	*Dependent on individual abilities*	*Mutable, it may change according to user's ability*
Norman 1988	*Dependent on perception and culture*	*Dependent on all the above plus cultural issues*	*Mutable, it may change according to user's background*
Maier & Fadel 2003	*Dependent on sensing mechanism*	*Dependent on perception of usefulness*	*Mutable, it may change according to design purposes*

Among these different viewpoints, perhaps the most important to the designer of products is how affordance characteristics relate to users' actions. Gibson stresses the action possibility as an affordance. Norman's definition is based on how that possibility is conveyed to the user and Warren's definition on how that possibility is accomplished within limits of the environment.

These views orient developers to consider affordances at three instances: 1) the affordances embedded in the product, 2) the perceptual attributes of these affordances in users' mental models, and 3) the instantiation of affordances when users perform their actions. For example, in the act of sitting on a bench, first there are affordances independent of our perception that exist in the product: the horizontal representation of the bench as flat, extended, rigid, and approximately knee-high off the ground, all relative to our proportions and position in space properties. Second, there is our perception of these properties—surfaces, proportions, textures, and relations with our body—that forms derived mental descriptions. And third, the act of sitting that instantiates affordances, closing the cycle. Figure 2 illustrates these instances as a magnification of the Product-Affordance-User model from Fig. 1, with a second stage in that process: the output of interaction that provides feedback and creates changes in the affordance perception within our mental models.

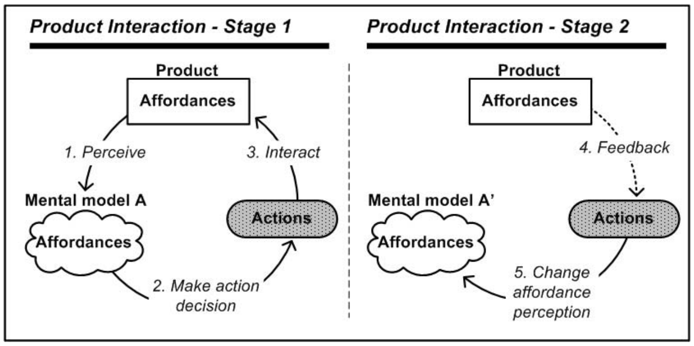


Figure 2. Magnification of Product-Affordance-User model.

2.2 Reflections on Affordances

The concept of affordances is not an easy to express in precise analytical terms [3]. In order to appropriate it for design purposes, the concept of affordance needs to be described precisely enough that developers can use it to manipulate specific product attributes. While a product can be described by its function and its features, affordances could provide additional understanding of the relationships that take place between the product and the user during product use.

An affordance, such as one for sitting on a bench, provides tangible constructs in reference to the physical medium, but it also raise issues regarding the user-perception domain. Vera and Simon described these affordances as *internal representations of complex configurations of external objects* [11]. This reinforces the concept that, in order for us to perceive products, some change in our mental representation must occur and the resulting mental representation also deserves to be called an "affordance."

Moreover, these mental representations are inherently associated with our actions—thus, affordances provide cues upon which we act, one after another. This leads to the notion of sequential affordances when they are revealed over time as we interact with products [12]. These sequential affordances have a one-to-one correspondence with our actions. In a simplistic manner, a glass of milk affords drinking, but drinking is composed of holding, sipping, and swallowing, all of which are afforded by the glass of milk.

In order to generate designs that require the use of human actions, developers need to understand what tasks the product should support and what behaviors should be built into the product. Users, therefore, must be observed in their messy reality in order to capture their workarounds and the problems they face during product use. This approach moves the creative thinking away from the functional aspects of the product. This paper proposes a systematic inclusion of user studies combined with affordance thinking in the product architecture generation as discussed in next sections.

Two important definitions result from the reflections on affordances above. They are "affordance" and "mental representation." The term "affordance" in this paper is used in much the same way as Gibson, paraphrased as affordances of the environment are what it offers the animal [7]. The scope of the concept, however, is purposefully narrowed to the product level and considered as a relationship between technical functions and user tasks, which suggests desirable product attributes that help users to accomplish their goals and which defines directions towards design implementation. The definition for "mental representation", as discussed in previous work [5], is considered a structure within "mental models" that can provide descriptions about users' perceptions. The concept of mental models consists of three knowledge sets: a set of elements and their identification, a set of relations among these elements, and a set of influences that affect our ability to make predictions about the interaction of these elements [13]. In section 4, a case study exemplifies affordances captured in mental models of experiment subjects and ways to tune this information for design purposes.

3. AFFORDANCES IN PRODUCT ARCHITECTURE

When developers are defining the ways in which affordances can manifest in the course of product architecture, the importance of the concept becomes somewhat vague, leaving developers in a quandary about how to reasonably make use of this concept. This paper responds to that by suggesting three concatenated methods that incorporate affordance levels relative to the user activities and technical functions of the product to be designed or redesigned.

These methods, listed in Fig. 3, are based on the output of early user and product studies, adding three steps in the product architecture generation process described in by Stone et al in [14]. Existing methods covered in steps 1, 2, 6, 7 and 8 are well documented in literature and are not the focus of this paper. Instead, identifying and applying product affordances (Steps 3, 4 and 5) are the focus. The Affordance-Based Generic method (Step 3) identifies product uses to create a generic list of affordances. The Function-Task Interaction method (Step 4) crosses technical functions with user tasks to identify existing affordance relationships, taken with reference to the user, and to determine affordance indexes that points to demands of the product on the capabilities of the users. And, the Affordance Levels and Attribute method (Step 5) defines solutions based on conceptual affordance levels, using design knowledge or experimental studies. An overview of the combined product architecture design process is shown in Fig. 3.

3.1 User and Task Analysis

Successful design requires a good understanding of users and their activities. Task analysis is an effective way to decompose activities into subtasks and provide deep understanding of the possible ways users fulfill their needs. More importantly, this study helps developers decide whether the product's technical function or the user, through his/her tasks, should carry out active roles. Work in this area includes methods to describe users' needs, their use-processes [15] and the context-of-use [16].

Within the scope of this paper, a task is what someone does to achieve a goal [17], such as making tea, hanging a picture on the wall, and giving a presentation. Tasks can be broken down into hierarchical descriptions with several levels of abstraction, from general tasks to specific subtasks. These subtasks may include operations users perform almost automatically. With this information available, developers can, throughout the stages of product architecture generation that follow, better comprehend the relationships their designs will eventually create in the hands of users.

3.2 Function Model Decomposition Methods

Function model decomposition approaches are useful tools for developers and probably the most accepted practice in the conceptual phase. In this step, developers define the overall functions the device needs to perform and then decompose them into sub-functions that delineate design problems to be solved [18]. The decomposition continues until it generates a function structure that organizes top- and low-level functions in enough detail and allow for the search of solution components and their combination. For the purposes of this paper, the information obtained from function structures is sufficient for the following reasons: first, because it provides enough information about product behaviors that are relevant to the affordance study, and second, it avoids a specific design.

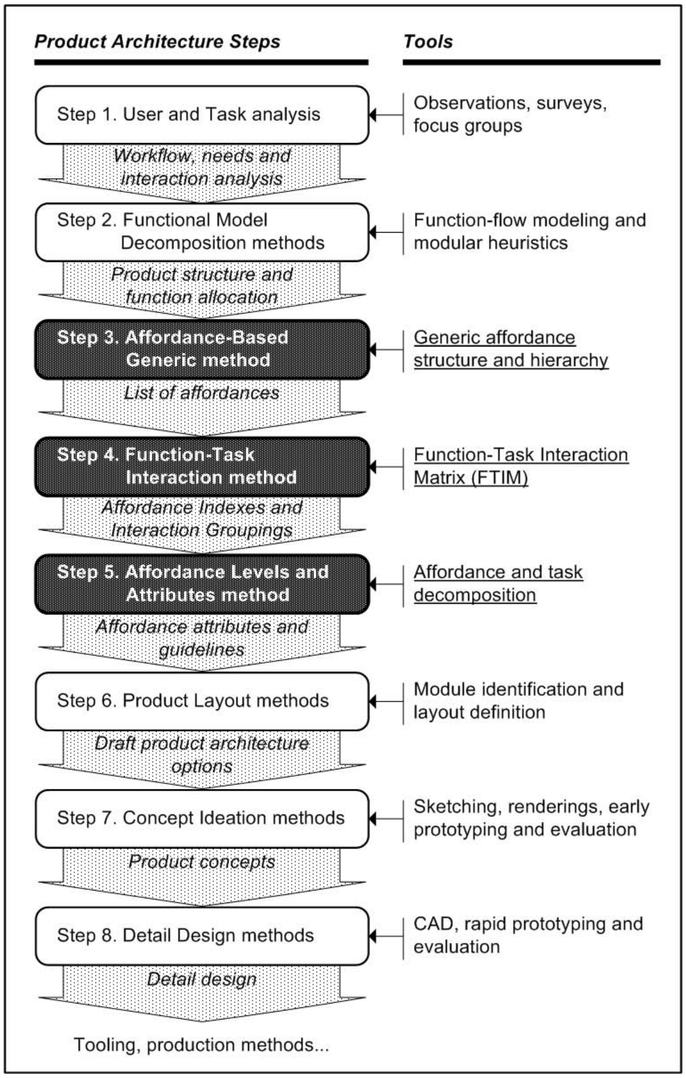


Figure 3. Existing product architecture generation process with additional affordance methods and tools (shaded blocks). Outputs are shown directly under respective steps.

3.3 Affordance-based Generic Method

Individual efforts have introduced affordances as a generic concept underpinning theoretical precepts of technical functions. Proposed methods consider affordances at higher levels of abstraction, adding to product studies by identifying “must have” relationships between the users and the product throughout the product lifecycle. The process of defining these affordances consists of three steps; 1) understanding and expressing user needs in terms of affordances, 2) creating a generic affordance list and 3) prioritizing affordances [3]. The output of this phase is a generic list of affordances prioritized by user groups. The list emphasizes product attributes that should receive greater attention during product architecture, generation based on the utility each provides for a particular user group. For example, a developer may chose to evaluate product requirements relating to manufacturability, usability, maintenance, and retirement. These requirements are then organized into positive affordances (what the artifact should afford) and negative affordances (what the artifact should not afford) in order to satisfy different user groups. Prioritized affordances become the input for the following phases of the proposed methodology.

3.4 Function-Task Interaction Matrix Method

The Function-Task Interaction Matrix, abbreviated as FTIM, is a template to represent the relationships between technical functions and user tasks. This matrix is an extension of the Dependence Structure Matrix (DSM) described by Steward [19], which was created for the analysis of the structure of a system’s architecture through analyzes of its relations and dependencies. Among other applications, this matrix allows different teams within the same organization to represent information about relations among components of a product [20] and input/output relationships. In this paper, the DSM is modified to model relationships between technical functions and users’ tasks. As long as there is some kind of interaction between users and the product being developed, this information remains relevant to the developer. The FTIM helps answer the questions: does this technical function interact with the user? If yes, what affordances attributes should it have?

The matrix proposed relates two types of elements: technical functions from functional model decomposition studies as the row headings, and user processes from task analysis studies as the column headings. For most products, there is a subset of technical functions that directly meet users’ tasks. Other functions do not necessarily address any users’ tasks, but are necessary to describe the product’s functionality and are called carrier functions [21]. These functions are important for product behavior and can be the sources of indirect affordances. Figure 4 provides a generic example of FTIM.

The completeness and accuracy of interactions captured in the FTIM increase as our knowledge of the design increases. The strength of the FTIM lies in the visualization of the interaction points and relationships established. Interaction here refers to technical functions that are required to engage in a certain user task and vice-versa. The relationships from technical functions to user tasks specify which product attributes are needed, while the relationships from user tasks to technical functions specify where affordances can aid users in accessing them. By mapping technical functions to user tasks, existing interactions are identified. This is a subjective evaluation process that depends on how familiar developers are with the product and tasks being considered.

In the generic example, filled squares and circles in the matrix denote acknowledgement of these interaction types: squares for physical contact interactions and circles for cognitive interactions that utilize the output of technical functions (e.g. noise or heat) as a cue for interaction. Note that it is possible for one task to have both types of interaction, and therefore each task in the FTIM has two sub-columns to accommodate these interaction types.

In addition to defining interactions within FTIM, task demands on technical functions are determined by summing points from physical and cognitive interactions vertically. The overall sum presented on the bottom line of Fig.4 provides affordance indexes. For example, on the third column of FTIM below, there are three interactions between ***Task 3*** and ***Technical Functions b***, ***c*** and ***d***—each interaction suggesting interdependence between elements and therefore great reliance on affordances. Note that ***Function b*** offers two interactions based on perceived signals from energy or material flow; these may be the result of unintended interactions between technical functions (i.e., noise, vibration, heat, as described by Ulrich and Eppinger in [22]) or signals specifically designed to inform users (e.g. ring tones). The boxes without interaction points are as important as the other ones, since they indicate a missing relationship or inexistent demand for product affordances in particular task.

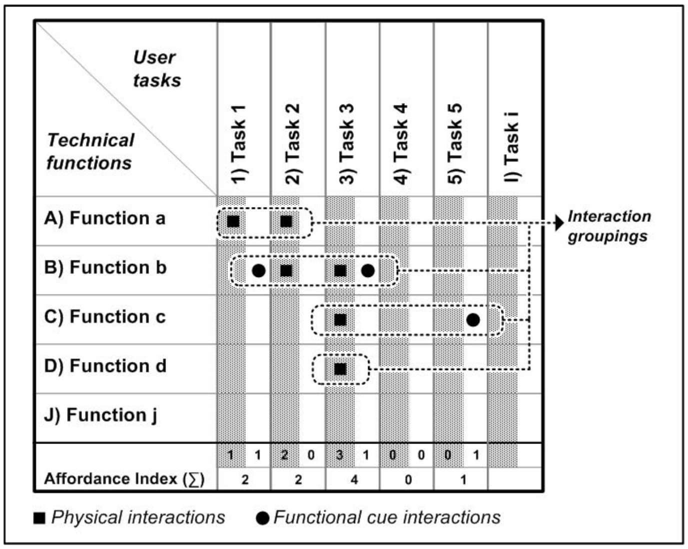


Figure 4. Simplified example of a FTIM.

The third output of FTIM is the groupings of interaction elements that are obtained from each row of FTIM. These groupings will be used to represent generic affordances and are important to the next steps in which tasks are decomposed into subtasks to define affordance attributes.

Interaction Group 1 $= \{ I_{A1}, I_{A2} \};$
Interaction Group 2 $= \{ I_{B1}, I_{B2}, I_{B3} \};$
Interaction Group 3 $= \{ I_{C3}, I_{C5} \};$
Interaction Group 4 $= \{ I_{D3} \};$
Interaction Group n $= \{ I_{JI} \}$, *where* I_{JI} *indicates*
an interaction element to perform j^{th} *function with* i^{th} *task.*

Note that it is possible different developers generate alternative FTIMs. In order to verify the correctness of the Function-Task Interaction Matrix, it should be presented to experts from different backgrounds, including those who were involved with the task analysis. Each expert can review the interactions in the FTIM and agree that most of the interactions captured were correct and affordance groupings were realistic. Another alternative to confirm these interactions points and related affordances is to map them back to a functional model diagram as shown in the case study. In the following section, we present ways to decompose and thereafter manage users' tasks. The methodological steps that have been discussed up this section are further outlined below:

Step 1: List technical functions based on functional modeling and user tasks based on task analysis.
Step 2: Enter technical functions and user tasks in the FTIM.
Step 3: Identify interactions existence in FTIM.
Step 4: List interaction groupings for next steps.
Step 5: Consult developers to confirm interactions and evaluate overall affordance ratings in FTIM.

3.5 Affordance Levels and Attributes Method

Up to this phase, the FTIM has been used to integrate the users' tasks and technical functions in a new way, and to combine them into single interaction groupings. The next step is to consider affordances as properties that can be analyzed in their own terms. To that end, two affordance levels are proposed: Functional and Operational. The **Functional Affordance Level** identifies high-level relationships that carry a component of utility or sense of usefulness throughout the product lifecycle and are described as "-abilities", such as "clean-ability". Affordances at this level give a user's bird's eye view of the product utility and provide developers with generic requirements to which a future concept must conform, including manufacturability, usability, re-configurability, maintainability, sustainability, and retirement. The **Operational Affordance Level** defines relationships that points to precise informative and structural attributes products carry, which are directly related to the functional affordance level. Figure 5 illustrates the way in which these affordance levels are within the FTIM.

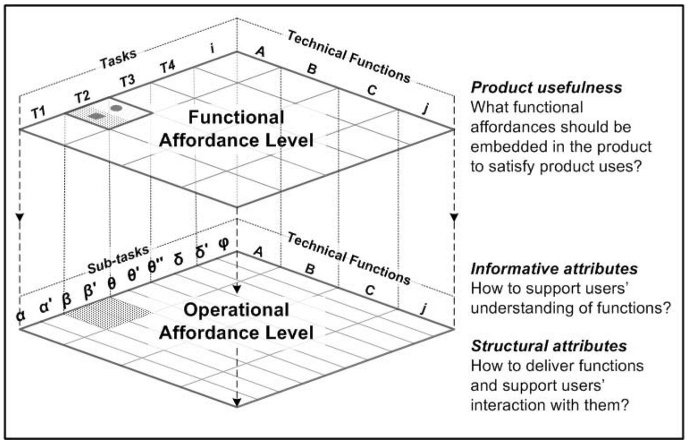


Figure 5. Affordance Levels and Attributes.

Within the Operational Affordance Level, **informative attributes** assist users' understanding about product behavior possibilities (e.g., energy and material signals helps users know what will happen if they act on the product, such as a blinking red light and buzzer suggesting a problem or demanding attention). These attributes are meaningful, noticeable elements that can assist users with their formation of mental models when perceiving the product and with their physical operations while acting upon it as indicated previously in Fig. 2. **Structural attributes** provide physical access to technical functions by pointing to specific product characteristics that can help users with their subtasks. These attributes, defined by form, color, material and layout, allow users to properly use the product (e.g. a surface that is large and rough enough so that users can push on it accurately with their fingers, such as a button by being slightly raised above an otherwise flat surface that suggests the idea of pushing it, or a lever by being an appropriate size for grasping that suggests the idea of pulling it). Both informative and structural attributes provide invitations for subtasks and are often coupled, since they refer single technical functions.

Restructuring affordances around these levels additionally requires careful thinking about tasks as smaller units or subtasks users perform. In this paper these subtasks refer to the action possibilities stressed by Gibson. This consideration will guarantee that the relationships between technical functions and correspondent tasks are analyzed in greater depth.

The first step is to expand the original tasks by looking into their subtask possibilities, a process achieved by hierarchical task analysis. For a simple FTIM, it is fairly simple to decompose the tasks into subtasks by prompting each task with the question: ***In order to* "technical function 'j' (rows in FTIM*)," the user has to* "user task 'i' (column in FTIM)" *by:* "subtask options."** For example, consider the task of cutting a piece of wood with a chainsaw. The subtasks pertaining to the task "cut a wood block" can be elicited by ruling: ***in order to*** "cut a wood block," ***the user has to*** "handle the chainsaw" ***by:*** aligning the saw, by keeping the saw in the right position, and by activating the saw. Figure 6 illustrates an iterative approach to elicit subtask options users engage in while performing their tasks using a specific function.

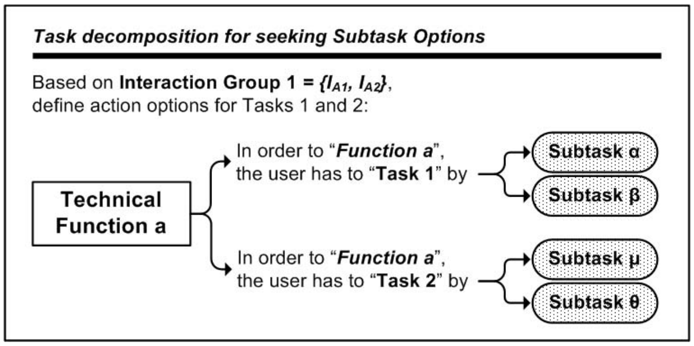


Figure 6. From technical function to subtask options.

Once all possible subtasks are listed, the final steps in this process are to look into each subtask derived from the technical function, compare it against existing solutions and explore both informative and structural affordances that might support the subtask. The reason for these last steps is twofold: 1) to seek suitable subtask possibilities for a specific technical function, and 2) to determine whether chosen subtask will require additional affordances. This analysis will point to specific informative and structural attributes related to product implementation regarding appearance and behavior. These attributes can be viewed as product requirements to match intended technical functions. Figure 7 illustrates this step with greater detail.

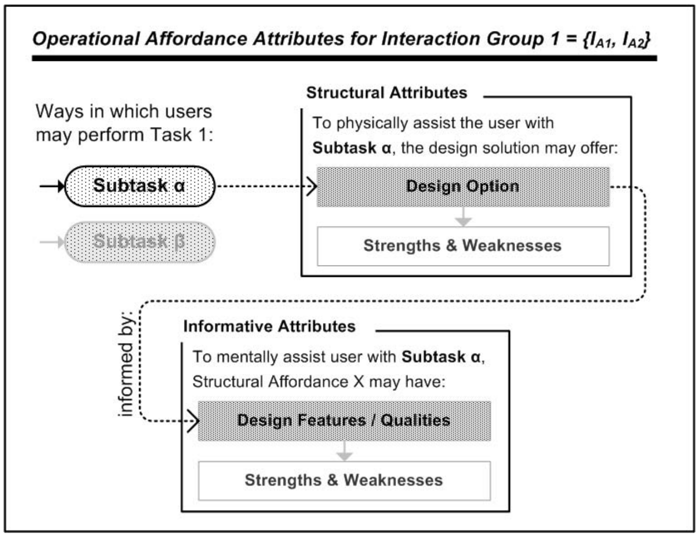


Figure 7. From subtask options to affordance attributes.

In summary, the task decomposition analysis provides the means to discover innovative ways to improve technical functions by aggregating structural and informative affordances attributes. The other phases of the product architecture generation process described in Fig. 3 (product layout, concept ideation, and detail design) are beyond the scope of this paper, but the output thus far has implications that should be explored in future studies.

4. CASE STUDY

The purpose of this case study is to illustrate the Affordance Level and Attributes methodology and to evaluate its appropriateness to the early phases of product architecture development. The goals of the experiment were to capture tasks performed by subjects and, most important, detect affordance levels and attributes introduced in sub-section 3.5. The experiment was video taped to ensure conditions affecting the interaction were captured for deeper investigation.

4.1 Understanding the User and Product

Most of the background information about the subjects was collected through semi-structured interviews to determine product-related skills, product knowledge, activities and methods of use. The skill level of each test subject is summarized in table 2. The investigation was defined around subjects' abilities during product-use and their understanding of kitchen blenders. Eight observations were conducted in U.S. family homes using their own blender. The choice of this product was based on three criteria: 1) product commonly used in the home kitchen, 2) familiarity with product operation, and 3) simplicity of operation.

Table 2. Characteristics of subjects.

Gender	Male (3); Female (5)
Age range	20 to 48
Cooking frequency (%)	Daily-12.5, Weekly-50, Monthly-25, Yearly-12.5
Product usage (%)	>1year-25; >5years-37.5; >10years-37.5

During observations, the subjects were asked to make a cold drink and prompted to talk aloud while being videotaped. The ingredients were provided but no instructions were specified. After the experiment, task analyses were carried down to the individual steps and decisions subjects made as they performed their individual tasks. Based on this experiment, it was found that the activity flow for operating a typical blender is influence by contextual factors and involves four main processes. These processes and tasks are shown schematically in Fig. 8.

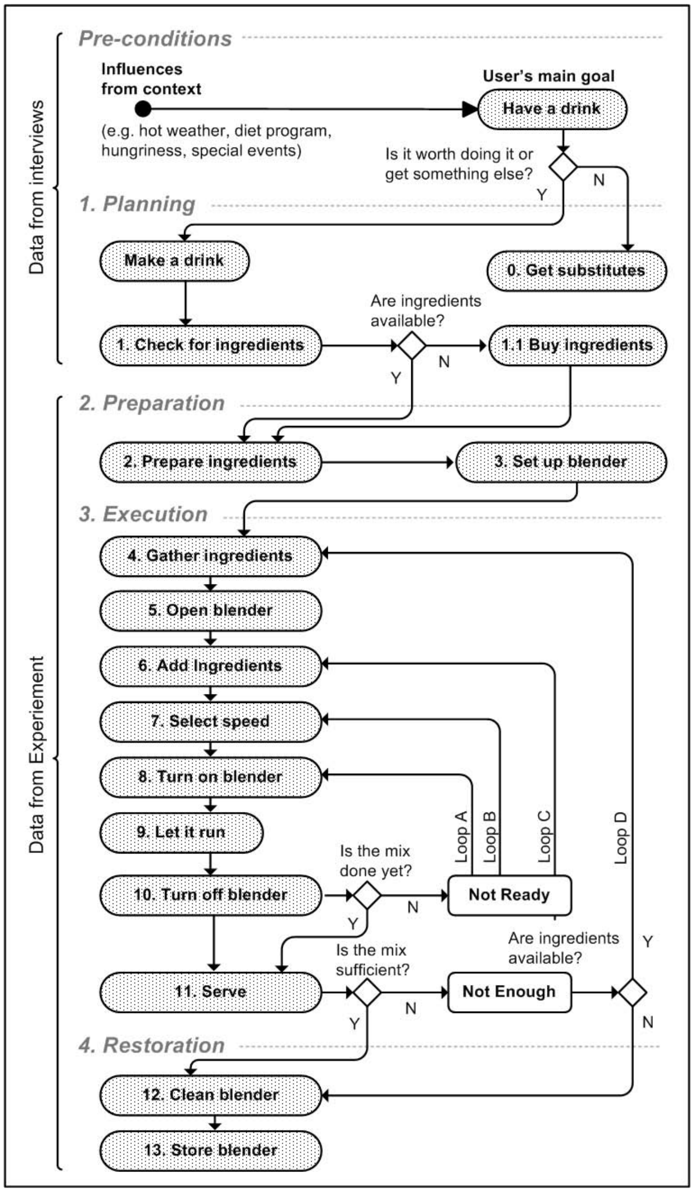


Figure 8. Typical task flow for blender appliance.

The process of using a blender starts with contextual influences that must take effect prior to product use. Examples of contextual influences include factors such as: weather affecting user's thirstiness state or diet program affecting the choice of ingredients. Assuming the contextual implications have taken effect, a typical process to make a drink from start to finish using a blender may include up to 13 tasks and several subtasks (in this case study, move, support, secure, press).

Subsequent to the task analysis, a range of blenders were assessed to quantify typical product archetypes and to define technical functions based on the functional modeling decomposition [23]. Twelve technical functions were identified and distributed among four modules (control, motor, material, and structural) based on four flows (human energy, electricity, material, and mechanical energy). The functional model diagram of a typical blender is shown in Fig. 9.

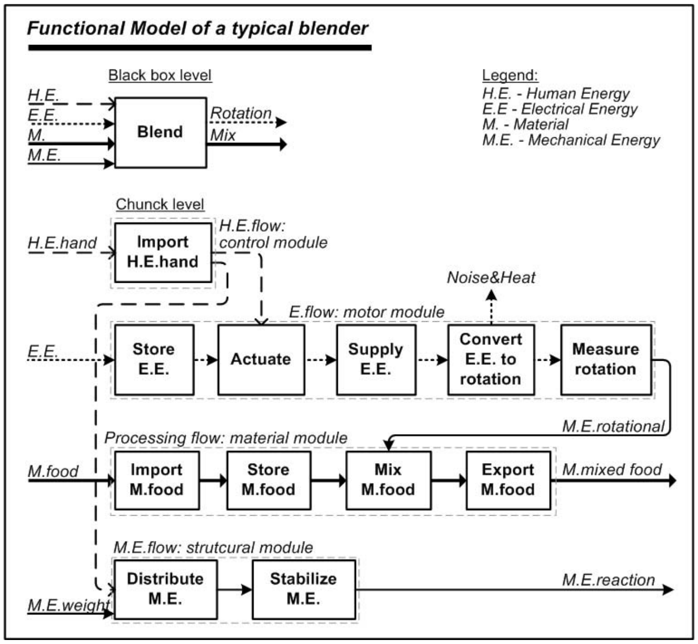


Figure 9. Functional Model of a typical blender.

More important than mapping the typical user process and defining the technical functions was to discover how users perceive affordance possibilities in the product architecture of blenders. Prior to making the drink, subjects were asked to create a rough sketch of their blender, to identify as many parts as possible and to describe the utility of each part. For repeatability assessment, simple frequency of occurrence of product parts was identified in drawings of experiment subjects. The results show that subjects used, in average, 63% of visible product parts to represent the product. Although this study did not (and was not meant to) have statistical significance, it demonstrates the range of subtask possibilities assigned to each product part.

In addition to documenting the frequency with which subjects represent product parts, their ability to identify product attributes and subtasks for product operation were also investigated by prompting questions about the usefulness of each part represented in their diagrams. The basic subtasks identified were: to pick up the blender (or the jar); to hold and carry it to the nearby countertop surface; to energize the blender; to measure the content; to fill the blender with contents; to start the blender to monitor the mix while it runs; to secure and pour the content into a cup; to clean the jar; and to store the blender. Table 3 shows these findings with the frequency of drawings, user concerns evoked by each part, and the subtasks required for operating each product part. The drawings together with the subtask rationale for each part provide descriptions for users' mental models they use during product interaction.

Table 3. Elements of perceived product architecture of blenders.

Part	Frequency of occurrence	User concerns	Related subtask
switches	1.00	size and layout	press, turn, slide, push
jar	1.00	size and stability	monitor mix, serve
handles	1.00	size and stability	move, hold
markings	0.86	size and layout	measure, gauge
cord/plug	0.86	size and accessibility	energize, reach wall
base	0.86	size and stability	move, support
lid and cap	0.71	seal-ability	pull, press, push, lift
blade	0.57	sharpness and cleanness	rinse, wipe off
gasket	0.12	sealing and cleanness	secure jar
sealing ring	0.00	cleanness	-x-
pivot	0.00	-x-	-x-
motor	0.00	-x-	-x-
Average	**0.63** (of visible parts represented)		

4.2 Describing Functional Affordances

From the user and product architecture studies, the following functional affordances were identified: 1) countertop-ability; 2) support-ability; 3) transportability; 4) multi-speed-ability; 5) mix-ability; 6) remove-ability; 7) clean-ability; 8) measure-ability; 9) seal-ability; 10) monitor-ability; 11) serve-ability; 12) non-splash-ability, and 13) kitchen-style-ability. These affordances describe high level functional requirements users expected a typical blender to incorporate in its design.

4.3 Generating the Function-Task Interaction Matrix

With the information obtained from sub-section 4.1, a function-task interaction matrix was created to determine potential interactions between technical functions and users' tasks. These interactions (multiple in some cases) represent the core of the FTIM and point to the functional affordances described. In the case study, 40 interactions representing 9 functional affordances were found, as presented in Fig. 10.

Most of affordances at the functional level should be identifiable in FTIM, but some will represent attributes of the device as a whole (in the case study, clean-ability; seal-ability, non-splash-ability, and kitchen-style-ability). This is because functional affordances can point to non-functional requirements when originating from users' determinations of undesirable and desirable product aspects, such as non-splash-ability and kitchen-style-ability respectively. Some of these requirements have cultural and social implications that are out of the scope of this study, but interestingly enough were left outside of the analysis which indicates that the matrix works as a tool to filter out non-functional requirements for separate analysis.

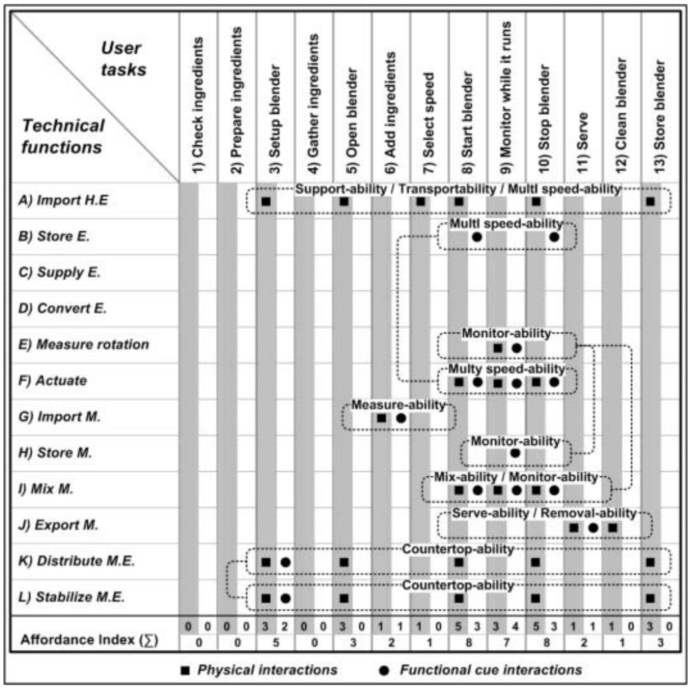


Figure 10. Function-Task Interaction Matrix of a typical blender.

Based on the FTIM above, several indexes are identifiable. A greater index indicates higher demand for affordances, since a larger number of technical functions must conform to a particular task. Careful evaluation of five tasks (*setup, open, start, monitor and stop*) results in the most relevant points to consider on a subsequent design or redesign of the blender device. With greater or fewer numbers of interactions, these indexes reflect the perception of affordance attributes. For instance, five attributes appearing in one technical function will more strongly suggest an affordance than having two attributes presented. Thus, the number of interactions in FTIM reflects how easily an affordance can be recognized. Table 4 shows these indexes.

Table 4. Affordance indexes.

User tasks	Technical functions	Affordance index (∑)
1) Check for ingredients	-x-	(0,0) = 0
2) Prepare ingredients	-x-	(0,0) = 0
3) ***Setup blender***	***A, K, L***	***(3, 2) = 5***
4) Gather ingredients	-x-	(0, 0) = 0
5) ***Open blender***	***A, K, L***	***(3, 0) = 3***
6) Add ingredients	G	(1, 1) = 2
7) Select speed	A, B	(1, 0) = 1
8) ***Start blender***	***A, F, I, K, L***	***(5, 3) = 8***
9) **Monitor while it runs**	**D, E, F, H, I**	**(2, 5) = 7**
10) ***Stop blender***	***A, B, F, I, L, M***	***(5, 3) = 8***
11) Serve	J	(1, 1) = 2
12) Clean blender	I, J	(2, 0) = 2
13) Store blender	A, K, L	(3, 0) = 3

4.4 Representing Functional Affordances

In addition to documenting the frequency with which users tasks intersect with technical functions, a new functional model diagram is developed to integrate functional affordances revealed within FTIM (Fig. 11) and expand the description of technical functions. Beyond improving the detail level of functional model diagrams, this additional information layer, based on functional affordances, provides subtask parameters for situations of use. In the blender example, different substances within the material flow define new uses and alternative technical function arrangement (see technical functions highlighted above).

This visualization also allows seeing functional affordances within functional affordances, a phenomenon described by Gaver [24] as nested affordances or affordances that are grouped in space, whether physical or abstract. In the blender example, the handle alone appears to afford holding. A jar alone may suggest an affordance for detaching due to its separation from the base, but not for the manipulation that will be effective in most use situations. Only by nesting the affordance for holding within the affordance for detaching the jar can be perceived as an object for serving. The affordances that did not fit within this diagram were again seal-ability, non-splash-ability and kitchen-style-ability.

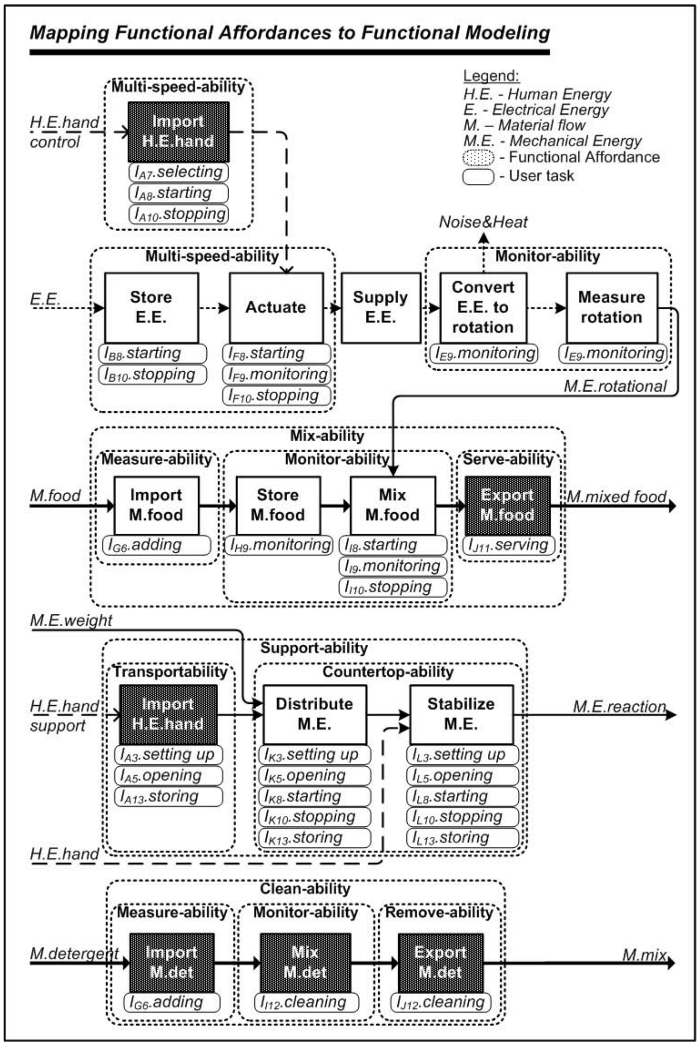


Figure 11. Functional Modeling with Functional Affordances.

Another way to integrate the functional affordances into the product architecture representation is to contrast technical functions with functional affordances using a layout diagram that sufficiently captures user-product interactions. As shown in Fig. 12, this representation provides an additional layer of information regarding the desired and undesired utility of each product part and highlights functional affordances which are not taken into account in current product architecture practices. Note that functional affordances not captured in FTIM and in the functional model are highlighted within this representation format with the exception of kitchen-style-ability.

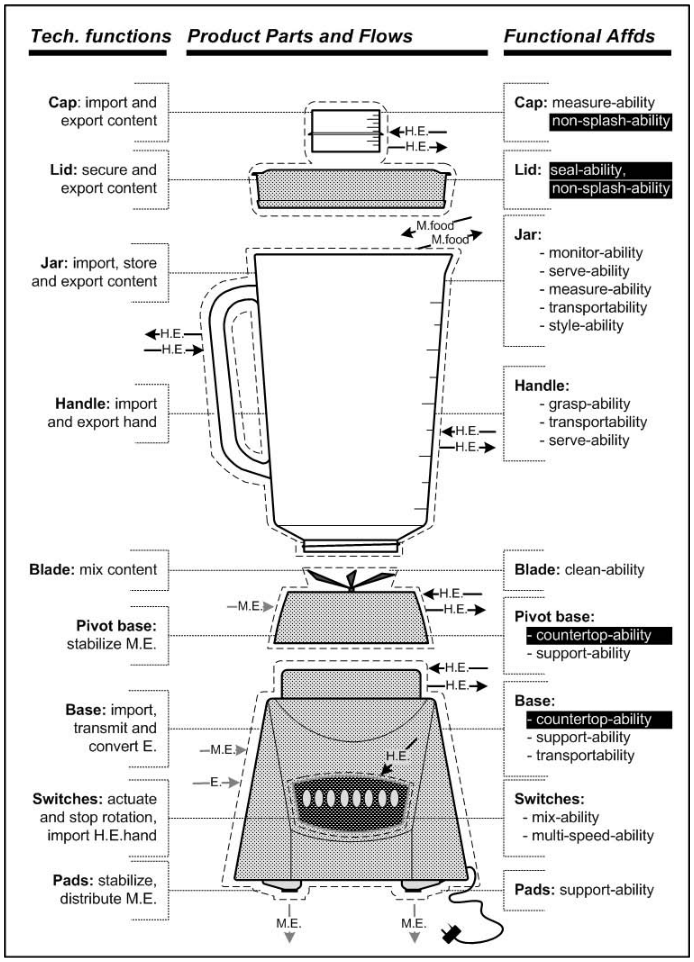


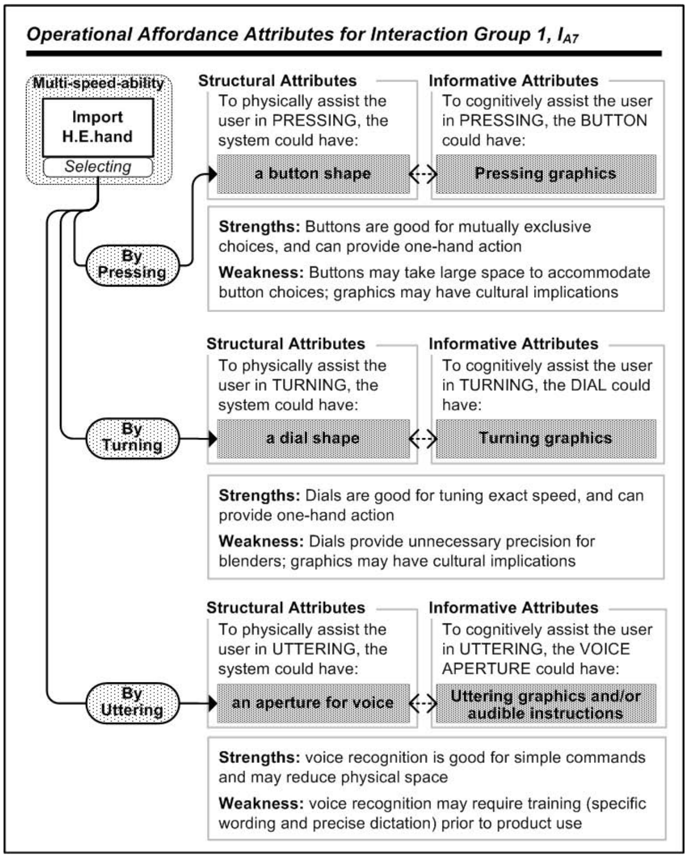
Figure 12. Function layout diagram with product boundaries, flows, and layout elements adapted from Van Wie et al [25].

4.5 Generating Operational Affordance Attributes

Shifting from a descriptive mode to a prescriptive mode, the last step is to capture affordance attributes in a practical manner. As described in sub-section 3.5, interactions from FTIM can help determining informative and structural attributes of the operational affordance level. The following example illustrates the interaction group 1 obtained from the first row of the blender's FTIM.

Interaction Group 1 = { $I_{A3}, I_{A5}, I_{A7}, I_{A8}, I_{A10}, I_{A13}$ ***}***
Subgroup Multi-speed-ability = { I_{A7}, I_{A8}, I_{A10} ***}***

When decomposing the tasks of interaction group 1, subgroup “multi-speed-ability” (*select speed, start blender, and stop blender*) into subtask options the following applies: ***In order to*** **“Import human command,”** ***the user has to*** **“Select speed”** ***by:*** **“pressing”, “turning”, “sliding”, and/or “uttering.”** Further, by matching each subtask with appropriate affordance attributes from the operational level, one can develop solutions that afford better product performance from user's perspective. In the case study, several blender concepts can be generated by considering multiple solutions for each technical function. The evaluation of these solutions is based on the strengths and the weaknesses of each structural and informative attributes. Figure 13 illustrates the process of finding these new combinations.



To physically assist the user in PRESSING, the system could have:

To cognitively assist the user in PRESSING, the BUTTON could have:

Strengths: Buttons are good for mutually exclusive choices, and can provide one-hand action

Weakness: Buttons may take large space to accommodate button choices; graphics may have cultural implications

To physically assist the user in TURNING, the system could have:

To cognitively assist the user in TURNING, the DIAL could have:

Strengths: Dials are good for tuning exact speed, and can provide one-hand action

Weakness: Dials provide unnecessary precision for blenders; graphics may have cultural implications

To physically assist the user in UTTERING, the system could have:

To cognitively assist the user in UTTERING, the VOICE APERTURE could have:

Strengths: voice recognition is good for simple commands and may reduce physical space

Weakness: voice recognition may require training (specific wording and precise dictation) prior to product use

Figure 13. Subtask options and related affordance attributes.

5. CONCLUSIONS AND FUTURE WORK

Correlating affordances to technical functions and to user tasks is a skill developers are rarely trained to do. A step-by-step methodology of how this association can be established during product architecture generation is presented in the form of two additional steps embedded in the original definition of Affordances. *Functional and Operational Affordance Levels,* introduced in this paper, provide abstractions to promote more acute awareness of user requirements when making product architecture decisions. The matching between the space of *Informative Attributes* and the space of *Structural Attributes* enable new products to incorporate careful evaluations of usability and usefulness.

By stripping away the ambiguity around the affordance concept during product architecture, the methodology gives developers more control over their future designs. The output of the methods proposed includes: 1) an exploration of users' tasks and subtasks, 2) an investigation on user-product interactions, 3) an indication of affordance demands on technical functions, 4) an introduction of visual means for representing functional affordances in product architecture, and 5) an exploration of solutions based on affordance attributes.

The blender case study, in identifying the affordance levels distinguished by task decomposition, highlights the importance of implementing a structured approach to the process of understanding affordances. It illustrates how technical functions can carry additional user requirements and how choices for usability, such as the size of controls or total weight of the device, need focused attention of developers. In addition, the FTIM from the case study was presented to entry-level design professionals who proposed only few modifications to the interaction groupings. These modifications were based on the information about the users' tasks involved, not detecting interactions. An extension of the case study can suggest appropriate future product configurations, especially for products that are merging functionality of other products (e.g., computerized kitchen appliances and cellular phones). This is because affordances levels provide understanding about users and how their requirements should be balance when merging products with distinct functionalities.

An additional step in the methodology needs to confirm with users which operational affordance attributes are most relevant for them so that developers can prioritize them during product architecture implementation. The next steps, in continuation of this research, include the development of means to automate parts of this methodology and systematically explore alternatives for manipulating this information. Another activity is to interview developers to assess their needs for management of that information. The use of a database to store affordance attributes and to provide a list of design solutions, similarly to a morphological analysis, can greatly improve and speed up this process.

6. ACKNOWLEDGEMENTS

The authors wish to thank the case study subjects for supporting the collection of data. The authors also thank Eric Swanson and Brianna Sylver for thoughtful insights. Adriano Braun Galvao is a Ph.D. student supported by the Brazilian National Council for Scientific and Technological Development (CNPq).

7. REFERENCES

[1] Ulrich, K., and Eppinger, S.D., 1995, *Product Design and Development*, McGraw-Hill Inc., New York, NY.

[2] Stone, R., Wood, K., and Crawford, R., 2000^a, "Using Quantitative Functional Models to Develop Product Architectures," Design Studies, 21(3):239-260.

[3] Maier, J.R.A., and Fadel, G.M., 2003, "Affordance-based Methods for Design," Proceedings of ASME DETC03/DTM-48673, 3:785-794, Chicago, IL.

[4] Derek, R., Sheppard, S., 1997, "The Role of Simulation Software in an Ideal Learning Environment," Proceedings of ASME DETC97/DTM-3891, Sacramento, CA.

[5] Galvao, A. and Sato, K., 2004, "Human-Centered System Architecture: A Framework for Interpreting and Applying User Needs," Proceedings of ASME DETC04/DTM-57545, Salt Lake City, UT.

[6] Swanson, E., Galvao, A., and Sato, K., 2003, "A Framework for Understanding Contexts in Interactive Systems Development," Proceedings of 7th World Conference on Systemics, Cybernetics and Informatics, 6:405-408, FL.

[7] Gibson, J., 1979, *The Ecological approach to Visual Perception*, Houghton Mifflin Company, Boston.

[8] Koffka, K., 1935, *Principles of Gestalt Psychology*, Harcourt Brace, New York, NY.

[9] Warren, W.H., 1984, "Perceiving affordances: Visual guidance of stair climbing", Journal of Experimental Psychology: Human Perception and Performance, 10;683-703.

[10] Norman, D.A., 1988, *The Psychology of Everyday Things*, Basic Books, New York, NY.

[11] Vera, A.H., and Simon, H.A., 1993, "Situated action: A symbolic interpretation," Cognitive Science, 17:7-48.

[12] Gaver, W., 1996, "Affordances for Interaction: The Social is Material for Design," Ecological Psychology 8(2):111-129.

[13] Markman, A.B., 1999, *Knowledge Representation*, Lawrence Erlbaum Associates, NY.

[14] Stone, R.B., Wood, K. L., and Crawford, R. H., 2000, "A Heuristic Method for Identifying Modules for Product Architectures," Design Studies, 21(1):5-31.

[15] Teeravarunyou, S., and Sato, K., 2001, "Use Process Based Product Architecture," Proceedings of World Congress on Mass Customization and Personalization, Hong Kong.

[16] Green, M.G., Tan, J., Linsey, J.S., Seepersad, C.C., and Wood, K.L., 2005, "Effects of Product Usage Context on Consumer Product Preferences," Proceedings of ASME Design Theory and Methodology Conference, Long Beach, CA, 2005. To appear.

[17] Hackos, J., and Redish, J.C., 1998, *User and Task Analysis for Interface Design*, John Wiley& Sons, New York.

[18] Suh, N., 1990, *The Principles of Design*, Oxford University Press, New York.

[19] Steward D.V., 1981, "The design Structure System: A Method for Managing the Design of Complex Systems," IEEE Transaction on Eng. Management, vol.EM-28, no.3, pp.71-74.

[20] Eppinger, Steven D., Whitney, Daniel E., Smith Robert P., and Gebala, D.A., 1994, "A Model-Based Method for Organizing Tasks in Product Development," Research in Engineering Design 6: 1-13.

[21] Kurfman, M., Rajan, J., Stone, R. and Wood, K., 2003, "Experimental Studies Assessing the Repeatability of a Functional Modeling Derivation Method," Journal of Mechanical Design, 125(4):682-693.

[22] Ulrich, K., and Eppinger, S.D., 2000, *Product Design and Development*, (2nd edition), McGraw-Hill, Inc., Boston, MA.

[23] Hubka, V. and Eder, W.E., 1988, *Theory of Technical Systems: A Total Concept Theory for Engineering Design*, Springer-Verlag , New York.

[24] Gaver, W., 1991, "Technology Affordances," Proceedings of CHI '91, ACM Press, pp.79- 84, New York, NY.

[25] Van Wie, M.J., Rajan P., Campbell, M.I., Stone, R.B., Wood, K.L., 2003, "Representing Product Architecture," Proceedings of ASME DETC03/DTM-48668, Chicago, IL.