

### **Research** Article

# A Hybrid Expert System for Estimation of the Manufacturability of a Notional Design

## Alexander Sommers <sup>(D)</sup>, <sup>1</sup> Shahram Rahimi <sup>(D)</sup>, <sup>1</sup> Tonya McCall, <sup>2</sup> Emily Wall, <sup>2</sup> Althea Henslee, <sup>3</sup> Larry Dalton, <sup>4</sup> Paul D. Babin, <sup>2</sup> Nathan Watson, <sup>2</sup> Gehendra Sharma, <sup>5</sup> and Milan D. Parmar<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, Mississippi State University, Starkville, MS, USA <sup>2</sup>Center for Advanced Vehicular Systems Extension, Mississippi State University, Starkville, MS, USA <sup>3</sup>Engineering Research and Development Center, Vicksburg, MS, USA <sup>4</sup>Institute for Systems Engineering Research, Mississippi State University, Vicksburg, MS, USA <sup>5</sup>Center for Advanced Vehicular Systems, Mississippi State University, Starkville, MS, USA

Correspondence should be addressed to Alexander Sommers; ams1988@msstate.edu

Received 8 August 2023; Revised 11 March 2024; Accepted 13 March 2024; Published 26 March 2024

Academic Editor: Sajad Fathi Hafshejani

Copyright © 2024 Alexander Sommers et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The more "manufacturable" a product is, the "easier" it is to manufacture. For two different product designs targeting the same role, one may be more manufacturable than the other. Evaluating manufacturability requires experts in the processes of manufacturing, "manufacturing process engineers" (MPEs). Human experts are expensive to train and employ, while a welldesigned expert system (ES) could be quicker, more reliable, and provide higher performance and superior accuracy. In this work, a group of MPEs ("Team A") externalized a portion of their expertise into a rule-based expert system in cooperation with a group of ES knowledge engineers and developers. We produced a large ES with 113 total rules and 94 variables. The ES comprises a crisp ES which constructs a Fuzzy ES, thus producing a two-stage ES. Team A then used the ES and a derivation of it (the "MAKE A") to conduct assessments of the manufacturability of several "notional" designs, providing a sanity check of the rule-base. A provisional assessment used a first draft of the rule-base, and MAKE A, and was of notional wing designs. The primary assessment, using an updated rule-base and MAKE A, was of notional rotor blade designs. We describe the process by which this ES was made and the assessments that were conducted and conclude with insights gained from constructing the ES. These insights can be summarized as follows: build a bridge between expert and user, move from general features to specific features, do not make the user do a lot of work, and only ask the user for objective observations. We add the product of our work to the growing library of tools and methodologies at the disposal of the U.S. Army Engineer Research and Development Center (ERDC). The primary findings of the present work are (1) an ES that satisfied the experts, according to their expressed performance expectations, and (2) the insights gained on how such a system might best be constructed.

#### 1. Introduction

We first present the user with an understanding of "manufacturability," a core concept to this work. We then introduce expert systems at a conceptual level, with implementation details made available later in Section 3. We then highlight this work's contributions and give a brief breakdown of the remainder of the paper. 1.1. An Introduction to Manufacturability. Manufacturability analysis is used to make an economic decision at the core of engineering: "Is this engineered solution (a design) to our problem worth building over other, competing, solutions?" The manufacturability of a design is defined as the ease with which a target manufacturer can use the resources at their disposal to manufacture a corresponding end product. This ease can be defined, and diagnosed, broadly or narrowly: narrowly, where considerations of production are spatially, temporally, and causally localized towards the core manufacturing facility, and broadly, when one endeavors to capture cause and effect further out down the supply chains involved.

For any given "problem-role" (ex: we need an aircraft that can carry this many passengers this many kilometers in this speed range, etc.), several designs may be proposed. As designs advance from concept to product, their numbers fall to cycles of selection. Only one or a few designs are ever manufactured. This actual manufacture might conclude the process, until a new generation of solutions is required. It might also be a further winnowing step, comparing the products in action. The highest volume of evaluations, simultaneously conducted with the least data and the most speculation, is at the concept ("notional") stage (DOD product life-cycle Milestone A [1]). Assuming any proposed notional designs would succeed as a solution to the target problem-role, there remains the matter of comparatively ranking those hypothetical solutions in terms of cost-tomanufacture, with the aim of selecting the least expensive, or most efficient, use of resources. Examples of these comparative analyses can be seen in Section 3.3.3.

Cost, here, is rendered as a "manufacturability score" (MS), rather than monetarily. Attempts to compute an MS require a direct interrogation of factors often obscured by money. By the fuzzy noise of the market, a product at a store has a specific price. This price is opaque as to the abundance, or dearth, of all that which occupies the supply chains which telescope behind the finished product. This includes such factors as labor, skill, material, machinery, transportation, danger, political complications, and more. The categories into which these cost concerns are sorted can be seen in Table 1.

1.2. Building a Rule-Based Expert System. This project constructed a rule-based expert system to allow MPEs to better solve this economic design problem. Rule-based expert systems can broadly be understood as sets of If-then rules which infer new knowledge from that already possessed [2]. A rule might be "if it is raining, then the ground is wet." Supplying a system that possesses this rule with the fact "it is raining" means it will then, on its own, infer that the ground is wet. This example, though perhaps unimpressive, demonstrates a mechanism by which sophisticated functions can be computed by the opportunistic firing of many rules sharing a database. The rules are like many workers gathering to shape a block of marble into a sculpture no one of them could produce. To give a domain example: if the primary material for a design is unavailable in the nation that wishes to manufacture the design, then the manufacturability is lowered in that context. Thus, the rule is "if the primary material is not a national product, then manufacturability is lowered." Details on the implementation of the ES can be found in Section 3.2.

A manufacturability score is a nonobjective measure. Firstly, the process by which it is derived is nonobjective. When produced by human evaluators, even experts, there is immense subjectivity. The ES we developed, though stable in that any iteration of its rule-base is deterministic, is an externalization of ultimately subjective human reasoning. Secondly, the authors are aware of no sense in which there is a "ground truth" to the "manufacturability" of a design. It is best used as a means to rank competing, closely comparable designs (as an "ordinal value").

1.3. The Motivations and Contributions of This Work. The motivation for this work is primarily a problem of scale. As mentioned, MPEs are subject matter experts (SMEs) who specialize in designing and assessing the processes which manufacture things and the manufacturability of proposed products. We will use MPE when we wish to emphasize the role of the experts in this particular project, which concerns specifically manufacturability. We will use SME when we are speaking more generically of the parties involved in constructing an ES. We will also occasionally use "expert" when we wish to talk about a knowledgeable person outside of their relationship to expert system construction. We will also sometimes use ES to refer to the system as a whole and sometimes to the rule-base it operates on and which is its core distinguishing feature. Given a design, MPEs assess its manufacturability relative to its alternatives. At the notional stage, products have very little in the way of quantitative detail, and the number of notional designs is quite high. There is thus a mismatch, there are not enough experts to evaluate all the designs, and each evaluation requires considerable expert knowledge due to the sparse design details.

An assessment assistant ES, which could be operated by non-MPE users, while allowing them to perform at an expert level, is an appealing but ambitious goal. Indeed, ESs have been used as assistants in such diverse domains as mineral discovery [3] and medicine [4], though more recent decision support systems often rely on artificial neural networks [5]. Our project moves towards this goal by creating a first draft of such a design-agnostic rule-base. While no longer as common a solution in the age of big data, this path was selected because the MPEs had no dataset sufficient for machine learning (ML) to be employed. The resulting system is a hybrid, containing two distinct ESs, assembled into a pipeline that manages the transformation from user inputs to the output of the manufacturability score.

This work's contributions include the following:

- (i) The design and development of a general-purpose rule-base to perform manufacturability assessments across an arbitrary domain of notional products (Sections 3 and 4)
- (ii) The formalization of a simple, systematic, and replaceable interview procedure to be employed by knowledge engineers with subject matter experts (Section 3.1)
- (iii) The articulation of heuristics for the construction of a rule-base for the class of problems that manufacturability evaluations belong to, allowing hard won experience to benefit other knowledge engineers (Section 4.2)

Criterion	Description/"cost-theme"
Sustainability	Costs associated with environmental impact, personnel safety, and long-term sustainability
Material availability	Costs associated with how easy or difficult it is to acquire the components and raw materials the design calls for
Process difficulty and experience	Costs associated with the maturity and robustness of the salient manufacturing processes
Industrial-base experience and capability gaps	Costs associated with how available the industrial pipelines, infrastructure, and environments are that support the salient manufacturing processes and production pipeline
Labor and workforce	Costs associated with the challenge of acquiring personnel of the required quantity and quality
Facilities and equipment	Costs associated with locating and equipping manufacturing facilities

TABLE 1: Cost criteria.

This concludes the introduction; the reader should now be equipped to understand the rest of the paper, which is formed as follows. Section 2 presents the prior art. Section 3 presents our methodology and salient implementation details (including our interview process and a breakdown of the ES and its components). Section 4 presents our results and discusses them to offer context. Finally, Section 5 summarizes the paper and offers suggestions for future work. An overview of the work can be seen in Figure 1.

#### 2. Prior Art

ESs have been used to solve a wide range of manufacturing problems. These range from relatively small problems, such as selecting 3D-printing materials [6], up to comprehensive planning of every step in a product's creation [7]. Indeed, attempts at computer-aided-process-planning are nothing new [8, 9], and their recurrence in the literature indicates the challenge of automating so abstract a set of tasks. However, a recent survey indicates a relatively low number of publications concerning their use in the design and implementation of manufacturing processes, with the majority focusing on the design of the tools used in working materials instead [10]. The utility of these systems extends beyond the mechanical and into the conceptual. Because manufacturing pipelines are built as much out of communicated expertise as they are from machines and materials, ESs have been brought to bear on managing interoperability between domains of expertise to improve the functionality of manufacturing processes [11]. While manufacturability has always been an integral concern when creating a product, it has rarely been the direct subject of assessment and is instead treated as a constraint on an ongoing manufacturing process [9]. The current work provides a modern contribution to this domain and instead concerns the direct assessment of the manufacturability of a design.

The MAKE ("Manufacturability Assessment Knowledgebased Evaluation") C assessment tool, developed by McCall et al., is best understood as rigorously developed rubric. Here, a rubric is defined as a formalized means of assessing something. A rubric does at least three things. First it defines/names the factors to evaluate. Secondly, it provides a means of operationalizing and scoring the factors. Lastly it defines a procedure for aggregating those scores. The MAKE C exists to standardize how to conduct assessments of manufacturability. It exists as software and is intended for conducting assessments of designs at DOD milestone C (prototyping) [12].

Prior to the present project, work was done that defined a taxonomy of concerns, which are the antecedents to the cost-subdomains, the "criteria" which this work uses [13] (see Section 3 for more details). Software tools which incorporated expert knowledge were also explored [14]. In all of these, the refrain has been that the earlier the manufacturability assessment can be made, the better. This is because a low manufacturability design, identified early on, can be avoided before more R&D resources are committed to it.

The MAKE A was developed during the present project, in tandem with the ES. Both are designed to conduct assessments at milestone A (notional). The MAKE A exists as an Excel spreadsheet using the same prompts as the ES, and an approximation of its control and inference rules (see Section 3.2.2 for more details). The ES and its associated software were the primary products of this project. The salient components are its variable definitions, control and inference rules, variable weighting schemes, and the behavior of these when used to perform manufacturability assessments (see Section 3 for more details).

#### 3. Methodology and Implementation

In their prior work [15], the MPEs broke down "cost" into six subdomains, described for the reader in Table 1. A manufacturability score summarizes and quantifies cost through these subdomains. Each of the six "criteria" corresponds to two things: first to a "cost-theme." As seen in Table 1, the "Sustainability" criterion concerns "costs associated with environmental impact, personnel safety, and long-term sustainability." These costs (like environmental pollution or workplace hazards) share the theme of sustainability in that their unchecked presence endangers the sustainability of any manufacturing project. The second thing to which each criterion corresponds is a set of ES variables and a set of ES rules which make inferences using those variables. These elements were implemented into our



FIGURE 1: We began with an initial interview (dashed) to establish the rule-base, which was then iterated on.

system. The variables and rules related to a criterion express expert reasoning about that criterion of cost. The themes, and names, of all criteria are listed for the reader in Table 1.

Designs at the notional stage assessed well before any kind of "blueprint" are available. Assessment of a design's manufacturability can be challenging with so little detail. The estimation the experts make when evaluating manufacturability is speculative and highly qualitative as a result, even though it is founded on their robust experience. The rulebase is a set of if-then inferences that are designed to embody the reasoning of these experts. As such, it operates at these same levels of hypothesis but benefits from being the synthesis of several experts. It is also, unlike them, deterministic.

3.1. The Interview Process. Following the guidelines set in [16], and using their terminology, our method of variable identification/definition, and rule extraction, was the interview. Semistructured interviews came first (see Figure 2), in the discovery stage, with structured interviews conducted in the review and refinement stage. The structure of these later interviews was exactly that of the then extant rules and variables. These were structured in that extant things with established relationships were being reviewed. We conducted two to four interviews per criterion, with each interview conducted by the knowledge engineer (KE) and a subset of the MPEs who would supply knowledge for that criterion. Each interview lasted in the range of a half hour to two and a half hours in person. This was followed by several hours to several days of asynchronous work. The interviews can be divided into the rule-formation (discovery, both of rules and the variables they act upon) interviews and the rule-validation (review) interviews. Prior to the interviews, the SMEs were briefed on rule-based expert systems. They were also briefed on the variable types they could use to

explain and express their reasoning, seen in Table 2. Of the four variable types listed in Table 2, the "Fuzzy" variables are the most mechanistically important in our work. This is because all ES values are eventually converted into fuzzy variables. Conflict resolution for fuzzy variables is easily performed. Defuzzification is performed to produce each criterion score.

Figure 2 describes the interview process used in the discovery interviews. The MPEs were asked to identify the nameable factors they used to evaluate a design's manufacturability within the criterion in question. These factors became the variables (vars) of the ES, and the relationships between vars became the rules. After supplying each var, and the range of values it might assume, the SMEs were asked if the target user(s) could be expected to know how to identify and assign a value to it when using the system. If the SMEs believed the user would be capable of providing a value for the var, then that branch of the interview ended. If not, then the SMEs were prompted to produce as many variables as necessary which the target user would be more able to supply values for. In Figure 2, these variables are described as "closer to the user" (see 1 for more on this terminology) because the user would be more familiar with them. That a variable is closer to the user does not mean that the user would yet be able to use it. Variables even closer to the user might need to be defined. The values supplied to user-close variables are used to infer the values of expert-close variables. These expert-close variables are important because they are the terms in which experts conduct their reasoning, which is what the ES needs to capture. The SMEs were prompted to provide these rules for inferring expert variables from user variables as well. This interview cycle repeated (notice that Figure 2 demonstrates recursion) until each chain of logic terminated at the user-end, with input variables the SMEs had declared the user would be able to



FIGURE 2: Flowchart of the discovery-interview process. As can be seen, the flowchart recurs when a variable has been generated at the current level of the interview which itself will need to be inferred from variables still closer to the user. The interview starts from the experts' domain of knowledge and works back towards the user's domain of knowledge.

provide values for directly. The ES infers the manufacturability score from the values of these inputs.

A var is "user-facing" if the user is expected to provide a value for it when using the ES. This is opposed to those variables for which the values are inferred. These "userfacing" vars are the closest to the user. As an example, the criteria score for the sustainability criterion is a manufacturability score across all costs associated with sustainability. The user cannot be expected to provide this score, and the ES is constructed to do this for them. But the user might be expected to provide any number of inputs concerning sustainability. For instance: "What percentage of the equipment used at the core manufacturing facility is electrically powered?" If the experts expected the user to be able to answer that, then the user would need to be able to supply a value. The experts might reason that any equipment which is electrically powered can eventually be powered by nonpolluting energy sources (nuclear power, for example) but equipment that is powered by the combustion of fossil fuels cannot be made nonpolluting. For user-facing vars, a prompt is required which would solicit the value from the user. Identifying which variables were user-facing, and what their prompts should be, was also initiated in the interview process described in Figure 2. Each prompt identifies the variable, presents the user with options or ranges for its value, and asks the user to input the value as they perceive it in/for the design they are evaluating. A named entity in the prompt, such as a manufacturing process, might be one of many. In these cases, the prompt instructs the user to only consider the most costly/risky one (weakest link) or to consider all instances as a whole (in aggregate).

#### 3.2. Components of the Hybrid System

3.2.1. Overview. In Figure 3, we can see a representation of the ES components, which were established in the interviews. The system we constructed consists of more than just the ES, as will be described later in this section. The ES actually comprises two separate ESs, the first constructing the second at run-time (see Section 3.2.4 for further explanation). Table 3 contains a breakdown of the system's major files and their functions. Our breakdown in this fashion was intended to facilitate easy editing by persons not necessarily familiar with the programming language or libraries we used. The definition files and the files holding the weight arrays are all read by programs but are formatted in user-readable syntax. This easy-to-edit syntax encourages iteration on the ES/pipeline. While superficially similar to diagrams of neural networks or control systems, the reader should understand Figure 3 to be figurative, not literal.

After the ESs, there is a linear function computing a weighted sum. In Figure 3, the user is given a set of weight arrays (Ex: array A:  $a_1, \ldots, a_6$ , array Z:  $z_1, \ldots, z_6$ ). These are

Example	How mature is the primary manufacturing process?	Is the primary material/component available in this state, an adjacent one, t nation as a whole, or only internationally?	Is a "high-bay" required in the core manufacturing facility?	How many times does the mode of conveyance (ex: sea, land) of the prima material/component change (ex: at a port)?
Use	Capturing subjectivity precisely	Capturing categorical or ordinal information	Capturing yes/no-true/false information	Capturing a number in a range
Type	Fuzzy	Multiple choice	Binary	Simple numerical

TABLE 2: ES variable type.



FIGURE 3: Logical pipeline of our system. Each subtree  $(C_1-C_6)$  figuratively represents the variables and rules concerning a particular cost criterion. The user supplies the inputs to both the ES (via the input variables, here represented as red and dark blue circles) and the weights  $(w_1, \ldots, w_6)$  used to modify the aggregation of the criteria scores  $(x_1, \ldots, x_6)$ . From some user-defined set of weight arrays (ex: array  $A_{a_1}, \ldots, a_6$ , array  $Z z_1, \ldots, z_6$ ) the user can select those which are used to participate in the weighted sum (summation at the right of the pipeline) which produces the manufacturability score (M). Dark blue circles represent control variables, which decide which input variables are active. Red circles represent input variables supporting rule execution. Light blue circles represent intermediate values produced by rule execution. Blue squares are penultimate variables used to compute the criteria subscores. Mathematically, these operations are executed in one of three ways. Fuzzy logic is used to convert crisp input values into fuzzy intermediate values in a process called fuzzification. Fuzzy rules map between fuzzy values. Last are defuzzification operations, which are not rules. The concluded weighted average uses predefined weights defined by the experts. More information is available in Section 3.2.3.

used to weight the relative importance of each criteria score according to the user's judgment. For example, a user evaluating a particular design may deem that sustainability is of little importance in the evaluation of that design. In that case, they would select/input a weight array which weighted the sustainability criteria score low, perhaps at 0. The weighted values being summed  $(x_1, \ldots, x_6)$  are the manufacturability scores for each criterion, the "criteria scores." These can be seen in Figure 3, the arrows exiting each criteria pipeline contributing one element of the set to be weighted. For example, the coefficient weighting the sustainability criteria value will reduce that value to p% of itself. Here, p is the proportion of the manufacturability score which sustainability concerns should account for (according to the user). Let us suppose that sustainability was fully 50% of what manufacturability should measure. If its criteria score (a value in [0, 1]) were 1, then the weighted sustainability value would be 0.5.

This same logic applies to each criterion score in the sum. The weighting coefficients used in our experiments are the product of an SME led "Analytic Hierarchy Process" [17]. Their derivation was the product of a small set of MPEs providing pairwise judgments of the importance of the criteria. Each criterion is judged to be more, or less, important than each other criterion using a ten-point scale. After a normalization step, these comparisons are rendered into the weights. Each weight indicates the average degree of importance of a criterion, in the eyes of the experts, relative to the other criteria. A weight greater than 1/6 (there are six criteria so, if they were equally important, the weight on each would be 1/6) indicates a criterion is more important than its fellows. The opposite is indicated when a weight is less than 1/6. At this time only one array of these weights is available but more could be easily added, to reflect different user preferences.

Users are shown the prompts and supply input via a command-line interface. The opportunistic inference rules infer new values from those supplied, or derived, without an explicitly expressed order of rule execution. In contrast, the control rules were programmed with explicitly expressed control flow. These rules determine which of the 70 input vars even receives a value, prior to inference by the ES, and were put in place to manage the large number of input variables.

iable definition file       Define the variables, and the values they can assume         a definition file       Define the rule-base. The rule-base transforms the values of input variables         a definition file       Define the rule-base. The rule-base transforms the values of input variables         trol rule definition file       Define the meta-rules which allow or disallow the various input variables to be         trol rule definition file       Define the meta-rules which allow or disallow the various input variables to be         construction program       Define the meta-rules which allow or disallow the various input variables to be         definition file       Define the meta-rules which allow or disallow the various input variables to be         construction program       Constructs the ES, updates the ES execution program, and executes that         edited       program, all at run-time         ght array       Weights the criteria level outputs prior to final aggregation         Accepts user input, runs the ES given that input, and presents the output of the       Construct output	Edited by the user, read by the ES construction program Edited by the user, read by the ES construction program Edited by the user, used as meta-input to the ES execution program dited by the programmer, is run by the user to then build and run the E execution program Edited by the user, used as meta-input to the ES execution program constructed by the ES construction program, edited by the programmer whe
---	--

TABLE 3: ES system files and functions.

#### 3.2.2. Rule Breakdown

- (i) Control rules: These rules operate on user-facing variables and determine which input variables are instantiated. They are executed with explicitly specified control flow in the ES. Control logic which modulates the firing of the rules is not impossible in a pure ES, but we found the value and effort incommensurate. The control rules are represented by the green lines in Figure 3, and their syntax is shown in Figure 4.
- (ii) Inference rules: These rules (red lines in Figure 3, syntax shown in Figure 5) form the core of the ES and are further divided between the crisp and the fuzzy sub-ESs. The crisp rules are implemented using the Experta library [18]. They infer the values of intermediate vars from input vars and then populate the database of the fuzzy ES. The fuzzy ES has fuzzy rules and is implemented in the Simpful library [19].

Our system contains 35 control rules and 78 inference rules. The way we implement the construction of our fuzzy ES by actions of the crisp ES is seen in Figures 6 and 7. We can see "rule\_list" being appended to many times, followed by two references to an object, "FS." FS is a fuzzy system, and the rule list is its rule-base. To it is added a linguistic variable, the value of which is then set. Thus, as our crisp ES executes its rules, it takes the actions which construct the corresponding infrastructure in the fuzzy ES. To learn more about fuzzy logic, Experta, or Simpful, we recommend to the reader [18–20], respectively.

Both types of rules are read at run-time from text files. The files contain a simplified syntax suitable for easy editing by nonprogrammers. Examples of each syntax can be seen, annotated, in Figures 4, 5, and 8. Figures 4 and 5 deserve further explanation.

In Figure 5, we see the syntax for expressing one of the fuzzy inference rules which accepts input from the user and modifies the value of a criterion's manufacturability score on the basis of that input. The reader will remember that rules have an "If x then y" structure. The "x" portion is a relationship an antecedent variable holds with a value, such as "if weather = rain." Here the antecedent is written on the line in Figure 5 which begins with "x =." The antecedent "Risk\_from\_materials\_of\_features\_consequence" is a variable articulated by the MPEs during the discovery interviews (as described in Section 3.1). It expresses the degree of risk a manufacturing project incurs as a function of having one or both of the following difficulties. A "materials difficulty" could include the challenge of machining a particular material, such as one that is hard, brittle, or must be kept at certain precise temperatures. A "features difficulty" could include the challenge of creating a particular shape, such as an organic curve with many hollows and nonlinear details. If the presence of one or both of these is necessitated by the design, then there is a reduction in manufacturability. The criteria category of this variable is "process difficulty and experience," as described in I. It is abbreviated as "Process\_availability" in the file excerpt Figure 5 shows. If the

Labor\_and\_workforce | Special\_training | True | Training\_checklist

Criterion	Name of antecedent variable.	Name of consequent variable (s).
	Value to adu consequent under.	nit variable (s)

FIGURE 4: Example of control rule syntax. Each of the four components is labeled in red for the reader.

f-	—т	he "f" marks the rule as "fuzzy". The name of the antecedent variable.
Х	=	Risk_from_materials_or_features_consequence
Z	=	none   low   moderate   high-Values, in order, of the antecedent
У	=	Process availability- The name of the consequent variable

w = very\_high|high|low|very\_low

Values, in order, of the consequent.

FIGURE 5: This is an example of a fuzzy rule as articulated in the user-editable syntax we developed. The definition is labeled (red) for the reader. This is a fuzzy rule with one antecedent (x line) and one consequent (y line) variable. The antecedent conditions (z line) and consequent conditions (w line) have a one-to-one relationship. As an example, if "Risk\_from\_materials\_or\_features\_consequence" equals "low," then "Process\_availability" is set to "high."

value of the risk representing antecedent rises towards the fuzzy value "high," then the value of this criterion-subscore falls towards the fuzzy value "very low." The values the antecedent might assume are written on the line below the antecedent itself. On the line beginning with "z =" we see fuzzy values separated by vertical bars. Below that, the line beginning with "y =" lists the consequent variable ("Process\_availability"). Lastly, the line beginning with "w =" lists the fuzzy values the consequent variable might assume. The correspondence between values on the z and wlines is both in order and one-to-one. Thus, in our "if x then y" articulation, we can assert that "if  $x = z_i$  then  $y = w_i$ ." The "f" on the line above the antecedent declaration marks the rule as fuzzy. All rules have a fuzzy consequent, but a rule is marked as being completely fuzzy if its antecedent is fuzzy as well because it is handled differently by the file parsing program.

In Figure 4, we see a single line of strings broken up by vertical bars. There are four segments which articulate a control rule (these rules were described earlier, in Section 3.2.2). Those four rule segments are as follows: first, the criterion the rule concerns; second, the antecedent variable of the control rule; third, the value or values (if there were multiple values, they would be separated by commas) which the antecedent variable must assume to allow the consequent variable(s) to be granted a value; and fourth, the consequent variable(s) (again, comma separated if multiple) which are being allowed to have values, or not, based on the antecedent variable's value. In the example shown in Figure 4, we see that the criterion is "Labor and workforce," as seen in Table 1. The antecedent variable "Special\_training," as articulated by the MPEs, is a binary variable asserting whether or not the manufacturing project will require its workforce to receive unusual or otherwise uncommon training. The value

```
@Rule(Fact(name = "Tech_complexity_risk_lesser_consequence", value = MATCH.value, linguistic_var = MATCH.linguistic_var))
def rule_number_5(self, linguistic_var, value):
    print("rule_number_5 fired")
    rule_list.append("IF (Tech_complexity_risk_lesser_consequence IS high) THEN (Tech_complexity_risk_greater IS low)")
    rule_list.append("IF (Tech_complexity_risk_lesser_consequence IS none) THEN (Tech_complexity_risk_greater IS high)")
    rule_list.append("IF (Tech_complexity_risk_lesser_consequence IS none) THEN (Tech_complexity_risk_greater IS very_high)")
    FS.add_linguistic_variable("Tech_complexity_risk_lesser_consequence", value)
@Rule(Fact(name = "Bigest_IB_tech_risk_likelihood", value = MATCH.value, linguistic_var = MATCH.linguistic_var))
def rule_number_6(self, linguistic_var, value):
    print("rule_number_6 fired")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS none) THEN (Industrial_base_size_and_prox IS very_high)")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS noderate) THEN (Industrial_base_size_and_prox IS high)")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS high) THEN (Industrial_base_size_and_prox IS very_high)")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS noderate) THEN (Industrial_base_size_and_prox IS low)")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS high) THEN (Industrial_base_size_and_prox IS very_high)")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS high) THEN (Industrial_base_size_and_prox IS very_high)")
    rule_list.append("IF (Bigest_IB_tech_risk_likelihood IS high) THEN (Industrial_base_size_and_prox IS very_low)")
    FS.add_linguistic_variable("Bigest_IB_tech_risk_likelihood IS high) THEN (Industrial_base_size_and_prox IS very_low)")
    FS.set_variable("Bigest_IB_tech_risk_likelihood", value)
```

FIGURE 6: Example of CRISP-TO-FUZZY rule syntax (Experta).

FIGURE 7: Example of FUZZY-TO-FUZZY rule syntax (Experta).

"True" is listed alone in the third segment, showing that the variables in the fourth segment are only allowed to be granted values of their own if the antecedent variable has the value "True." In the fourth segment, we see the single consequent variable "Training\_checklist." The user is only required to supply values to this consequent variable if the binary is true. This consequent variable gathers user input on the type of training required. We feel Figure 8 is sufficiently understandable on its own.

The file parser expands the contents of the text files into the code implementations of the rules. This was enabled by the low complexity of each rule in this rule-base, with one antecedent and one consequent. The control and inference rules are articulated in separate files, each with its own syntax. The fuzzy and crisp inference rules share a file and syntax. Crisp and fuzzy rules are converted into code after being read by the ES construction program described in Table 3. Examples of the code can be seen in Figures 6 and 7.

#### 3.2.3. Variable Breakdown

- (i) Control Variables. These variables (red circles in Figure 3) are used in the antecedents and consequents of the control rules.
- (ii) Expert System Variables. These variables are used in the antecedents and consequents of the inference rules. These are further subdivided into the following:

- (1) Input vars (dark blue circles in Figure 3).
- (2) Intermediate vars (pale blue circles Figure 3).
- (3) Output vars (pale blue squares Figure 3).

Users provide values to input vars, some of which are control vars and some inference vars. An input var can be both a control and an inference variable. The values of intermediate vars are inferred from those of input vars and/ or other intermediate vars. Output vars have their value inferred from intermediate vars and are used in computations in the post-ES pipeline. Many input vars are crisp, and all output vars are fuzzy and are defuzzified before being displayed. The syntax used to define each variable is shown in Figure 8.

Variables are rendered in a highly descriptive syntax (see Figure 8). All variables are defined with this syntax, intended to facilitate easy editing by the users. There are four variable subtypes (multiple choice, binary, simple numerical, and fuzzy, see Table 2) that determine what kind of input the user is asked for. This simple syntax eases edits and preserves the easy interpretability of the rule-base, a long-standing benefit of ESs.

3.2.4. The Two-Stage Expert System. A vast majority of our rule-base is implemented in Experta, which has no native fuzzy logic support. By virtue of being implemented in Python, a general-purpose programming language, the consequent of each rule can perform tasks other than edit the ES fact base. Functions, variable declarations, and more can be placed here. By the nature of our particular rule-base,



FIGURE 8: Here we see an example of variable definition syntax for a multiple-choice variable. Each subsection of the definition is labeled (red) for the reader. This variable is named "LAP\_to\_CMF\_prox." It describes the proximity (prox) of the source of the "least available process" (LAP) to the "core manufacturing facility" (CMF). It is a multiple-choice variable. We see also a definition of the prompt the user is shown. The user is asked a question, shown the choices, and selects the one which best applies to the notional design they are assessing.

many of our rules had crisp variables in the antecedent segment of the rule and a fuzzy variable in the consequent segment (as seen in Figure 6). These fuzzy variables were themselves often antecedents in later rules where only fuzzy variables were concerned (as seen in Figure 7). We implemented the hybrid rules using a two-stage method. The crisp antecedents were stage one and were implemented in Experta. The fuzzy consequents were stage two and were implemented in Simpful. In any such hybrid rule, the consequent segment contains the rule definitions and variable declarations to be executed as part of Simpful's fuzzy ES. We thus had two ESs. The first one was crisp. It constructed the second one, which was fuzzy. The latter executed only after the former had concluded.

#### 3.3. Use of the ES and the MAKE A

*3.3.1. Use of the MAKE A.* As mentioned before, the MAKE A, produced by team A, is an approximation of the ES. To establish its accuracy, and thus demonstrate that the ES

could be judged against it, a test was needed. To establish independence in this test, a second team was brought on. The two teams were to both conduct the same manufacturability assessment with the MAKE A. The MAKE A would be deemed accurate and reliable to the degree that it could produce agreeing manufacturability assessments when used by the two teams as they performed the same manufacturability assessment task. This was again to establish that the MAKE A was a suitable baseline of accuracy and reliability for the ES to be compared against. Team A and Team B (Team B was composed of a single member) both possessed some experience in the aerospace engineering doseparately main. These two teams performed manufacturability evaluations using the MAKE A. The evaluation assessed the manufacturability of four pieces of a notional UAV wing (see Figure 9) across three materials: aluminum, fiberglass, and carbon composite. There were four components, and each component could be made of any of the three materials. Thus, there were twelve permutations of part and material. Each team committed



FIGURE 9: Anatomy of a wing [21]. Skin (not shown): the outer surface of the wing. Spar: structural element running parallel to the longest axis of the wing. Rib: structural element running parallel to the direction of airflow over the wing. Stringer: thin linear elements connecting the skin to the frame.

a manufacturability evaluation of each permutation using the MAKE A assessment system. There were thus twentyfour data points in total. Given that these components of the wing are independent of one another, any hypothetical wing might have any permutation of parts-materials (eighty-one permutations were scored). The manufacturability score of a wing permutation was computed as the sum of the manufacturability scores of its parts. Teams A and B produced different manufacturability scores for each of the twelve foundational assessments. The manufacturability score of the aluminum variant of a part was always the highest, the fiberglass variant the second highest, and the carbon composite score the lowest. This pattern held, across all parts, across both teams. Thus, the teams always agreed on which material made a part the most manufacturable. The members of the teams felt the manufacturability scores produced by their use of the MAKE A accurately reflected the manufacturability of the notional product assessed. In conclusion, the two teams of MPEs conducted assessments with the MAKE A which, in the expert judgment of both teams, were accurate assessments of manufacturability. They also agreed, across all parts, that aluminum makes for a higher manufacturability score than fiberglass, and fiberglass a higher score than carbon composite.

3.3.2. Updates to the ES. Based on the SMEs' experience with the prompts (shared between the ES and the MAKE A), the ES was updated and revised after the wing-component assessment. The ES shares with the MAKE series the property that it is substantially dependent on the expertise of the team/user(s) who employ it in their assessments. The MAKE C (the rubric) is wholly dependent on their reasoning, including their understanding of the terms used in the prompts, their domain knowledge informing them of which response is correct, and which feature in the notional design makes that response correct. The ES differentiates itself in an important way from the MAKE C. The ES is not a guide, but an embodiment of knowledge and reasoning outside of the experts (team A in particular). It was constructed in an attempt to capture a significant portion of their knowledge and reason in its rules and variables. It is thus much more of a function (taking input and giving output) than a rubric (guiding an evaluation). The ES and MAKE A are currently equally dependent on the user possessing the expert-level knowledge needed to understand, and respond accurately to, the prompts.

3.3.3. Use of the ES. In order to check the consistency of the ES with their own evaluative reasoning, team A performed a side by side comparison. Team A took the MAKE A, derived from the ES, as their baseline of correctness, and tested the correctness of the ES against it. They did this by evaluating notional rotor blade designs, giving the same input to each evaluation system. We describe the details below before describing the results of the comparison.

Team A used an updated version of the MAKE A (included revised prompts, and newly added control rules/ variables, all to match the updated ES) to conduct an assessment of the manufacturability of notional rotor-blades (see Figure 10). They conducted the same assessment in parallel using the ES. Both the ES and MAKE A were given the same inputs.

The rotor-blade evaluation assessed the manufacturability of three different "core morphologies" by way of two different additive methods, as explained to our KE team by one of the MPEs. Here a core morphology refers to the shape of the hollow interior channel (the "core") that runs the longest axis of the rotor blade. The three variants were as follows: first, a "uniform core," where the cross section of the core is uniform along the length of the blade; second, a "modified as is" core, where the cross section narrows towards the center of the blade and widens towards the extremities of the rotor-blade; and lastly, an "original" core, which has a nonuniform cross section that varies across the length of the blade in a not simply articulated way. The exact nature of this variance is immaterial to its use in this project, save that it presents the most varied of the three morphologies.

The two additive methodologies were the "layup" and the "fiber winding" methods. In the layup method, the rotating armature that the blade is formed around has sheets of the selected material layered around it, somewhat analogous to wrapping a gift. In the fiber winding method, a filament of the material is wound around the armature instead. In both cases, glue is used to make the structure cohere. The space made inside the blade when the armature is removed is the "core," even though it is hollow.

The manufacturability of the six permutations of core morphology and additive method was assessed by team A using these two assessment tools (the MAKE A and the ES), and twelve scores were produced: six by the MAKE A assessment and six by the ES assessment. The outputs of the two tools were ranked in terms of manufacturability. There was ordinal agreement across tools for the four highest ranks. Thus, the tools agreed on which four permutations were the most manufacturable relative to one another. The



FIGURE 10: Anatomy of a rotor-blade excerpted from [22, 23]. (a) A blade cross section showing the structural elements, including the hollow inner cavity. (b) We see the whole of the blade fixed to the point of rotation. The local chord length *c* and the blade angle  $\beta$  (twist angle) depend on the radius coordinate *r*.

two lowest-scoring permutations (layup method—original core, winding method—original core) were ranked 5th and 6th in one assessment and 6th and 5th in the other. By the scales of both assessments, the scores of these two lowestscoring variants are close to one another. There is thus ordinal agreement between the two methods of assessment (MAKE A assisted and ES assisted) on the rank of four out of the six assessed variants.

3.4. Computational Costs. In practice, the users are the operational bottleneck of this system, with costs to read and respond to the questions far exceeding execution time. Users must also evaluate queries in sequence, limiting the degree to which parallelization can take place. For any given firing of a particular rule, its cost is constant. There are thus two ways the execution of the ES on a design would be distinguished. If two designs are evaluated using the same rule-base, then the answers to the control rules would be the only differentiating factor, since these control the number and identity of input variables which in turn control which rules are executed. If rule bases vary across the evaluations, then larger rule bases, and more permissive control rules, can create larger execution loads on the inference engine. Different rule types do have a different cost establishing feature. Control rules have a constant cost in their type of conditional. Thus, Boolean control rules cost less than those which compare an entry to a list of values. Crisp-to-Fuzzy rules are the most expensive, as they require the construction of all fuzzy sets across the universe of discourse for their target variables. This then triggers the execution of all fuzzy rules for that set, even if only a few sets are involved. Fuzzy-to-Fuzzy rules instead only instantiate and execute the fuzzy sets already involved, as the values have already been fuzzified.

#### 4. Results and Discussion

Our results are encouraging, and the primary contributions can be seen in Table 4. We were able to construct an expert system with 35 control rules, 78 inference rules, and 94 variables in total, all to the satisfaction of the MPE team. The performance satisfied both the MPE/SMEs, experts in this domain, and also our funding agency's technical person of contact. Due to the qualitative nature of these results, the metric of performance was the ES having acceptable fidelity to the SME's own subjective evaluations during the use case trials. The SMEs declared the ES as a success becasue their evaluations with and without it produced equivalent results. The construction of a rule-base, especially one of this size, is often quite time-consuming, but initial drafts of our rulebase were completed in eight months (from June 2021 to January 2022). Despite a large number of rules, the execution time is rapid, a testament to both the quality of the libraries we used and the efficiency of logic the MPEs articulated. The MPEs' desire to make the ES conform to the behavior of the MAKE A has been facilitated by the ease of editing the rulebase and weight arrays. This rule-base is an excellent first step, the first of its kind (to our knowledge), and can be distributed across other teams seeking to advance the state of the art in this area.

The software, made to facilitate easy revision, also facilitates easy iteration, even without a complete understanding of the underlying code. Our production of a rule-base reminds us that an absence of a dataset is not a dead end, provided experts are on hand to replace it. Lastly, the insights into constructing an ES, and the methodology of the interviews, were not available to us before we started and are here made available to others who can benefit from them going forward.

To make this work more meaningful, we now discuss the tractability of producing a system such as what was attempted here and present the heuristics we advise people to use, based on our experience of our production process.

4.1. The Tractability of Reliability and Validity Testing of a Manufacturability Assessment System. Cost is objective; some dollars were spent and we can count them. But as any economist will attest to, the cost of a product is subject to

TABLE 4: Contributions of this work.

Contribution	Explanation
Easy platform for iteration	The simple syntax we developed, which requires little needed knowledge of the underlying code, allows others to iterate on our work without having to write their
	own programmatic infrastructure
Demonstrates the tenability of an ES in this domain	We started this project aware of our lack of data with which to pursue a more conventional machine learning approach. Our methodology suggests that experts are a suitable replacement for data in the manufacturability domain.
Methodological insights	The insights into constructing an ES, and the methodology of the interviews, were not available to us before we started and are here made available to others who can benefit from them going forward

many factors. A clear example is the specialization of an economy's industrial base. Petroleum plastic may have material properties that make it energetically and mechanically less expensive to deploy for a given use case than metal (higher required temperatures) or glass (that plus being heavy and fragile). But even if methods were invented today which eliminated these advantages, our industrial base already has matured pipelines for deploying plastics that make doing so hypothetically less expensive than refitting competing pipelines. Thus, the manufacturability of plastic is high because it is a mature, often problematically widespread, and affordable option. All this is to point out that, but for a different choice of industrial specialization in the past, the cost of a given product could be very different. This itself is to give an example of how subject to change cost can be, to show that even it is not so steady and objective as its familiar quantification might suggest.

Even compared to this, manufacturability is yet more subjective. It departs the measurable without becoming a purely subjective narrative. It cannot be objectively quantified, even after the fact of a thing's production, and the current requirement of experts as the creators of datasets, or the pilots of evaluation systems, presents a bottleneck to the generation of the kinds of datasets from which mathematics can draw insights. The authors view the validation of any manufacturability scoring method to be challenged by the liminality of the manufacturability metric between the objective and the subjective, and between the measurable and the figurative. These subjectivities confound reliability and bring into question what it means for a manufacturability assessment to be accurate.

4.2. Recommendations for Knowledge Engineers. Firstly, our two-stage ES architecture is appropriate for specific applications only. Its crisp-first-fuzzy-second sequence is specific to a highly hierarchical rule-base with simple rule structure and few structural classes of rules. If only crisp (or only fuzzy) computations are performed, then there is no need for a division like this. A less predictable or less separable firing order of a mixture of fuzzy and crisp computations would not find this hybrid appropriate either.

Secondly, our experience producing our rule-base suggests to us several recommendations listed below. We feel they apply most strongly to rule-base construction for an ES used for a particular archetype of task. This task archetype has two main features. Firstly, it should concern expert judgments which are challenging to objectively quantify. For example, the task of answering this question: "If a rotor-craft with engine type A and blade arrangement B were compared to one with engine type C and blade arrangement D, all else being equal, then which is more manufacturable?" Secondly, there should be a significant knowledge gap between the expert and the layperson. In cases where the "expert" is hardly different from the layperson, and where evaluations are more objective, we expect these observations to be less applicable. We wish to emphasize that though the following recommendations are given in good faith and as a result of deliberation, their validation would be its own project.

4.2.1. Rule "Distance". Let a rule be "distant" from a layperson and "close" to an expert to the degree that it articulates reasoning that is nonobvious to a layperson. An expert is so because they possess rare knowledge and/or a rare concentration of common knowledge. Some rules an expert makes will make sense to a layperson, and the layperson would have generated them too. Some will be understandable after the fact by the layperson, but the layperson would not have thought of the rule themselves. Some rules are not even understandable to the layperson. It is these last rules which are closest to the expert. We suggest that every team constructing an ES attempt to start here. Exploit the highly nonobvious relationships before moving on to anything closer to the layperson, with the below caveat.

The users of an ES are often going to be less expert than those used to create it. Indeed, this is often the point. The less training it takes to make someone suitable to pilot an ES, and the better that ES is at replicating the reasoning of higher experts, the more functional experts one can simulate. The rule-base of an ES must exploit these nonobvious relationships to embody expert reasoning. These nonobvious relationships will be articulated in very expert-close rules, and they must be connected to antecedents the less expert person can easily and accurately supply. Producing rules that bridge the gap between the layperson and the expert is itself a creative and nonobvious process, but if the system is to augment the capabilities of a layperson, it must do so. This gap need not be crossed in one step, moving directly from the layperson to the expert. Chains of inference can take smaller steps, but the gap must be crossed. The expert must

try to see how they can perform their reasoning by proxy, using what the user will understand and be able to accurately know. This is analogous to the task before space probe designers, who have to design an automated system that can conduct science and move around with sub/nonhuman abilities.

The "target user," then, must be modeled. They are the kind of person the system is designed to be used by. A rulebase designed for use by one level of expertise might well be suboptimal for use by another. An expert made to perform with the terms of a layperson would be forced to articulate approximately what they know exactly. A layperson forced to perform at the level of an expert would be unable to perform accurately, being unfamiliar with the terms the rulebase uses. In our own work, the target users were defined as teams of MPEs. This allowed the SMEs, who were themselves a team of MPEs, to estimate well the terminology that would be familiar to their peers.

4.2.2. The Generality of the Features Focused on by a Rule. A team should strive first to make the rules concerning the most robust variables and relationships, being general before they are specific. One must decide for one's use case what this means based on functionality. It may mean that one should incorporate rules pertaining to manufactured vehicles of all kinds in an ES that will be used to evaluate aircraft designs. This is related to reference class forecasting [24] and the reference class problem. For one, rules concerning specifics can easily multiply the number of rules and input variables (see Section 3.2). But beyond this, there is a danger in overspecifying. Let us understand "feature" here to be anything the user might report on to the ES. Some aircraft have fixed wings, while others have rotors. If the rules generated for assessing manufacturability only concern fixed wing aircraft, then there is a large subset of designs the ES cannot usefully be applied to. More rules would need to be written, and the ES would need to be designed to ask early on which aircraft archetype is being assessed. The basic lesson here is to know exactly the job the ES will be asked to do ahead of time. This might be much more work and so one might focus on features that determine manufacturability across both rotor and fixed winged craft. As an example, the maturity of engine production pipelines is relevant to both fixed wing and rotorcraft, and should thus be more focused on than any manufacturing element unique to either subgenre. These rules are more robust across use cases, and should thus be generated first.

4.2.3. The User-Side Cost of an ES. Rule complexity and rule amount are different. The complexity of a rule describes the simplicity of logic it articulates. "If x then y" is simple while "If x and (z or w) and (a xor b) then y" is less so. While complex rules are harder for the KE and SMEs to create, the user is often not required to understand them. The burden on the user is instead a function of the number of input variables, and the difficulty of supplying them. In the present

case, most rules in the ES had a unique anticedent variable. This means that the number of rules and variables is roughly equal. A large number of input variables is often less desirable as it requires the user to provide more input. For every degree of effort it takes to use a tool, the tool is less likely to be used. It may also be desirable to have your SMEs feel a pressure towards efficiency. This pressure can drive them to identify the most salient, potent, and robust signals/ variables/inputs from which can be derived the desired output(s). We should thus prefer to use as few input variables as we can while still achieving useful performance. We should strive to justify the addition of every new input as worth the cost of having to enter it, and worth the associated costs like querying it in a database or figuring it out by doing a literature review. The best ES never used does no more good than the worst ES.

4.2.4. Inputs Should Be Observations rather than Judgments. Ideally, a good expert system, operated by separate users with similar expectations, would produce similar outputs for similar inputs. In short, its performance should be robust across users. This robust performance requires both validity and reliability, and reliability is a prerequisite for validity because an inconsistent tool cannot be trusted. While the designers of an ES will need to model their target user, the actual users will have some variation and that variation can be a source of inconsistency, if given an opportunity. We can consider a spectrum between two extremes. On the one end we place judgments which, if made by any two individuals, we would be surprised to find any salient difference in. Examples:

- (i) Given a picture of a small crowd, how many people are there?
- (ii) What is the length of a given piece of timber in meters?
- (iii) What operating system is a computer using?

At the other extreme we find judgments where we would be surprised to find exact agreement between any two randomly selected individuals. Examples:

- (i) Who is the greatest science-fiction novelist of the 20th century?
- (ii) What is the experience of dying like?
- (iii) When are humans likely to colonize Mars?

We often casually call the first group objective, and the second group subjective (though a great deal of rigor can be brought to bear in attempts to make the latter judgments). An ES can broadly do two things. First, it can ask the user to supply basic facts, upon which the SMEs, via the ES, commit reasoning/make judgments. Second, it can direct the user to make judgments, which the ES might then use in its own reasoning. As much as can be, rule-base designers should prefer the former because any variation in the users is amplified as the judgments they are asked to make become more "subjective." As this subjectivity grows, the reliability of the ES is endangered, as different users may produce importantly different subjective evaluations of the same situation given the same prompt.

#### 5. Summary

In summary, we constructed a functional externalization of a portion of the expertise of our MPEs, progressing towards a stabilized, standardized, design-agnostic pipeline for evaluating the manufacturability of notional designs. The produced system uses a two-stage design where a crisp ES executes first and constructs a fuzzy ES in the process. The primary computations are carried out by this hybrid expert system which contains 113 rules and 94 variables in total. As a result of our efforts, we also identified potential heuristics by which future rule-bases may be made for analogous problems. These heuristics advise the careful and early modeling of the target user as a guiding constant during the design of the rule-base. They further advise the minimization of reasoning not captured in the rule-base itself.

#### 6. Future Work

As to our future work, the standardization step forms the groundwork for our automation step, where we will engineer the human out of the evaluation. With automation, we can achieve the goal of actually increasing the number of designs that can be evaluated and can move towards rapid, higher scale R&V tests and iteration, to ensure the quality of the now automated evaluations. Alongside automation, there are many directions for specialization of such an ES. While the current rule-base is designed to evaluate a general product, versions could be specialized towards consumer electronics, militarized vehicles, and infrastructure, to name a few highly valuable areas. Additionally, expert systems have continued to evolve, and applications of neural networks (neuro-expert-systems), and other forms of knowledge representation could be applied to produce better results.

#### **Data Availability**

The data are controlled by the U.S. Army Engineer Research and Development Center (ERDC) and will be made publicly available one year after the initial publication of this work in accordance with Federal regulations on R&D funds.

#### Disclosure

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Army ERDC or the U.S. DoD.

#### **Conflicts of Interest**

The authors declare that they have no conflicts of interest.

#### Acknowledgments

This work by Mississippi State University was financially supported by the U.S. Department of Defense (DoD) High-Performance Computing Modernization Program, through the U.S. Army Engineer Research and Development Center (ERDC) Contract #W912HZ21C0014.

#### References

- Office of the Under Secretary of Defense for Acquisition and Sustainment, "Dod instruction 5000.02 operation of the adaptive acquisition framework," https://pypi.org/project/ experta/.
- [2] M. Negnevitsky, "Rule based expert systems," Artificial intelligence: a guide to intelligent systems, Pearson Education, pp. 25–54, London, UK, 2005.
- [3] E. Hart, R. O. Duda, and M. T. Einaudi, "Prospector—a computer-based consultation system for mineral exploration," *Journal of the International Association for Mathematical Geology*, vol. 10, no. 5, pp. 589–610, 1978.
- [4] A. K. Meena, "Study and analysis of mycin expert system," *International Journal of Engineering and Computer Science*, vol. 4, 2016.
- [5] M. Ghaderzadeh, "Clinical decision support system for early detection of prostate cancer from benign hyperplasia of prostate," *Studies in Health Technology and Informatics*, vol. 192, p. 928, 2013.
- [6] W. Yahya and W. M. W. Muhamad, "Plastic additive manufacturing expert systems for small and medium automotives industries," *Tap chí Khoa học Đại học Văn Hiến*, vol. 9, no. 3, pp. 89–102, 2023.
- [7] X. F. Zha, H. Du, and J. H. Qiu, "Knowledge-based approach and system for assembly oriented design, part i: the approach," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 1, pp. 61–75, 2001.
- [8] D. K. Liu, "The hiclass software system: a manufacturing expert system shell," in *Proceedings of the The Fourth Conference on Artificial Intelligence Applications*, pp. 256–261, San Diego, CA, USA, March, 1988.
- [9] T. Shojaie, L. Zeidner, S. Sadri, and Y. Hazony, "The boston university manufacturing expert system (bumes)," ACM SIGAPL- APL Quote Quad, vol. 21, no. 4, pp. 278–288, 1991.
- [10] S. P. Leo Kumar, "Knowledge-based expert system in manufacturing planning: state-of-the-art review," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4766–4790, 2019.
- [11] B. S. Adamczyk, A. L. Szejka, and O. Canciglieri, "Knowledgebased expert system to support the semantic interoperability in smart manufacturing," *Computers in Industry*, vol. 115, Article ID 103161, 2020.
- [12] S. Fuller, T. McCall, E. Wall, and T. Falls, "Utilization of a manufacturability assessment methodology and metric: a case study application," in *Proceedings of the 10th Model Based Enterprise Summit (MBE 2019)*, Gaithersburg, MD, USA, March, 2019.
- [13] C. Walden, T. McCall, and R. Gedik, "Taxonomy based assessment methodology: improving the manufacturability of a product design," in *Proceedings of the from the 2016 Industrial and Systems Engineering Research Conference*, Anaheim, CA, USA, May, 2016.
- [14] E. S. Wall, S. C. Fuller, and T. C. Falls, "Design of the manufacturability assessment knowledge-based evaulation tool," in *Proceedings of the International Annual Conference of*

the American Society for Engineering Management, pp. 1–10, American Society for Engineering Management (ASEM), Denver, CO, USA, January, 2018.

- [15] T. G. McCall, C. H. Rinaudo, and D. B. P. D. P. E. Paul, "Development of an early lifecycle conceptual configuration manufacturability assessment metric for tradespace analytics," in *Proceedings of the International Annual Conference of the American Society for Engineering Management*, pp. 1–11, American Society for Engineering Management (ASEM), Denver, CO, USA, November, 2020.
- [16] R. R. Hoffman, "The problem of extracting the knowledge of experts from the perspective of experimental psychology," *AI Magazine*, vol. 8, no. 2, p. 53, 1987.
- [17] A. Cahyapratama and R. Sarno, "Application of analytic hierarchy process (ahp) and simple additive weighting (saw) methods in singer selection process," in *Proceedings of the* 2018 International Conference on Information and Communications Technology (ICOIACT), pp. 234–239, IEEE, Yogyakarta, Indonesia, June, 2018.
- [18] R. Pérez, "Experta: Expert Systems for python," 2022, https:// pypi.org/project/experta/.
- [19] S. Spolaor, C. Fuchs, P. Cazzaniga, U. Kaymak, D. Besozzi, and M. S. Nobile, "Simpful: a user-friendly python library for fuzzy logic," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 1687–1698, 2020.
- [20] M. Negnevitsky, Fuzzy Expert Systems. Artificial Intelligent a Guide to Intelligent Systems, Pearson Education, London, UK, 2nd edition, 2005.
- [21] "Mesh of the wing," https://community.sw.siemens.com/s/ question/0D54O00006VOssISAT/mesh-of-the-wing.
- [22] D. Salimi-Majd, V. Azimzadeh, and B. Mohammadi, "Loading analysis of composite wind turbine blade for fatigue life prediction of adhesively bonded root joint," *Applied Composite Materials*, vol. 22, no. 3, pp. 269–287, 2015.
- [23] L. O. Nicolas, "Simulação em dinâmica dos fluidos computacional de hélices," Master Thesis, Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora, Brazil, 2019.
- [24] D. Kahneman and A. Tversky, "Intuitive prediction: biases and corrective procedures," Decisions and Designs Inc Mclean Va, Technical report, Decision Research, A Branch of Perceptronics, 1201 Oak Street, Eugene, Oregon 97401, 1977.