

Toward automating affordance-based design

IVAN MATA, GEORGES FADEL, AND GREGORY MOCKO

Department of Mechanical Engineering, Clemson University, Clemson, South Carolina, USA

(RECEIVED March 1, 2014; ACCEPTED February 1, 2015)

Abstract

The objective of this research is to develop a computational representation of knowledge associated with affordance-based design (ABD). The ABD ontology formalizes the entities, properties, and relationships within the domains of ABD. The ontology enables designers to describe the affordances of existing products and specify the intended affordances of future products in line with ABD. The ontology consists of 14 concepts and 5 relationships. The ontology is developed using Protégé 4.3 and DL-query to query and reason with the ontology. The ontology is demonstrated using a consumer vacuum cleaner. The formal ontology serves as the basis for developing computer support for ABD applications. When implemented, these design tools will help designers manage the affordances of artifacts being designed, specifying the interacting entities of every affordance when a three-dimensional model of the artifact is available. Further, these software tools could be used to support ABD methods.

Keywords: Affordance; Affordance-Based Design; Affordance-Based Design Automation; Ontology

1. INTRODUCTION

The concept of affordances was introduced in the engineering design field by Maier and Fadel (2001) with affordance-based design (ABD). ABD is a systematic design methodology that focuses on the interactions between the user and the artifact being designed as well as the interactions between the components within the artifact. To expand the capabilities of ABD, software tools need to be created; particularly, software tools that aid in the embodiment and detail design phases of the design process. The creation of ABD software tools will allow designers to capture and share their design knowledge, enabling collaboration scenarios. Specifically, tools can be developed to gather affordance information from crowds. For example, users can be asked to assess the quality of affordances of a virtual prototype, providing information that can be used to evolve design solutions (Nguyen et al., 2012).

The reason to focus on the later stages of design is related to the dependence of affordances on physical embodiments, contrary to functions, which benefit from representing abstract transformations. Anticipating the creation of multiple design tools that achieve this automation, formal concept definitions need to be provided, and the different relationships between the concepts present in ABD need to be formalized.

An ABD ontology will provide the formal definitions found in ABD so that researchers have a common understanding of the domain concepts when creating their tools to achieve ABD computer support.

1.1. ABD

The term affordance originated in the field of perceptual psychology (Gibson, 1979). As Gibson defined it, “The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill.” It was created to describe what a system (e.g., an artifact) provides to another system (e.g., a user). Norman (1988) then extended the term to aid in the design of everyday things, but he stopped short of incorporating the concept of affordance as fundamental to the design of any artifact. Maier and Fadel (2001) introduced the concept of affordance as being fundamental to engineering design and defined it as a relationship between two subsystems in which potential behaviors can occur that would not be possible with either subsystem in isolation (Maier & Fadel, 2009b).

There are four basic entities associated with describing affordances in any design activity: the **artifact** being designed, the **user** of the artifact, the **environment** where the artifact is used, and the **designer** of the artifact. Affordances describe the interactions among artifact, user, and environment, whereas it is the goal of the designer to maximize the affordances that carry positive interactions between the design

Reprint requests to: Georges Fadel, Department of Mechanical Engineering, 202 Fluor Daniel Building, Clemson University, Clemson SC 29634, USA. E-mail: fgeorge@clemson.edu

entities while minimizing the effect of the affordances that carry negative interactions.

The designer–artifact–user entities form a complex system. The design entities and associated properties are specified, such that positive affordances are achieved and negative affordances are eliminated or minimized. The designer is capable of modifying the properties of the artifact by specifying a set of affordances; similarly, the designer can modify the affordances of an artifact by changing the properties of the artifact.

According to Maier and Fadel (2009a), affordances are composed of five key properties:

- *Complementarity*: This property tells us that an affordance has to have two interacting entities. The interaction of two design entities is represented by an affordance. The affordance cannot exist without the interaction of such entities.
- *Polarity*: Refers to the direction of influence of the affordance. Polarity may be used to distinguish if the affordance is a directed or undirected relationship. An artifact may cause harm to a user, the environment, or another artifact; this means that some affordances can also have a negative impact. There can be positive and negative affordances. For example, the cutting-ability of a knife is a negative affordance, because it can cut the user.
- *Multiplicity*: An artifact could potentially have multiple affordances.
- *Quality*: A measure of how well an affordance is achieved. For example, a chair and a briefcase both have the affordance of sitting-ability. It is expected that the sitting-ability of a chair has a higher quality than that of the briefcase.
- *Form dependence*: It is the structure of the artifacts (i.e., their geometry) that determines what they afford to a user. For example, the affordance of grasp-ability is present in a water bottle but not in a wall; the geometry of the water bottle allows the user to hold and manipulate it, but a wall does not allow such interaction with a user due to its geometry.

There are three types of affordances. Maier and Fadel (2009a) named two types of affordances: artifact–user affordances (AUAs) and artifact–artifact affordances (AAAs). To account for entities that are neither users nor artifacts (e.g., substance, medium, and natural objects), the environment is also considered a design entity in ABD (Hu & Fadel, 2012), and therefore, artifact–environment affordances (AEAs) were introduced to describe such interactions. AUAs are interactions between the user and the artifact in which properties of the artifact may be perceived to be useful to the user. In a similar way, AAAs are interactions between artifacts where such interactions are possible due to specific properties of each artifact. While interactions between user and environment may be important in some cases, the designer cannot control those

interactions, which is the reason user–environment interactions are not used in the ABD framework. The focus is the design of the artifact.

The AAA concept has not been fully endorsed by the affordance-based design community because the basis of affordances in perceptual psychology requires a user to perceive the affordances. This, however, does contradict some statements by Gibson (1979), which state that the user could be an agent that could be extended to be an artifact (e.g., a robot).

Although there has been some effort to define affordances, little progress has been made toward computational support tools. Specifically, affordances must be formalized in the context of engineering design.

1.2. Formalizing ABD knowledge using ontologies

Ontologies are explicit formal specifications of terms (i.e., knowledge) in a domain and relations among those terms (Gruber, 1995). An ontology defines classes that are concepts in a domain. Each class is defined by properties that describe features and attributes of the various concepts in a domain (Noy & McGuinness, 2001). Classes can also have subclasses, which allow the description of the hierarchy of the concepts in the domain to be specified. It is important to note that ontologies are abstractions of reality and, thus, do not represent all the details of reality.

Ontologies allow the sharing among people of common understanding of the structure of information (Noy & McGuinness, 2001). The reuse of domain knowledge is another important advantage of ontologies, and it becomes an important feature that will help achieve the ultimate goal of this research (i.e., automating the design process using ABD). Subsequent computational tools will be able to use previous knowledge bases and add functionality to make better tools. Ontologies have been used to formally capture knowledge in engineering design. For example, Fenves (2002) proposed a core product model for representing design information. The core product model representation consists of two main classes, the object and the relationship classes, from which different subclasses are defined to help describe the information generated and used in the product development process. Another example is a group of ontologies (Grosse, 2014) used for modeling, sharing, and integrating engineering design knowledge.

There are currently no ontologies that describe the concepts and relationships specific to the domain of ABD as described by Maier and Fadel (2009b). However, there are affordance ontologies in the field of ecological psychology (Turvey, 1992; Sanders, 1997). These ontologies model the existence of the concept of affordance. They are not engineering ontologies where classes and their properties are specified; hence, they are not useful for engineering design applications. The proposed ontology does not formalize all knowledge associated with affordances. The world of knowledge of this ontology is ABD.

A related affordance ontology that describes common household objects in terms of their affordance features (structure, material, and grasp features) has been proposed to enable robots to recognize objects based on their specific geometries (Varadarajan & Vincze, 2012) from visual data. This ontology would not help in engineering design, where any type of object needs to be described and all the interacting entities need to be specified. Ortmann and Kuhn (2010) proposed an affordance ontology based on an ontology of observations, where affordance is considered a quality, a subclass of quality in the ontology (the affordance does not describe interacting entities in their ontology). Quality (attribute or characteristic) is used as a broad concept to define things that could be physical, temporal, or abstract. Affordances in ABD are not seen as qualities; they are seen as relationships between design entities, which have qualities (the affordance defines the interacting entities). This difference in the concept makes these ontologies incompatible with an ontology that uses ABD concepts.

2. DEVELOPMENT OF THE ABD ONTOLOGY

A guide for extracting and formalizing knowledge and developed ontologies as proposed by (Noy & McGuinness, 2001) was followed for the creation of the ABD ontology (ABDO). As the authors of the guide mention, the creation of an ontology is an iterative process. The ABDO may not be comprehensive. It is meant to be updated regularly, depending on the applications needed to achieve ABD computational support. The ABDO defines the core structure of affordances for any design situation, and at this stage the ABDO is not a repository. There are more methodologies for creating ontologies. Ahmed et al. (2007) proposed a six-step methodology to develop engineering design ontologies for the purposes of indexing, searching, and retrieving design knowledge. However, this methodology makes use of the concept of function. Functions are often represented as a transformation of mass, energy, or signal and cannot describe nontransformative aspects of a design. It is worth mentioning that these methodologies have similar steps toward creating ontologies; even though the latter is specific to engineering design, the former was used to create the ontology for reasons previously mentioned. Ahmed et al.'s ontology does contain elements that could be added to the ABDO ontology in the future, for example, the inclusion of a class that describes the different ABD stages.

The concept of affordances comes from the field of perceptual psychology, but the ontology does not necessarily need to be applicable to ecological psychology only. As Sanders (1997) points out, there are multiple levels of ontologies that might be generated. The structure of the ontology depends on the domain it will represent, which in this case is the engineering design domain as described by Maier and Fadel (2001) using the concept of affordances. Other affordance-based design approaches can be addressed in the future by expanding the ontology.

The ABDO is developed primarily to support the analysis and refinement of affordances (Maier & Fadel, 2009b) phase.

It will help design database schemes that will be applied in computer software to keep track of the affordances of artifacts and their entities involved, making it possible to see which affordances get affected due to changes in the geometry of a product and to look at the affordances associated with assemblies, subassemblies, or components. The ontology could also be used to create repositories of affordances, adding records to describe newly discovered affordances in the design process.

The ABDO is designed to answer these types of questions:

- What affordances, if any, are associated with this component/assembly/subassembly?
- What are the entities that a certain affordance is related to?
- What type of affordance is this?
- What affordances will be affected if the geometry of the artifact is changed?
- Can affordances be listed by priority/type-of-user?

To identify affordances, a description of the architecture of the artifact being designed is needed. The artifact description in the ABDO has the information needed to list the different components of the artifact and describe assemblies and subassemblies, because affordances describe not only interactions between individual components but also interactions where a group of components could provide an affordance. The description of the architecture (i.e., form) does not include other physical properties, such as temperature, density, specific heat, or electrical conductivity. This information can be captured as data properties associated to parts in future iterations of the ontology.

The concepts and their relationships included in this proposed ABDO were taken from the literature without questioning their adequacy with respect to other views on designing with affordances; therefore, this ontology is only meant to be applied when designing artifacts under the ABD methodology described by Maier and Fadel (2009b). The ABDO provides a formalized structure for representing affordances related to a designed artifact. The ontology does not identify or formalize the affordance; that is the task of the designer.

2.1. ABDO classes

Classes in ontologies represent concepts in a domain; each concept can be described by its hierarchical level with respect to other concepts and by its properties that define it. The classes of the ontology were determined by listing all the concepts that are relevant to ABD. A top-down approach was followed to identify the hierarchy of the concepts, meaning that the most general concepts in the domain were defined, and then more specific terms were selected as subclasses if the concepts were related. Classes of the ABDO are shown in the first column of Table 1. All main classes are attached to the class called “thing,” as is customary in ontology structures. The classes represent important concepts within ABD. Each class can be

Table 1. *ABDO classes and their properties*

Classes and Subclasses	Object Properties	Data Properties
Affordance	hasPrimaryEntity	affordancePhrase
Artifact artifact affordance	hasSecondaryEntity	
Artifact user affordance		polarity
Artifact environment affordance	hasContributor	priority
		quality
Artifact	hasParts	partDescription
Assembly	hasAssembly	assemblyDescription
Part		material
User		characteristics
Human		
Nonhuman		
Environment		characteristicsEnv
Medium		
Natural object		
Substance		

described with more concepts, and these concepts are attached to classes as object and data properties.

The ontology was created with the open source ontology editor Protégé 4.3. The ontology can be accessed from the URL (Mata et al., 2014). The classes and their properties within ABDO are presented in Table 1 and explained next.

2.2. Affordance class

An instance of an *Affordance* is a description that represents what one instance (the primary entity) provides to another instance (the secondary entity). The primary entity and the secondary entity are related to the instance of *Affordance* using OWL object properties (*hasPrimaryEntity* and *hasSecondaryEntity*, respectively, with domain *affordance*). An *Affordance* instance is fully defined when its *affordance phrase* (i.e., how affordances are represented in words), its *polarity*, *priority*, *quality*, and relating entities are specified; these characteristics become the properties of an *affordance*.

The subclasses are more specific types of affordances. There are three different types of affordances (Fig. 1), where each type is defined according to the pair of interacting entities. *Artifact-artifact* affordances (AAA) describe an *affordance* between two artifact instances that can be parts or assemblies. *Artifact-user* affordances (AUA) describe an interaction between an instance of artifact and an instance of user, and *Artifact-environment* affordances (AEA) are the interactions between an artifact instance and an environment instance.

The *Affordance* parent class has three object properties. The first one is called *hasPrimaryEntity*. Because this property is attached to instances of the *Affordance* class, its domain is the *Affordance* class, and because every *Affordance* instance has to have an *Artifact* as one of its interacting entities, the range of this property is set to

instances of the *Artifact* class only. The second object property is called *hasSecondaryEntity*. Its domain is the *Affordance* class, and its range is set to accept an instance of the *Artifact*, *Environment*, or *User* classes. For example, when designing a steering wheel, the *affordance* of “grasp-ability” describes the interaction between the user and the ring of the steering wheel: the primary entity of this *affordance* is the ring of the steering wheel, an artifact, and the secondary entity of this interaction is the user, which makes “grasp-ability” an artifact–user *affordance*. The third object property is called *hasContributor*. The contributor property specifies entities that have some influence on the *affordance*. For example, the *affordance* of *maneuverability* of a vacuum cleaner would specify the handle of the vacuum cleaner and the user as the interacting entities. The user physically interacts with the handle of the vacuum cleaner to use its *maneuverability*, but the wheels of the vacuum cleaner have an influence on this *affordance*. The wheels in this case are considered to be contributors.

The subclasses inherit the properties mentioned earlier, but each *Affordance* subclass redefines the second object property (*hasSecondaryEntity*) according to the type of *affordance*. For example, subclass AAA defines this property’s range with the *Artifact* class, while subclass AUA defines it with the *User* class and subclass AEA does it with the *Environment* class.

One of the data properties of the class is called *affordance-Phrase*. There are three formats used to represent an *affordance phrase*, including “verb + -ability” (e.g., grab-ability, twist-ability), “verb + noun (or noun + verb) + -ability” (e.g., lift handle-ability, rotate gear-ability), and “transitive verb + noun (or intransitive verb)” (e.g., collect water, lubricate part; Hu & Fadel, 2012). The ontology does not define any format, so that designers are able to choose their preferred way to represent *affordance phrases*. The domain of this property is the *Affordance* class, and its range is set to be a string data type so that anything can be written for the phrase with no format restriction. The lack of format restriction might become an issue later on, and research will be necessary to identify problems. If problems arise, standardization should be implemented.

Another property is called *polarity*. Every *affordance* could be positive or negative: if the *affordance* describes an interaction where one of the entities is harmed, then the *affordance* is negative. The domain is again the *Affordance* class and the range is set to be a string value: “Negative” meaning it is a negative *affordance* and “Positive” meaning it is positive.

The *priority* property informs the user how important the *affordance* is compared to the other *affordances* of the project. The domain is the same as the rest of the properties, and its range is set to an integer value. The scale is currently not set, but it could be, for example, integer values from 1 to 5, 1 being the highest priority. The importance of having a priority value for each *affordance* is to be able to list the *affordances* by level of priority, to easily identify the most important *affordances*. The priority could be set when the *affordance* is

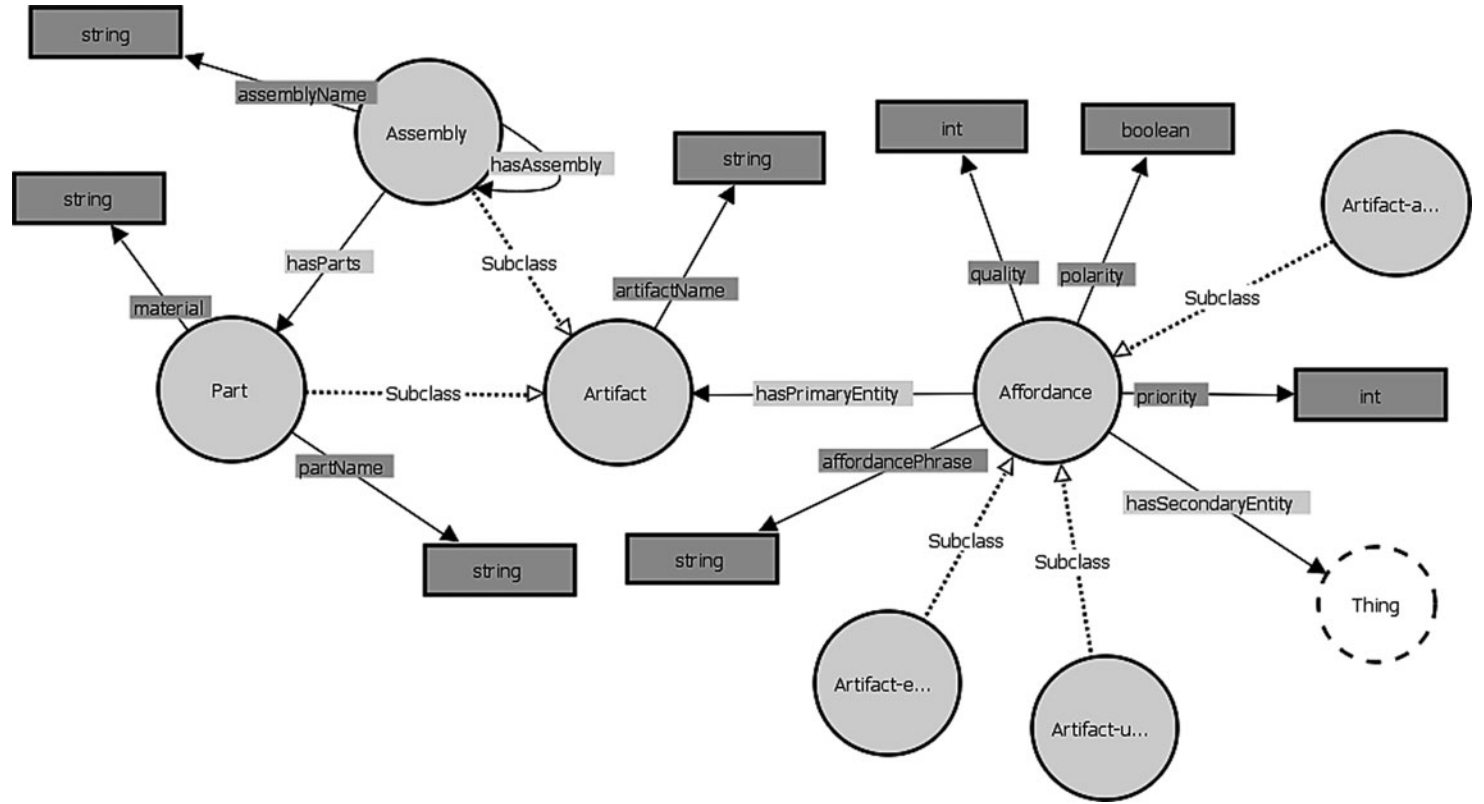


Fig. 1. ABDO affordance and artifact class.

identified; it could also change during the design process as new affordances are identified.

The last data property that defines an affordance is *quality*. The range for this property has been set to integer values. These values could vary from 0 to 3. A value of zero indicates that the quality of the affordance is bad, and a positive number indicates good quality. The scale proposed here is linear, based on current research the authors are conducting. Note that quality is subjective; the value assigned to every affordance might differ between designers.

2.3. Artifact class

An important concept in ABD is that of the artifact, an artifact is a human-made object. In the ontology, an *Artifact* is an assembly, a subassembly, or a part. The object being designed is typically a group of subassemblies, each comprising different parts, so the subclasses of *Artifact* are *Assembly* and *Part* (Fig. 1). The reason assemblies are considered is because there are some cases where groups of physically related parts work together to provide an affordance; for example, many components allow “turnability” of a car to a driver, not just the steering wheel.

The object properties in this class are *hasParts* and *hasAssembly*. The property *hasPart* has the *Assembly* class as its domain, and its range is set to the *Part* class. It describes how an assembly is composed of many instances of the *Part* class. The *Part* class does not have this property because its instances are components of the artifact that can no longer be decomposed (e.g., a screw). The property

hasAssembly has the *Assembly* class as its domain, and its range is set to the *Assembly* class.

A data property linked to the assembly subclass called *assemblyDescription*, with its domain being the *Assembly* class and its range being a string value, provides a brief description of the assembly. In a similar manner, the part subclass has a property called *partDescription*, with its domain being the *Part* class and its range being a string value.

The class *Part* has a property that defines the *material* of the part, with its domain being the *Part* class and its range set to be a string value. *Manufacturing process*, *mass*, and *geometry* of a part are properties that will be added to the ontology in the future, but for the purposes currently outlined, this is not a necessity.

2.4. User class

The *User* class defines users who interact with the artifact at any point of its lifecycle. The subclasses are *Human* and *Nonhuman* (Fig. 2). The *Human* class is further broken down into subclasses to specify the user that interacts with the artifact at specific stages of the artifact’s lifecycle (end user, maintenance, manufacturing, transportation, etc.). Currently, the class only has one property, a data property called *characteristics* that defines specific characteristics of a user. For example, the data property could contain information about a disability that the user might have: information that could help the designer identify unconsidered affordances. The *characteristics* property has a domain of *User* class instances and a range set to be string values.

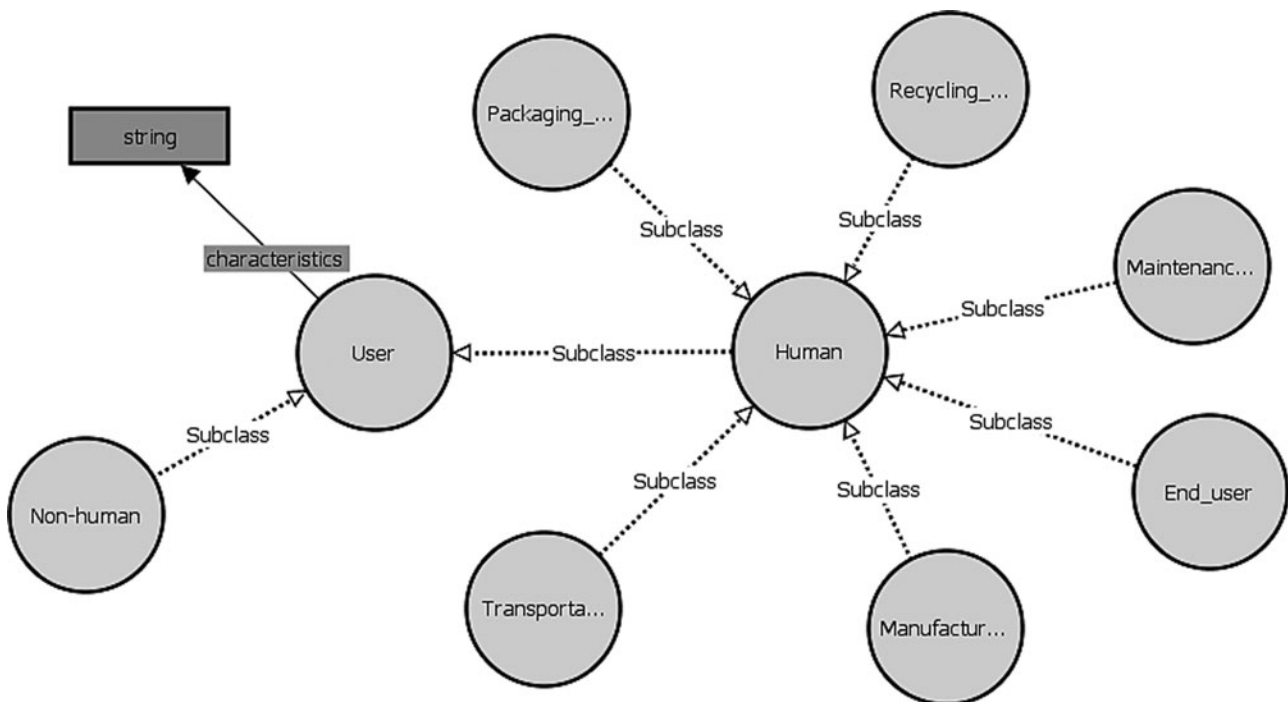


Fig. 2. ABDO user class.

2.5. Environment class

The Environment class defines interacting entities that are neither a user nor an artifact; its subclasses are Substance, Medium, and Natural objects (Fig. 3). Substance is a particular kind of matter with uniform properties. Medium is a substance that makes possible the transfer of energy from one location to another and that surrounds the artifact being designed. There is a similarity between substance and medium. To make the concepts clear: if a thing surrounds the artifact, then it is considered a medium. For example, water is the medium when designing a tidal stream generator, but it is a substance when designing a vacuum cleaner. Natural objects are things in the environment that are not human made and that come from nature. All the instances of the Environment class can be defined by their characteristicsEnv property. Because there are many possible objects that could fall under the environment class, each can be described with its characteristics. For example, if designing a tidal stream generator, the water medium can be characterized by its specific weight, viscosity, salinity, and average temperature.

3. EXAMPLE APPLICATION OF ABDO

Having detailed the ABDO, it is used and demonstrated on the design of a household vacuum cleaner. The components, assemblies, and affordances were added to the ontology as individuals using the Protégé 4.3 ontology editor. There are a total of 11 affordances, 3 assemblies, and 15 parts. Table 2 shows the components of the vacuum cleaner and its affordances as ontology individuals and the classes they belong to.

Each individual is defined by its object and data properties. Due to space limitations, examples of only two individuals are shown in Table 3 along with their properties' values.

In addition to representing the knowledge associated with the vacuum cleaner, the description logic query language (DL Query) is used to answer the questions presented in Section 2. Table 4 shows the queries that were executed, the questions

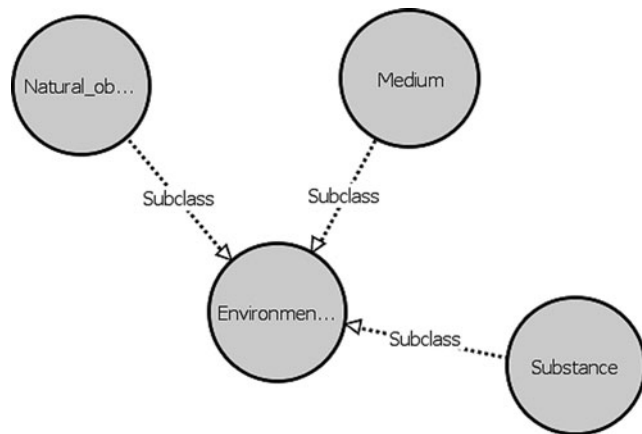


Fig. 3. ABDO environment class.

Table 2. Vacuum cleaner components and its affordances

ABDO Class	Individual
Affordance	Multiple_surface_reachability
Artifact artifact affordance	Scrubbingability
Artifact user affordance	Maneuverability
Artifact environment affordance	Auditory disturb
	Electric_shockability
	Filter_accessability
	Visual pleasing
	Filterability
	Cloggingability
	Power_consumption
	Suction_capability
Artifact	Vacuum
Assembly	Motor
Part	Impeller assembly
	Handle
	Filter
	Hose
	Indicator light
	Belt
	Storage cover
	Surface
	Wheels
	Bumper
	Power button
	Sound proofing cover
	Power cord
	Brush cover
	Brush
	Filter bag
User	User
Human	
Nonhuman	
Environment	Electric energy
Medium	Dirt
Natural object	
Substance	

that queries answer, and the results of the queries. Note that each query can be modified to provide more information.

Some queries could potentially answer different questions. For example, the first query shown in the table answers two types of questions. This query can provide the affordances associated with a specific artifact to check how important an artifact is based on the number of affordances it is related to. If the designer decides to change the geometry of an artifact, this query can also show the affordances that will be affected due to the change. Making the designer aware of these dependencies could restrict changes in the geometry of the artifact, so that the change is targeted to improve a specific affordance and not all of the affordances the artifact is associated with.

The second query can be useful when separating the users involved at each lifecycle stage of the product. For example, the designer could check the affordances of the product that are relevant to the packaging type of user. Focusing on that set of affordances can help the designer identify

Table 3. Example ontology individuals and their properties

Ontology Individual	Property and Value
Cloggingability (affordance)	Object properties
	hasPrimaryEntity: Hose
	hasSecondaryEntity: Dirt
	hasContributor: Filter bag
	hasContributor: Brush
	hasContributor: Impeller assembly
Belt (artifact)	Data properties
	affordancePhrase: Cloggingability
	polarity: Negative
	quality: 3
Belt (artifact)	Data properties
	material: Rubber
	partDescription: Belt that drives brush

more affordances to improve the interactions between the packaging user and the product.

One of the goals of the designer when using ABD is to reduce the effects of negative affordances. The ABDO can be used to query all the negative affordances and their related entities. By identifying the negative affordances and the entities involved, the designer can change the geometry or topology of the artifacts, so that the effect of such negative affordances can be reduced.

The ABDO can answer design questions, but it is not expected that designers will use it directly in their design

projects. The ABDO will serve as the conceptual basis for computer applications that aid designers when designing products with ABD.

4. FUTURE WORK DEVELOPMENT

The concepts and relationships in ABDO can be used to create a software tool that manages the affordances of the project in later stages of the design process, that is, when the optimal embodiment is determined based on the analysis of the product's affordances. This tool would allow designers to assign the affordances of the artifact being designed and associate the entities that each affordance describes when a three-dimensional model is available.

Multiple designers can be expected to be working on a single project simultaneously, so a web-based application would make it easier for designers to access the project affordance management system. Databases are used in applications like these. An ontology is generally used to share knowledge about a specific domain (Noy et al., 2001), but they cannot be directly applied to database systems. However, due to their similarities with database models, ontologies could be used to create the database schemes needed in information systems design.

As the artifact evolves in the design process, the architecture and the affordances can change. The affordance count can increase if the geometry changes and additional affordances are identified. Databases could easily handle this scenario, warning the designers if adding an affordance affects other affordances by querying related components. Database

Table 4. ABDO DL Query commands

Question	Query (DL Query)	Results (Ontology Individuals)
What affordances are associated to the Hose? Or What affordances will be affected if the geometry of the Hose is changed?	Affordance and (hasPrimaryEntity value Hose) or (hasSecondaryEntity value Hose)	Multiple_surface_reachability Cloggingability
What are the affordances associated to end users?	Affordance and hasSecondaryEntity some End_user	Maneuverability Auditory disturb Electric-shockability Filter-accessibility Visual pleasing
Can affordances be listed by priority?	Affordance and priority value 1	Maneuverability Electric-shockability Cloggingability Power_consumption Suction_capability
	Affordance and priority value 2	Auditory disturb Multiple_surface_reachability Visual pleasing
What are all the negative affordances?	Affordance and polarity value "Negative"	Auditory disturb Electric-shockability Cloggingability Power_consumption
What are the entities associated with the affordance of Cloggingability?	(inverse hasPrimaryEntity value Cloggingability) or (inverse hasSecondaryEntity value Cloggingability)	Hose Dirt

systems could warn the designer which affordances might be affected due to changes to the product geometry. A database scheme created from the ABDO will provide information about the interacting entities for each affordance, which is valuable information for the designer whenever modifying the architecture of the artifact. To aid the designer throughout the embodiment phase of the design process, a web-based database implementation with the ABDO information can be created.

There are many more helpful ways in which the ABDO can be implemented in a design tool besides the functionality mentioned earlier. One of these applications is using the database to automatically create and populate an affordance structure matrix (Maier & Fadel, 2007).

The ASM is a concept exploration and attention directing tool where the different affordances of an artifact are correlated with the different components of the artifact (Maier & Fadel, 2007); the affordances are placed on the rows of the matrix and the components of the artifact are placed on the columns. The more relationships an affordance has with the components of the artifact, the more important it is to the design of such an artifact, alerting the designers to improve the affordances that have low scores.

ABDO provides the concepts and their relationships, so that developers can use them in a consistent manner, making different software applications compatible. Database information could be shared and used for different applications.

5. CONCLUSION

The creation of an ABDO is needed to share the knowledge about the concepts and their relationships within ABD and to anticipate the development of computer software with a focus on assisting the design process using the concept of affordance. Having an ontology is one of the initial steps toward software support of the affordance-based design process.

The application of the ABDO is not just for sharing the knowledge of a domain. As demonstrated in this paper, it can be used to answer design questions; these questions can vary according to the needs of the designer. The value of designing with structured ontology information is that there would be no need to ask an expert on the subject; the use of its concepts would be consistent throughout the different applications created.

REFERENCES

- Ahmed, S., Kim, S., & Wallace, K.M. (2007). A methodology for creating ontologies for engineering design. *Journal of Computing and Information Science in Engineering* 7(2), 132–140. doi:10.1115/1.2720879
- Fenves, S.J. (2002). A core product model for representing design information. In *NISTIR 6736*. Washington, DC: National Institute of Standards and Technology.
- Gibson, J.J. (1979). The theory of affordances. In *The Ecological Approach to Visual Perception*. Hillsdale, NJ: Erlbaum.
- Grosse, I.R. (2014). Ontologies for engineering knowledge modeling, management, and sharing. *eDesign UMassAmherst*. Accessed at <http://edesign.ecs.umass.edu/ontology-downloads/>
- Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human–Computer Studies* 43(5–6), 907–928.
- Hu, J., & Fadel, G.M. (2012). Categorizing affordances for product design. *Proc. ASME IDETC/CIE 2012*, pp. 1–15. Chicago: ASME.
- Maier, J.R.A., & Fadel, G.M. (2001). Affordance: the fundamental concept in engineering design. *Proc. ASME IDETC/CIE 2001*, Paper No. DETC2001/DTM21700, Pittsburgh, PA.
- Maier, J.R.A., & Fadel, G.M. (2007). The affordance structure matrix—a concept exploration and attention directing tool for affordance based design. *Proc. ASME IDETC/CIE 2007*, pp. 1–11, Paper No. DETC2007-34526, Las Vegas, NV.
- Maier, J.R.A., & Fadel, G.M. (2009a). Affordance based design: a relational theory for design. *Research in Engineering Design* 20(1), 13–27.
- Maier, J.R.A., & Fadel, G.M. (2009b). Affordance-based design methods for innovative design, redesign and reverse engineering. *Research in Engineering Design* 20(4), 225–239.
- Mata, I., Fadel, G.M., & Mocko, G. (2014). *Affordance based design ontology (ABDO)*. Accessed at http://people.clemson.edu/~gmocko/ontology_public/Affordance_Ontology.owl
- Nguyen, M.T., Guarneri, P., Fadel, G.M., & Mata, I. (2012). Genetic algorithms applied to affordance based design. In *ASME IDETC/CIE 2012*, pp. 1–7, Paper No. DETC2012-70332, Chicago.
- Norman, D.A. (1988). *The Design of Everyday Things*. New York: Currency Doubleday.
- Noy, N.F., & McGuinness, D.L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford, CA: Stanford Knowledge Systems Laboratory.
- Ortmann, J., & Kuhn, W. (2010). *Affordances as Qualities*. Munster, Germany: University of Munster, Institute of Geoinformatics.
- Sanders, J.T. (1997). An ontology of affordances. *Ecological Psychology* 9(1), 97–112.
- Turvey, M.T. (1992). Affordances and prospective control: an outline of the ontology. *Ecological Psychology* 4(3), 173–187.
- Varadarajan, K.M., & Vincze, M. (2012). AfRob: the affordance network ontology for robots. *Proc. 2012 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1343–1350. Vilamoura, Algarve, Portugal: IEEE.

Ivan Mata is a PhD student in mechanical engineering at Clemson University. He holds a BS in mechanical engineering from Central American University. Mr. Mata focuses on design theory (ABD). He is a member of ASME.

Georges Fadel is a Professor and Exxon Mobil Employees Chair in Mechanical Engineering at Clemson University. He holds a PhD in mechanical engineering and an MS in computer science from Georgia Tech., and a Diploma in mechanical engineering from ETH Zurich. Dr. Fadel focuses on packaging or layout optimization, material by design, additive manufacturing, and design theory (affordances). He has published over 250 refereed research articles. He is Fellow of ASME and past chair of its Technical Committee on Design Automation; a member of AIAA, SAE, ISSMO, and MCDM; and on the Board of Management of the Design Society.

Gregory Mocko is an Associate Professor of mechanical engineering at Clemson University. He is the Coordinator of the Mechanical Engineering Capstone Design Program. Dr. Mocko earned a PhD from Georgia Tech., an MS from Oregon State University, and a BS from the University of Connecticut. His research interests include knowledge representations for design and manufacturing and concept generation in mechanical design. His research has been funded by the US National Science Foundation, NASA, BMW, and US Army TACOM. He has published over 60 peer-reviewed articles.