Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2017
August 6-9, 2017, Cleveland, OH, USA

DETC2017-67467

DESIGN HEURISTICS: A CONCEPTUAL FRAMEWORK AND PRELIMINARY METHOD FOR EXTRACTION

Bumsoo Lee
Georgia Institute of Technology
Atlanta, GA, USA

Kenton B. Fillingim
Georgia Institute of Technology
Atlanta, GA, USA

William R. Binder
Georgia Institute of Technology
Atlanta, GA, USA

Katherine Fu
Georgia Institute of Technology
Atlanta, GA, USA

Christiaan J. J. Paredis
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

In designing complex systems, systems engineers strive to turn an existing situation into a situation that is most preferred. A rational decision maker would choose the alternative that maximizes the expected utility of the existing situation, but there are significant computational challenges to this approach. To overcome these challenges, most decision makers revert to heuristics. In this paper, we propose a conceptual framework for heuristics in design. A preliminary empirical study of designers for a robotics design problem was conducted to observe how participants apply heuristics. Participants having at least 2 years of experience designing robots were recruited to partake in a robotics design session in which participant were given 45 minutes to work on a design problem. A preliminary heuristics extraction method was developed, and the identified heuristics were studied to find trends within the overall set. These trends were the basis of a taxonomy of heuristics consisting of three initial classification methods: design phase, field of study, and action intent. The heuristics and classifications are presented, along with the challenges from extracting heuristics and recommendations for future work to further research design heuristics and to improve the method for extraction.

1 INTRODUCTION

To introduce the concept of heuristics and why they are an integral part of design, we start the story with Herbert Simon, who in his seminal work on the Sciences of the Artificial [1], indicates the key objective of design: "Everyone designs who devises courses of action aimed at changing existing situations into preferred ones." Framed slightly more strongly, we could rephrase this in the context of systems engineering and design (SE&D): *Systems engineers and designers should strive to change an existing situation into the situation that is most preferred.*

Building on the mathematics of decision or choice theory [2, 3], the extent to which a situation is preferred, can be measured as *value*. If a situation A is more preferred to a situation B, it is assigned a higher value, so that the most preferred situation is the one that maximizes value. Decision theory clarifies further that one must also take into account the time and the risk preferences. Starting from four simple axioms, von Neumann and Morgenstern [3] proved that a rational decision maker chooses the alternative that maximizes expected utility, where a utility is a nonlinear transformation of value constructed such that risk preferences are accounted for by taking the expectation. Time preference is captured mathematically using a discount function. Combined, this allows us to express the objective for systems engineering and design as the following equation [4]:

$$\mathcal{A}\text{: } \max_{a \in A} \mathrm{E}[u(NPV(a, t(\mathcal{A}), C(\mathcal{A})))] \quad (1)$$

In other words, a designer must search over the set of all artifacts, $\mathcal{A}$, for the artifact, a, that maximizes the designer's expected utility of the Net Present Value (NPV). Notice that the NPV depends not only on the value we expect to derive from using, trading, or selling the resulting artifact, but also on the time, $t(\mathcal{A})$, and the cost, $C(\mathcal{A})$, needed for the search/optimization process, that is, the cost and the time of design and development.

The challenge with this framing of an SE&D problem is that the optimization problem in Equation (1) cannot (and should not) be solved in a mathematically rigorous sense, that is, by using optimization algorithms to find the mathematically guaranteed global optimum. The set of all artifacts would require an infinite number of parameters to describe mathematically, and the analysis of all these artifacts would require an infinite amount of time. In addition, because the time and cost of searching affect the objective, the designer must carefully balance the value of the resources invested in the search process with the value of the artifact. At some point, continuing to search will cost more than it's worth.

In artificial intelligence and operations research, such computational complexity challenges are overcome by using heuristics. Heuristic search sacrifices guarantees of optimality and completeness of the solution set for increased solution speed [5]. Similarly, in design, a heuristic is a rule of thumb that provides guidance for choosing what action to pursue, given the current state of the design process. Design heuristics rely on experience and knowledge to suggest actions that provide a good tradeoff between the cost of the SE&D process and the value of the resulting artifact.

We use the term "heuristic" broadly here. For simple detailed design decisions, a heuristic may directly constrain the artifact alternative. For example: "When designing a sheet-metal hem, the hem length should be at least four times the sheet metal thickness." For more important decisions that strongly affect the value of the artifact, a heuristic may specify a sequence of design steps for how to constrain the artifact alternative, where each step in the sequence involves additional heuristics. For instance, a heuristic may suggest framing the design decision as an optimization problem across a heuristically defined design-space parameterization and heuristically suggested analysis-model approximations and idealizations. Finally, heuristics could also embody planning guidance, as in a heuristic suggesting how to decompose a high-level goal into sub-goals. In all three cases, the heuristic knowledge reflects previous experiences regarding the value-of-information tradeoffs [6, 7] between the accuracy and cost of approximating Equation (1) in the specific design context encountered. The resources allocated to a particular design choice should be commensurate with the potential impact the choice has on the artifact value.

There is poor agreement over how humans actually use and select heuristics. This is often the case because heuristics are the result of experience, and users may use them without being consciously aware of the heuristics. Even for users that acknowledge their use of heuristics, describing the heuristics can be challenging. Individuals typically perform on a relatively closed set of examples, such as the design of pressure vessels. Those designers will likely use heuristics that may work in other scenarios, but because of their experience they cannot describe, or do not believe the heuristic applies in other scenarios. This presents a challenging task for research about how heuristics are currently employed.

In this paper, we propose a conceptual framework for heuristics in design after reviewing relevant literature. We conducted a preliminary empirical study of designers for a robotics design problem to research how users apply heuristics. We then developed a method for extracting heuristics from the results of the survey in order to develop a taxonomy for the heuristics. The focus of this paper is not to present the heuristics themselves, but the method used for extraction. There are many papers that present heuristics, but our heuristics are simply a byproduct of the method we are trying to design. Finally, we discuss the results and challenges of our investigation and recommend future work to further research into design heuristics. The research questions are: Can the use of heuristics be justified from a normative decision theory perspective? How do human designers use design heuristics? What is a repeatable method for extracting heuristics from design observations?

2 RELATED WORK

For the duration of this study, a formal definition for all heuristics should be identified as a reference point for heuristics. Currently, there is no standard definition for a heuristic being used in literature. After studying how design principles are expressed in current literature, Fu et al. provides a formal definition of a heuristic.

Heuristic: A context-dependent directive, based on intuition, tacit knowledge, or experiential understanding, which provides design process direction to increase the chance of reaching a satisfactory but not necessarily optimal solution.

Example Heuristic: "A properly designed bolt should have at least one and one-half turns in the threads" (adapted from [8]).

Another example of heuristics used in design is Altshuller's TRIZ [9]. TRIZ offers a method to solve design conflicts between multiple parameters in the design process. Based on context (the two conflicting parameters), one or more of 40 directives are presented to increase the chance of reaching a satisfactory solution to the conflict.

What is not considered a heuristic? Principles such as "F = ma" are not considered heuristics. Fu et al. also breaks down what separates design heuristics from a design principle or guideline. In comparison to design principles and guidelines, design heuristics are generally less formalized with the least amount of supporting evidence or experimental validation. Heuristics are more prescriptive and offer a certain level of reaching a successful solution. The prescriptive design heuristic should be stated in the grammatical imperative form, include a prescriptive action for the designer to take, and increase the likelihood of reaching a desirable consequence. On the other hand, principles and guidelines are more descriptive and do not have specific attributes regarding success [10].

In similar studies of heuristics extraction performed by Yilmaz, extraction methods consisted of two coders analyzing sketches and verbal reasoning from study participants [11]. The coders performed independent analyses, and the final set of heuristics were presented when the coders were in mutual agreement. Coding identified each concept and documented when change occurred from one concept to the next [12]. The action or process initiating such change was considered a potential heuristic, and then generalized in a way that the heuristic could be applied to other design contexts. Coders also recognized heuristics by identifying unique design features not shown in the concepts of other participants [13].

A retrospective interview allowed participants to explain their design process and provide clarification to why they made certain decisions in the study. In other protocol studies, coders worked as a team to examine each design concept until a consensus list of heuristics was reached [14].

In one study of innovative products currently on the market, Yilmaz et al. presents a clearly detailed method for extracting heuristics. The process includes identifying key functions or features of an innovative product compared to other products of the same domain. Because insight into the cognitive processes could not be extracted, the focus was on key differences between designs [15]. However, the nature of our protocol study produces concepts that cannot be compared to finished products on the market. While the Yilmaz studies focus on creativity and ideation heuristics, we are focusing on a broader spectrum of heuristics. Many of the heuristics we hope to observe, such as process heuristics, cannot be obtained by only reviewing final concepts.

Humans use heuristics to approximate normative decision theories such as expected utility theory. This is often necessary because of an individual's limited cognition or the constraints of the problem, as described by bounded rationality [16]. Instead of trying for unbounded rationality, Gigerenzer and Selten recommend bounded rationality for individuals who are constrained by limited resources, which in fact applies to all decision makers [17]. As a result, individuals use heuristics for challenging tasks, such as estimating probabilities, that often introduce biases [18]. Thus, individuals make decisions that are inconsistent with expected utility theory [19, 20]. To describe this behavior, Kahneman and Tversky introduce prospect theory [21]. Prospect theory can be used to explain phenomena such as anchoring, a cognitive bias that describes how humans evaluate alternatives against an "anchor" that typically is formed from initial information [22]. Even so, prospect theory had its limitations in explaining decision making behavior, prompting cumulative prospect theory [23]. Cumulative prospect theory still does not provide us a reliable way to consider other important aspects of decision making, such as the emotional state of the decision maker. Though our ability to describe heuristic decision-making behavior has advanced, there is still room for improvement in understanding heuristics.

3 A CONCEPTUAL FRAMEWORK FOR HEURISTICS IN DESIGN

To explain the nature and importance of heuristics, we first need to provide a conceptual framework to think about design. We introduce a framework in which design is conceptualized very generically as an information-gathering search process. To make this search process more explicit, we reframe Equation (1) in terms of searching for a sequence of design actions:

$$\mathcal{P}:\ \max_{p\in P}\ \mathrm{E}\left[u\left(NPV\big(a(p), t(p), C(p)\big)\right)\right] \qquad (2)$$

where the optimization occurs over the set of all sequences of process steps (i.e., design actions), $p \in P$. The end result is still an artifact specification $a(p)$, but it is obtained implicitly as the consequence of following an SE&D process, p, rather than explicitly through optimization over $\mathcal{A}$.

Although this reformulation of the design problem is equivalent to Equation (1), it reflects more directly that the irrevocable allocation of resources to which a designer commits (i.e., the design decision) is the allocation of resources needed for the subsequent design actions (e.g., further analysis, artifact refinement, physical testing, design optimization at a certain level of abstraction, etc.). These process choices are truly the decisions made by designers, as opposed to artifact "decisions" that can always be reconsidered and reversed.

When one briefly explores what would be involved in solving Equation (2) rigorously, the equation implies that one should search across all possible processes, p, each consisting of a sequence of actions that lead to an artifact specification, $a(p)$. One should choose the process, p, that maximizes the expected utility reflecting the designer's preferences. However, each action in the process, p, results in new information and influences the best choice for subsequent actions. It is thus best to commit only to the first action, obtain the information it results in, and then consider subsequent actions. In addition, the information obtained from an action is not known in advance—it is uncertain. To determine even the best first action in a sequence is extremely challenging because it would require considering every possible outcome of that action and every possible outcome of each optimally chosen subsequent action—in essence, an infinitely deeply nested decision tree. Solving such a decision tree is computationally intractable, and reliance on approximations and heuristics is thus the only alternative. In summary, the question is therefore not: "Should we use heuristics in design or not?" but "Which heuristics should we use in design?"

Before continuing the discussion, it is important to be more precise about what we mean by "heuristic." At each point in a design process, the designer finds herself in a contextual situation. She has particular objectives, has certain information about the global socio-political-economic context, and has collected information during the preceding steps of the design process. Based on this contextual situation, a heuristic then constrains the actions the designer should consider for the subsequent action. Consider, for instance, the heuristic "When using a bolt connection, design it to have at least one and one-half turns in the threads" (adapted from [8]). The condition "When using a bolt connection" constrains the situations in which this heuristic should be considered. It defines a set of situations, which we call the applicability context of the heuristic. If a designer finds herself in a situation, in which this condition is satisfied, the heuristics directs her to choose an action that is consistent with the constraint "design it to have at least one and one-half turns in the threads." Mathematically, a heuristic, $h_i = \{s_i, p_i\}$, is thus a tuple consisting of a set of contextual situations, s_i, and a subset of design actions, p_i, as shown in Figure 1. We call s_i the applicability context of the heuristic h_i and p_i the set of possible actions of h_i. s_i is a subset of the power set of all possible applicability contexts, S, and p_i is a subset of the power set of all possible actions, P.

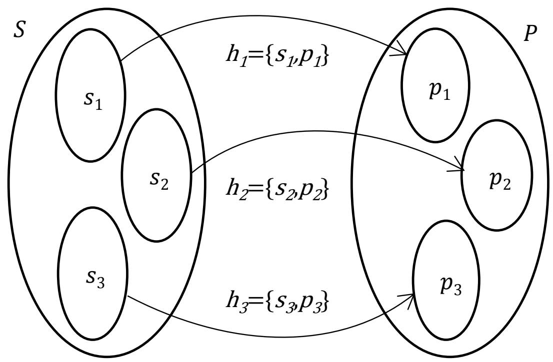

Figure 1. Each heuristic relates a set of contextual situations,s_i, to a corresponding set of design actions, p_i

Note that heuristics do not specify a single action, but a set. While they constrain the actions to be considered, the designer must still choose an action from the set of possible actions. Which particular action to perform is left as a choice to the designer. Heuristics are only suggestions that help the designer quickly home in on the most promising design actions to consider. Again, consider the heuristic "When using a bolt connection, design it to have at least one and one-half turns in the threads". In this case, the set of actions consists of actions directing the designer to specify certain characteristics of the bolt connection. The heuristic does not prescribe the designer to specify one particular bolt connection but provides a set of possible bolt connections from which the designer can choose (e.g., different bolt lengths, diameters, materials, etc.)

It may occur that multiple heuristics apply (i.e., that the current contextual situation satisfies the applicability condition for multiple heuristics). Often, these heuristics constrain different aspects of the design action to be taken, so that the actions to be considered are in the intersection of the action sets. "When designing a robot manipulator, start by specifying the kinematic structure" may be combined with "When selecting a kinematics structure for a mechanism, consider first how many degrees of freedom are needed," leading the designer to analyze the required number of degrees of freedom for the robot manipulator being designed. However, it is also conceivable that two heuristics have overlapping applicability contexts, but non-overlapping action sets. In such a situation a designer must apply good judgment and choose the action she believes to be most valuable.

This raises the issue of the quality of a heuristic. Is it meaningful to say that heuristic A is good, or heuristic B is bad? What determines the "goodness" of a heuristic? What we ultimately care about is the expected value (or more precisely, expected utility) of the outcome as expressed in Equations (1) and (2). The "goodness" of a heuristic must therefore be tied to this same criterion. It should reflect the designer's ability to achieve preferred, valuable outcomes through the application of the heuristic. To capture this more explicitly, we will use the term "value" rather than "goodness."

Even with this clarification, it is still not clear what the precise meaning is of the value of a heuristic. The outcomes, and thus the value, depend not only on one heuristic but also on any subsequent actions chosen by the designer. It is therefore not meaningful to refer to "value" as a property of an individual heuristic but only as a property of the set of all heuristics used by the designer. Following Koen [8], we call this set, the designer's state of the art, or "sota."

Finally, because preference cannot be measured in absolute terms [3], the value of a sota also is not an absolute measure. Rather than saying that "sota A is good," or "sota B is bad," one can only characterize A relative to B: "sota A is better than sota B."

Next, we consider how to determine which sota is better. One perspective argued in the literature is that design practices (i.e., a sota) should be consistent with normative decision theory [25-27]. Practices, such as the use of system requirements to define a systems engineering problem, have been critiqued as being irrational and inconsistent with the normative theory. However, we need to be careful not to jump to conclusions. In light of Equations (1) and (2), we need to recognize that the use of requirements impacts not only the artifact being designed, but also the communication and synchronization between teams of engineers inside a potentially very large organization or possibly even across multiple organizations. In addition, the communication and synchronization processes are performed by humans as cognitive, emotional and social agents. In other words, a sota includes heuristics regarding artifacts, processes and organizational design, and thus needs to be assessed according to its impact on the overall outcomes, not only on the artifact, but also on the design processes and the human organizations responsible for executing these processes.

Normative decision theory states that one should act in a way that is consistent with one's preferences and beliefs regarding these overall outcomes. To the best of our knowledge, an assessment of consistency based on such a broad perspective has not been performed. Even if inconsistencies were identified in a sota, it should not be dismissed right away. While pointing out the potential inconsistencies can aid in identifying opportunities for improvement, one should only abandon a sota once an improved sota has been identified.

In conclusion, a best-practice sota should use an approximation of the normative theory that is attuned to the economic and technological context and is well aligned with the characteristics of the human designer as cognitive, emotional and social agent. Comparing the relative value of two sotas from this perspective cannot be achieved through deductive reasoning based on an axiomatic, normative theory. It requires empirical testing and abductive reasoning.

As a first step towards gaining a better understanding of the influence of the impact of human psychology on the value of sotas, we have created an experiment to observe the use of heuristics by design engineers in academic robotics laboratories. We aim to observe how human designers sift through the sota to identify applicable heuristics and ultimately select a design action. From the observations, we hypothesize the following mechanisms:

- *Applicability.* Based on the applicability contexts of heuristics, quickly determine whether a heuristic is relevant.
- *Combination.* Further reduce the set of possible actions by taking the intersection of the action sets of multiple heuristics that address different aspects of the design action.
- *Value assessment.* Rely on experience and judgment to determine among the remaining actions which single action is adds the most value.

In addition, most design problems are too complicated to be addressed in a single heuristic. The designer's sota therefore includes a large number of divide-and-conquer heuristics that decompose high-level goals into sub-goals and suggest the order in which to pursue these sub-goals.

4 STUDY OF HEURISTICS IN ROBOTICS DESIGN

A study was designed to deepen understanding of how heuristics are used in the process of robotics design. The experimental design and method for data collection is described in this section (4). The method for data analysis is described in Section 5.

4.1 Participants

Participants were graduate students in robotics research labs at Georgia Institute of Technology, all having at least 2 years of experience in designing robots. The 5 participants in this preliminary study included 2 females and 3 males, with an average age of 24.4 years.

4.2 Study Design

In this study, participants were interviewed about their own approach to robotics design. Then, they were given a 45 minutes time frame to solve and to work on the following design problem:

Objective: *design wearable "third arm" robot to assist in everyday task requiring cooperative manipulation. For example, designing a robot that can grasp and hold objects (e.g. a flashlight) that the human wearer directly hands to it. You will be defining "every day task" yourself and define the following requirements accordingly:*

Performance requirements

- *Payload*
- *Speed*
- *Maneuverability (obstacle avoidance)*
- *Sensing for grasping objects and for determining human intent from motion cues (e.g. when to open/close grasper, where to move)*

Technical areas

- *Controls*
- *Mechanism design*
- *Optimization*
- *Computer vision*
- *Machine learning*
- *Signal processing*
- *Human factors*

Participants were asked to brainstorm, to sketch concepts, to write out thoughts, use resources such as the internet, calculate and/or analyze while designing. In addition, they were asked to speak their thoughts out loud as they worked on the design problem. Participants were video and audio recorded to capture their design process and behaviors. Participants were allowed to use any kind of resources and methods to help them approach the design task and record the design process. Audio and video data were used to extract heuristics that were used by each participant to complete the design task. The method for extracting heuristics is described next.

5 METHOD FOR DESIGN HEURISTIC EXTRACTION AND CLASSIFICATION

5.1 Extracting Heuristics

After data collection was complete, heuristics were extracted by two independent coders. The coders watched the recorded design process once before extraction for each participant in order to capture the context of the design, the design process and the concepts being generated. While coders were reviewing the video recordings, each coder wrote down identifiable actions and behaviors of the participants and the reasoning behind them, both verbal and written.

Among the actions taken by the participant, heuristics were identified based on the design context, participant's reasoning, and any other characteristics of a heuristic associated with the action. The intermediate set of possible heuristics was refined once more to generalize any heuristics that may be applicable to other robotics design contexts. Then, the coders discussed and came to an agreement on a final set of robotics design heuristics. The initial comparison between two coders reached a 77.8% match among all extracted heuristics.

5.2 Grounded Theory for a Taxonomy of Heuristics

Grounded theory is an inductive research technique, rooting in social science, in which researchers take an iterative approach to extracting categories within empirical qualitative data. Qualitative data, like the transcripts of the audio data collected here, is first reviewed and examined to extract common themes, and tagged with codes. These codes are grouped into emergent categories that then allow for classification and trends to be extracted from the data. The categories are then compared to existing literature for corroboration and refinement, and the data is analyzed again to adjust the first pass of coding and categorization [28, 29].

Using a grounded theory approach, the set of extracted heuristics are iteratively grouped into a proposed theoretical classification or taxonomy. With a broader goal of creating a taxonomy that could apply to heuristics beyond those extracted from the results of this study, the set of potential attributes that might be considered are:

<u>**General Applicability Attributes**</u>

- Phase of the design process
- Type of design process
    - Parametric
    - Variant
    - New product
- Field of engineering
    - Mechanical, etc.
- Type of system
    - Complexity
    - Level of uncertainty and risk
    - Level of software intensity
    - Level of cybersecurity
- Available resources
    - Human resources
    - Material resources
    - Time
- Action Intent
    - Planning
    - Analysis
    - Synthesis

This set of attributes is not meant to be exhaustive, but an illustrative starting point for future work.

Table 1. Classification Methods & Sub-categories

Design Phase			
Task Clarification	Conceptual Design	Embodiment Design	Detail Design

Action Intent		
Planning	Analysis	Synthesis

Field of Engineering Study				
Mechanical	Electrical	Computer Science	Chemical	Other

The taxonomy presented here, based upon the heuristics extracted from this study, is organized according to several attributes characterizing the applicability of the heuristics. In this study, three classification methods were used to identify and to categorize heuristics that each coder found, shown in Table 1, including design phase, field of engineering study, and action intent classification. Each classification method has its own sub-categories. After studying most standardized engineering design, Pahl and Beitz provides a formal definition for most of the categories [30]. After identifying these categories, the coders independently classified the heuristics according to each category and subcategory. Using Cohen's kappa inter-rater agreement, the coders were at an average agreement level of 0.71 as shown in Table 2.

<u>**Design Phase Classifications**</u>

Task Clarification: "to collect information about the requirements that have to be fulfilled by the product, and also about the existing constraints and their importance" [30].

Conceptual Design: determines the principle solution "by abstracting the essential problems, establishing function structures, searching for suitable working principles and then combining those principles into a working structure" [30].

Embodiment Design: "determine the construction structure (overall layout) of a technical system in line with technical and economic criteria" [30].

Detail Design: "the arrangement, forms, dimensions, and surface properties of all of the individual parts are finally laid down, the materials specified, production possibilities assessed, costs estimated, and all the drawings and other production documents produced" [30].

<u>**Action Intent Classifications**</u>

Planning: an action to define, to structure or to arrange the given problem, and to identify the essential from the non-essential elements of the system [30].

Analysis: an action to resolve or to decompose of anything complex into its elements and to further study the interrelationships between these elements [30].

Synthesis: an action to select a specific part or product or to combine parts or elements together for new effects, and to demonstrate that the combination of parts creates an ordered system [30].

<u>**Field of Engineering Classifications**</u>

Mechanical: field that relates to physical structure, geometry, material properties, kinematics, forces, or assembly

Electrical: field that uses or involves an electronic device or electricity

Computer Science: field that involves computation, programing, algorithms, or numerical analysis

Chemical: field that involves any kind of chemical reaction or extraction

Other: any other field that does not fit into one of above categories.

Table 2. Cohen's Kappa Values for Each Classification by participant

	Cohen's Kappa Value			
	Design Phase	Field of Eng. Study	Action Intent	
A	0.68	0.89	0.52	
B	0.72	0.81	0.71	
C	0.75	0.72	0.60	
D	0.82	0.63	0.75	
E	0.73	0.75	0.66	**Total Avg.**
Avg.	0.74	0.76	0.65	0.71

6 METHODOLOGICAL CHALLENGES FOR EXTRACTING HEURISTICS

Significant challenges were faced during the extraction of heuristics from the design study data. These challenges are presented to show what conflicts arise during the extraction process and how the methodology could improve with the resolution of these conflicts.

The limited duration of a controlled experiment excludes the observation of heuristics that result in actions that take longer than the duration of the experiment. In other words, the designer may leave some actions out because they know it will take more than the allotted 45 minutes to carry out. It may also force the designer to incorporate required actions that would not ordinarily reflect the participant's design process. These actions must be identified and labeled as not a true heuristic but an action influenced by given prompt. These instances may affect whether or not we are adequately identifying the participant's usual design process, so future studies will analyze the supplemental interview to compare the participant's idea of their own design process to what is actually done in the study.

The knowledge base of the participant may be broader in some areas than the coder's knowledge. If a participant has a background in computer science and refers to a specific software program unfamiliar to the coder, the coder will need to closely follow the verbal reasoning or perform research to get a general understanding of the product's function.

The observation of a participant's action does not provide sufficient information to extract the corresponding heuristic. To go from an instance of an action choice to a heuristic requires generalization, and the generalized applicability set and action set of the heuristic that was used by the participant cannot be inferred without subsequent corroboration. A participant may not be aware or have the ability to articulate the reasoning behind an action, so the coders must be aware of avoiding implied reasoning during heuristics extraction process and avoid bias and influence during corroboration. However, implicit understanding of the design process must play a role in the coding of the heuristics in applicability sets. The participants will rarely verbalize, for example, whether they are in the planning and task clarification phase or computer science domain, but this is knowledge that must be inferred by the viewer.

As the coders advanced from participant A to participant E in extracting and classifying heuristics, certain heuristics were considered adequate to combine and evolve into a generalized condition. Some heuristics were also considered too broad and broken into multiple heuristics. For example, "install motor for the robotic arm" and "put DC motor in the shoulder and elbow joints" were combined to say, "use motor on the robotic arm." Additionally, "use smartphone to connect to robot" was extracted from "put functions on a smartphone app to control robotic arm and read data." This portion of the methodology leaves enough room for subjectivity that biases can begin to form. The lack of formal specificity of granularity is a challenge to extracting heuristics that will only improve as a formal extraction method is created.

The method presented here for extraction and classification of heuristics is preliminary, and can be improved by addressing any or all the above challenges. The greatest and most philosophical challenge in extracting heuristics is one that many psychologists often face - how can we, as researchers, be sure of what is happening in someone else's mind? Therefore, at this time, we cannot comment on the conscious use of heuristics, though this is a key future direction for this work.

7 INITIAL OBSERVATIONS: HEURISTICS IN ROBOTICS DESIGN

A total of 110 heuristics were extracted from 5 participants. The heuristics are broken down in Appendix A into each classification followed by the action taken. For example, in the embodiment phase and mechanical engineering domain with an intent to synthesize, one action would be to "use backpack to mount battery and processing". At this point, the context of the heuristic is very broad. As our classifications become more specific, the context of the heuristic will also become more specific.

A total of 20 heuristics were used by multiple participants. One example of this is "Attach Bluetooth onto the robotic arm" (embodiment phase, electrical engineering domain, synthesis intent). Participant B used this for a major feature of the design by connecting the arm to a mobile phone app. The app could be used to store data as well as send commands to the robotic arm. Participant E used the Bluetooth feature as a means to charge the robotic arm with no disconnection required. The high number of unique heuristics found could be due to a lack of similar design processes in robotics, a high level of creativity among the participants, or diversity of backgrounds between each participant.

Table 3 presents the quantities and the percentages of heuristics in each of the classifications and sub-categories, organized by participant. Initial observations of the classifications reveal a significant shortage of planning and task clarification heuristics. Over half of the participants used two or fewer task clarification heuristics before moving onto conceptual design, and none of the participants progressed to the detail design phase. The authors hypothesize that this could be driven by a short design solving time window, which influenced participants to move into the conceptual design phase earlier than usual and to exclude the detail design phase from their processes. Another explanation could be that the design prompt defines the task so clearly and specifically that the participants felt comfortable enough to move onto the conceptual design phase without missing any crucial details that would otherwise have been hashed out in the task clarification phase.

Some participants tended to simply follow the suggested requirements and technical areas in the same order as they appeared on the prompt. This trend would not only potentially eliminate some typical planning heuristics, but also would lead to having less time available for the areas on the bottom of the list, such as machine learning and signal processing. Having less time for these areas could have resulted in fewer heuristics used in certain field of engineering study categories, such as computer science. In contrast, mechanical and electrical engineering categories accounted for the majority of heuristics, such as mounting the robot arm, battery, motor, and processing unit. Computer science accounted for far fewer heuristics on average. Another possibility is that the lack of computer science heuristics is related to factors like the phrasing of the prompt or the background of each participant, but a larger sample size would be required to further inspect these hypotheses.

Table 4 shows the cross-categorization of heuristics between action intent classification method and the other two classification methods. Most heuristics were mechanical and in the conceptual design phase. Planning action heuristics tend to be mostly mechanical engineering, while the majority of synthesis action heuristics tend to be in electrical engineering. Another interesting trend is that in the task clarification design phase, heuristics tend to be mostly planning actions, with a couple of analysis action and no synthesis action heuristics. Embodiment design phase heuristics tended to also be synthesis action heuristics.

Table 3. Quantities, Percentages and Classification of Heuristics by Participant

Design Phase	A		B		C		D		E		Avg. # of Heuristic	Avg %
Task Clarification	3	11%	7	30%	2	20%	2	10%	0	0%	2.8	14%
Conceptual	19	68%	11	48%	6	60%	12	60%	19	58%	13.4	59%
Embodiment	6	21%	5	22%	2	20%	6	30%	14	42%	6.6	27%
Detail	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
Total	28	100%	23	100%	10	100%	20	100%	33	100%		

Field of Eng. Study	A		B		C		D		E		Avg. # of Heuristic	Avg %
Mechanical	13	46%	10	43%	3	30%	7	35%	20	61%	10.6	43%
Electrical	9	32%	6	26%	1	10%	6	30%	9	27%	6.2	25%
Computer Science	4	14%	3	13%	5	50%	3	15%	2	6%	3.4	20%
Chemical	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
Other	2	7%	4	17%	1	10%	4	20%	2	6%	2.6	12%
Total	28	100%	23	100%	10	100%	20	100%	33	100%		

Action Intent	A		B		C		D		E		Avg. # of Heuristic	Avg %
Planning	9	32%	13	57%	3	30%	2	10%	4	12%	6.2	28%
Analysis	10	36%	2	9%	4	40%	9	45%	13	39%	7.6	34%
Synthesis	9	32%	8	35%	3	30%	9	45%	16	48%	9	38%
Total	28	100%	23	100%	10	100%	20	100%	33	100%		

Table 4. Cross-Categorized Heuristics by Participant

		Action Intent														
		Planning					Analysis					Synthesis				
		A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
Design Phase	Task Clarification	3	6	2	1	0	0	1	0	1	0	0	0	0	0	0
	Conceptual	6	7	1	1	4	9	1	4	7	13	4	3	1	4	2
	Embodiment	0	0	0	0	0	1	0	0	1	0	5	5	2	5	14
	Detail	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Field of Eng. Study	Mechanical	8	7	3	1	3	2	0	0	3	9	3	3	0	3	8
	Electrical	0	1	0	0	0	3	0	0	2	1	6	5	1	4	8
	Computer Science	0	2	0	0	1	4	1	3	2	1	0	0	2	1	0
	Chemistry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Other	1	3	0	1	0	1	1	1	2	2	0	0	0	1	0

For heuristics based on the participant's experience, heuristics such as "put functions on the smartphone app to control robotic arm and read data" led to a new way of thinking about heuristics for the coders. This heuristic not only shows the participant's experience with designing wireless connections, but also the participant's own social experiences with current technology. Because the participant has experienced the rise of smartphones and the internet of things over the past decade, this heuristic may be a product of living with today's technology. This reference to a time period in technological advancement links this finding directly to Koen's definition of sota, which he states must "be dated with a timestamp to indicate when it is safe for use" [8].

8 DISCUSSION AND FUTURE WORK

As our knowledge of heuristics and how heuristics play a part in the design process expands, the methodology should become more formal, with a distinct line of open ended questions for extracting heuristics.

There are other aspects of the data still to be analyzed. The preliminary interviews may be used to determine:

- How does the participant's view of their own design process compare to that which was extracted in the form of heuristics?
- How can the interview process be adjusted or expanded after the design task in order to validate the heuristics extracted by the coders?

Heuristics can be presented in way to guide a designer through a robotics design process if they are less experienced in some aspects of robotics design. If a designer has a mechanical background, a set of heuristics for the computer science design elements can be presented. The applicability attributes presented in the taxonomy section can be used to achieve this goal. More applicability attributes can be added to form a tree structure for decision making. The designer could follow the structure and use heuristics to guide her design process as she navigates from one branch to the next. The heuristics will also come back to the broader aim of the research and bridge the gap between normative and descriptive perspectives of design. After learning how a designer approaches the design process, we can work toward presenting a method of how they should do design.

Some initial hypotheses that the authors have formulated as to how designers use heuristics are presented next. These hypotheses are open for inquiry in future research.

1. Applicability contexts are simple so that they can be easily remembered and evaluated by designers in determining whether to use a heuristic. This is founded on the concept that a heuristic used by a designer must be chosen relatively quickly, as it is meant to be inexpensive guidance towards promising, valuable actions.
2. Designers rely on planning heuristics to identify sub-goals for which more specific heuristics are applicable. If reaching a goal requires a sequence of elementary design actions, then associating this sequence with one applicability region becomes difficult to memorize. In addition, it is likely that different contextual situations require variants of the long sequence so that it is easier to break the large problem down into a sequence of smaller, manageable problems.

3. Transferring heuristics from one context to another is based on abstraction and mapping, rooted in analogical thinking skills.
4. Designers assess heuristics' value based on its utility in a given context.
5. Designers update their value assessments of heuristics based on experience, improved understanding, learning, observation of other designers, and other forms of new knowledge that expand their understanding of how, when, and why a heuristic is useful.

Finally, it is important to distinguish between a value assessment by a designer, and a value assessment by a design researcher. A value assessment of a heuristic by a designer must be made quickly and inexpensively. A value assessment by a researcher, however, is not bound by such constraints. It may be very valuable for researcher to determine carefully and at great expense which action should be chosen in a particular set of situational contexts, so that this value assessment can be shared with practitioners to inform their rapid, in-the-field assessments.

ACKNOWLEDGEMENTS

The research presented in this paper was supported in part by the National Science Foundation under Award CMMI-1645316. The United States Government retains, and by accepting the article for publication, the publisher acknowledges that the United States Government retains, a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for United States Government purposes.

REFERENCES

[1] H. A. Simon, *The sciences of the artificial*. MIT press, 1996.

[2] R. A. Howard and A. E. Abbas, *Foundations of decision analysis*. Pearson, 2015.

[3] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton university press, 2007.

[4] B. D. Lee and C. J. Paredis, "A conceptual framework for value-driven design and systems engineering," *Procedia CIRP,* vol. 21, pp. 10-17, 2014.

[5] R. A. Moore, D. A. Romero, and C. J. Paredis, "Value-based global optimization," *Journal of Mechanical Design,* vol. 136, no. 4, p. 041003, 2014.

[6] R. A. Howard, "Information value theory," *IEEE Transactions on systems science and cybernetics,* vol. 2, no. 1, pp. 22-26, 1966.

[7] D. B. Lawrence, *The economic value of information*. Springer-Verlag New York, 1999.

[8] B. V. Koen, *Definition of the Engineering Method*. ERIC, 1985.

[9] G. Altshuller, L. Shulyak, and S. Rodman, *40 Principles: TRIZ keys to innovation*. Technical Innovation Center, Inc., 2002.

[10] K. K. Fu, M. C. Yang, and K. L. Wood, "Design Principles: The Foundation of Design," in *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2015, pp. V007T06A034-V007T06A034: American Society of Mechanical Engineers.

[11] S. Yilmaz, S. Daly, C. M. Seifert, and R. Gonzalez, "Cognitive heuristic use in engineering design ideation," in *2010 Annual Conference & Exposition*, 2010.

[12] S. Yilmaz and C. M. Seifert, "Creativity through design heuristics: A case study of expert product design," *Design Studies,* vol. 32, no. 4, pp. 384-415, 2011.

[13] S. R. Daly, S. Yilmaz, J. L. Christian, C. M. Seifert, and R. Gonzalez, "Design heuristics in engineering concept generation," *Journal of Engineering Education,* vol. 101, no. 4, p. 601, 2012.

[14] S. Yilmaz, S. R. Daly, C. M. Seifert, and R. Gonzalez, "Evidence-based design heuristics for idea generation," *Design Studies,* vol. 46, pp. 95-124, 2016.

[15] S. Yilmaz, C. Seifert, S. R. Daly, and R. Gonzalez, "Design Heuristics in innovative products," *Journal of Mechanical Design,* vol. 138, no. 7, p. 071102, 2016.

[16] H. A. Simon, "Bounded rationality," *The New Palgrave: utility and probability,* pp. 15-18, 1987.

[17] G. Gigerenzer and R. Selten, *Bounded rationality: The adaptive toolbox*. MIT press, 2002.

[18] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," in *Utility, probability, and human decision making*: Springer, 1975, pp. 141-162.

[19] M. Allais, "Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école Américaine," *Econometrica: Journal of the econometric society,* vol. 21, no. 4, pp. 503-546, 1953.

[20] K. R. MacCrimmon and S. Larsson, "Utility theory: Axioms versus 'paradoxes'," in *Expected utility hypotheses and the Allais paradox*: Springer, 1979, pp. 333-409.

[21] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica: Journal of the econometric society,* pp. 263-291, 1979.

[22] F. Strack and T. Mussweiler, "Explaining the enigmatic anchoring effect: Mechanisms of selective accessibility," *Journal of personality and social psychology,* vol. 73, no. 3, p. 437, 1997.

[23] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk and uncertainty,* vol. 5, no. 4, pp. 297-323, 1992.

[24] W. B. Powell and I. O. Ryzhov, *Optimal learning*. John Wiley & Sons, 2012.

[25] G. A. Hazelrigg, "A framework for decision-based engineering design," *Transactions-American Society of Mechanical Engineers Journal of Mechanical Design,* vol. 120, pp. 653-658, 1998.

[26] A. E. Abbas, "Normative perspectives on engineering systems design," in *Systems Conference (SysCon), 2013 IEEE International*, 2013, pp. 37-42: IEEE.

[27] P. Collopy and P. Hollingsworth, "Value-Driven Design," presented at the 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Aviation Technology, Integration, and Operations (ATIO), Hilton Head, South Carolina, 2009.

[28] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research*. Transaction publishers, 2009.

[29] A. Strauss and J. Corbin, "Grounded theory methodology," *Handbook of qualitative research,* vol. 17, pp. 273-85, 1994.

[30] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, *Engineering Design: A Systematic Approach*. London: Springer-Verlag London, 2007, p. 617.

APPENDIX A: PRELIMINARY EXTRACTED HEURISTICS

Design Phase	Field of Eng.	Intent	Action
C	CS	A	Consider embedded or transmitted computation
C	CS	A	Decompose machine learning testing process
C	CS	A	Give robot different grasping methods for different objects
C	CS	A	Have pre-programmed gestures
C	CS	A	Include feedback learning and reinforcement learning for the machine learning
C	CS	A	Include learning by demonstration in machine learning
C	CS	A	Let robot rank objects in order of usefulness
C	CS	A	List expected system inputs and outputs
C	CS	A	Put functions on the smartphone app to control robotic arm and read data
C	CS	A	Use forward and inverse kinematic processing
C	EE	A	Divide controls based on range of frequencies (low, high, and task)
C	EE	A	Investigate where EEG or EMG can be attached on arm
C	EE	A	List out all the possible sensors that can be used
C	EE	A	Use other device for microphone access rather than mounting onto the arm itself
C	ME	A	Consider arm design of metal and plastic, rigid parts, and motor actuator
C	ME	A	Consider snake-like actuation for robotic arm design
C	ME	A	Consider unique end effectors like coffee-filled balloon
C	ME	A	Have a total of 6 degrees of freedom for the robotic arm
C	ME	A	Have a total of 5 degrees of freedom for the robotic arm
C	ME	A	List pros/cons for design alternatives
C	ME	A	List requirements for material properties
C	ME	A	Make arm retractable
C	ME	A	Make the elbow joint belt-driven
C	ME	A	Research motor placement and connection to ball joint
C	ME	A	Sketch mechanical design
C	other	A	Consider robotic arm as a personal assistant format design
C	other	A	Consider robotic arm as an optimal prosthesis format design
C	other	A	Create a user study for robot predictability
C	other	A	Sketch drawings to explain and understand design schematics
C	other	A	Use logical architecture for perception, cognition, action
E	CS	A	Use vision data from camera to calculate distance and identify motion cues
E	ME	A	Design motor as large as human can comfortably carry
TC	other	A	Design hardware then software subsystems individually, then combine for overall architecture
TC	other	A	Mechanical design → sensor design → command generation design → vision system design → control system design
C	CS	P	Design robot to reciprocate human movements
C	EE	P	Place sensing human intent system on top of shoulder
C	ME	P	Attach arm to back / shoulder
C	ME	P	Attach robotic arm on the front of the human body
C	ME	P	Begin design process by sketching mechanical design
C	ME	P	Consider ergonomics
C	ME	P	Design anthropomorphic gripper
C	ME	P	Design end effector with an opposable thumb
C	ME	P	Design robotic arm based on off the shelf parts
C	ME	P	Design the robotic arm similar to the human arm structure
C	ME	P	Embed all electrical components into the robotic arm
C	ME	P	Give several joints & degree of freedom for the robotic arm
C	ME	P	Have an adjustable mounting system
C	ME	P	Set measure of success for robot grasping
C	other	P	Sketch early concepts
TC	CS	P	Design planning system and control system before performance requirements
TC	ME	P	Define payload range
TC	ME	P	Give several different degrees of freedom to the end effector
TC	ME	P	Limit the total weight of the robotic arm
TC	ME	P	Do not set the robot arm location in the design concept phase
TC	ME	P	Set robotic arm speed to mirror human body arm speed
TC	other	P	Focus on function over aesthetics
TC	other	P	Start with mechanical design before sensing system
C	CS	S	Keep computation attached to robot
C	CS	S	Use viewing system to estimate human pose and apply skeleton model to identify human vs. object
C	EE	S	Use battery and processing system to operate robot
C	EE	S	Use current sensing as a safety feature
C	EE	S	Use market feedback controllers for ease of designing joints
C	EE	S	Use vision based and touch based sensing features on the end effector
C	ME	S	Add gripping material to the end effector
C	ME	S	Use backpack weight for counterbalancing
C	ME	S	Use light/stiff links for speed/accuracy
C	ME	S	Use pneumatic/ hydraulic actuator on robotic arm
C	ME	S	Use traditional serial arm
E	CS	S	Compare 3D CAD models and PCL segmentation for object labeling
E	EE	S	Add force sensors for gripping or for safety reasons
E	EE	S	Add force sensors to other parts of robot for safety reasons
E	EE	S	Attach Bluetooth onto the robotic arm
E	EE	S	Put encoders to every joints
E	EE	S	Put servos for the wrist
E	EE	S	Put torque/pressure sensor on the end effector and arm joints
E	EE	S	Use camera for sensing system
E	EE	S	Use camera to get vision data
E	EE	S	Use cameras so robot sees what human sees
E	EE	S	Use microphone for voice recognition
E	EE	S	Use motor on the robotic arm
E	EE	S	Use RGB-D sensor as a viewing system
E	EE	S	Use same camera for obstacle avoidance and gaze following
E	EE	S	Use smartphone to connect to robot
E	ME	S	add a ball joint for the robotic arm
E	ME	S	Mount arm using backpack
E	ME	S	Place microphone on shoulder
E	ME	S	Use backpack to mount battery and processing
E	ME	S	Use harness/strap to mount robotic arm onto human body
E	ME	S	Use leather material for the shoulder girdle
E	other	S	Set the "stand still" motion as a safety behavior

A = analysis, P = planning, S = synthesis, TC = task clarification, C = conceptual design, E = embodiment design, D = detail design