PROMETHEUS: PROcedural METHodology for developing HEuristics of USability

Cristhy Jiménez, Héctor Allende Cid, Ismael Figueroa

Abstract—Usability is a key discipline related to the development of modern software systems. Its goal is to assess the user-friendliness and effectiveness of a software product from the user point of view. Therefore, proper methodologies and techniques to perform this assessment are definitely relevant. Usability evaluation is probably the most commonly used method for usability assessment. Initially developed by Nielsen and Molich in the ’90s, traditional heuristic evaluations rely on Nielsen’s well-known 10 usability heuristics. However, recent evidence suggests that such heuristics are not sufficiently complete for dealing with new domains such as interactive television, virtual worlds, and many others. In addition to the lack of suitability of traditional heuristics, in the past years the lack of a robust methodology or process to effectively develop and validate these new domain-specific heuristics has been documented. In this paper, we summarize current evidence regarding the lack of suitability of traditional heuristics, as well as the need to develop new domain-specific heuristics. After identifying and acknowledging existing gaps in heuristics for state-of-the-art technology, as documented by other researchers, we present PROMETHEUS, a PROcedural METHodology for developing HEuristics of Usability. PROMETHEUS refines the methodology of Rusu et al. (2011), and is composed of 8 stages. PROMETHEUS clearly defines the artifacts that are required and produced by each stage, and also presents a set of quality indicators in order to assess the need for further refinement in the development of new heuristics. As an initial validation of PROMETHEUS, we apply a questionnaire to several researchers that have used the methodology of Rusu et al., and we have also performed a small retrospective study, computing the quality indicators of several previous studies. Our results suggest that PROMETHEUS is a very promising methodology, and that the metrics and indicators are indeed pertinent with respect to the conclusions of previous studies.

Index Terms—usability, heuristic evaluation, usability heuristics, procedural methodology, human-computer interaction, empirical studies.

I. INTRODUCTION

Onstant technological development is reflected in the development of new applications, many of them based on new information technologies or emerging technologies, such as mobile devices, touch interfaces, distributed applications, virtual learning environments, to name just a few. These applications characteristically belong to a specific domain, and are accessed using different physical and logical devices, as well as specific contexts of use. Since it is increasingly common that non-specialist users need to access and use such applications on a frequent basis, it is necessary and important to facilitate such users’ experience to improve their satisfaction levels and task performance.

Usability is a discipline which, in the field of Software Engineering, allows us to estimate the degree to which a software product can be used by specific users to achieve specific goals effectively, efficiently and satisfactorily in a specific context of use [2]. Therefore, one of its main activities is determining usability problems in specific applications, in order to improve the quality of this attribute in future iterations. Given the high costs of usability tests [3], where the behavior of real users is observed and evaluated, two lower-cost techniques are generally used to identify usability problems: the inquiry and the usability inspection. Usability inquiries characteristically use a qualitative approach based on user opinions, and include methods such as questionnaires, field studies, and different kinds of interview. On the other hand, usability inspections consist of detailed examinations carried out by groups of evaluators, who report on specific and general problems regarding a particular application. One of the most widely used inspection methods is heuristic evaluations, proposed by Nielsen and Molich [4]. In this method, a small group of 3-5 evaluators inspect a particular software system’s usability, guided by a list of usability criteria known as usability heuristics [4]. Heuristic evaluation is a low-cost easily applicable method, which can be used at various stages of an application’s development [4]. It is also an effective method, as it can detect around 75% of usability problems with groups of 3 to 5 evaluators [6]. However, while heuristic evaluations are traditionally carried out using Nielsen’s 10 usability heuristics as criteria [7], the need to develop domain-specific heuristics, or put simply, domain heuristics, is ever more pressing.

In a recent study, Hermawati and Lawson [8] performed a bibliographic review of 70 articles related to the development of domain heuristics, identifying two large problems: significant deficiencies in efforts to validate new heuristics, and lack of rigor, robustness and standardization in the effectiveness analysis of domain heuristics. As a solution to these problems, in this article we present PROMETHEUS, a PROcedural METHodology for developing HEuristics of USability, which refines the existing methodology of Rusu et al. [9]—which we call R3C, based on its authors’ initials—for greater formality, rigor and precision in the process of constructing new domain heuristics. PROMETHEUS has 8 stages, in each of which the activities to be performed, the expected and produced artifacts, and the quality indicators for heuristics — which in turn guide the 8 stages’ continuous refinement
process — are described in detail. In this article, we begin by presenting an overview of how domain heuristics have been usually generated (Section [1]) and the R3C methodology (Section [III]), to then present the main contribution, which is PROMETHEUS (Section [IV]). Finally, we conclude with the initial validation of PROMETHEUS (Section [V]).

II. HOW DOMAIN HEURISTICS ARE DEVELOPED

There is vast literature on heuristic evaluations and the development of usability heuristics. In fact, as is studied by Hermawati and Lawson [8], there are at least 70 relevant studies regarding the development of domain heuristics, so a detailed review is beyond the scope of this article. Below we present a summary of the main conclusions of [8], which serve as the foundation for the artifacts and stages proposed in PROMETHEUS.

A. Collecting and transforming information

The first point to emphasize is that most of the studies analyzed in [8] include two large stages: collecting information on heuristics, and transforming that information into domain heuristics. Hermawati and Lawson describe 4 strategies for collecting information:

1) Adopt existing theories.
2) Analyze context of use.
3) Analyze existing case study reports.
4) Analyze and/or create a common set of usability problems.

Then, the transformation process uses one of the following 3 strategies:

1) Make a list of all the information, eliminating redundancies and overlaps, and using the end result as the new domain heuristics.
2) Perform the same normalization process, and sort the resulting information into categories, which are then transformed into heuristics.
3) Extend and/or modify Nielsen’s heuristics.

B. Validating the heuristics

An important result from [8] is that they detected deficient validation of domain heuristics. In short, 34% of the analyzed studies reported no validation. Of the remaining 66%, the most commonly used methods to validate domain heuristics are:

1) Experts applying the domain heuristics (24 studies).
2) Comparing the results with those from other heuristics (20 studies).
3) Comparing the results, based on tests with real users (5 studies).

However, there are few studies that attempt to determine the effectiveness of the new heuristics. As reported in [8], only 19 of the 70 cases studied present a comparison in terms of effectiveness i.e. the other cases do not present validation, do not use a control group of heuristics, nor do they perform any comparison or quantitative analysis. This notwithstanding, in the cases where the new domain heuristics versus a set of control heuristics are studied, the following categories of validation techniques were identified:

1) Quantify all usability problems, per heuristic, and compare with an evaluation using control heuristics.
2) Quantify and compare the frequency, severity and distribution of those problems via each specific heuristic and via each set of heuristics.
3) Identify and quantify problems detected in both sets of heuristics, and unique problems detected separately by the heuristics sets.

However, there is great variability in the extent and rigor of the quantitative analyses. On the other hand, only 3 studies used the predefined metrics proposed by Hartson et al. [10]. To conclude this summary, the authors of [8] identified 3 areas to improve the rigor and robustness of validating domain heuristics:

1) Adopt robust and rigorous validation metrics, to determine the effectiveness of new domain heuristics.
2) Create new domain heuristics based on existant heuristics in the domain.
3) Improve the definition and categorization of expert, so as to control the variability introduced by evaluators.

III. R3C METHODOLOGY

In this section, and with the objective of contextualizing the contribution of PROMETHEUS, we describe the R3C methodology. This is a methodology consisting of 6 stages, which we quote textually [9]:

1) Exploratory: to collect bibliography related with the main topics of the research: specific applications, their characteristics, general and/or related (if there are some) usability heuristics.
2) Descriptive: to highlight the most important characteristics of the previously collected information, in order to formalize the main concepts associated with the research.
3) Correlational: to identify the characteristics that the usability heuristics for specific applications should have, based on traditional heuristics and case studies analysis.
4) Explicative: to formally specify the set of the proposed heuristics, using a standard template.
5) Validation (experimental): to check new heuristics against traditional heuristics by experiments, through heuristics evaluations performed on selected case studies, complemented by user tests.
6) Refinement: based on the feedback from the validation stage.

Evaluation criterion. R3C identifies and defines three types of problem that can be detected in the experimental stage, consisting of groups of evaluators who use the new heuristics, and groups of evaluators using the control heuristics. Note that we use a new notation to refer to these types of problem.

- Common problems, $P^\ast$: Problems identified by both groups of evaluators.
- Domain problems, $P_D$: Problems identified only by the group of evaluators who used the new heuristics.
Control problems, $P_C$: Problems identified only by the group of evaluators who used Nielsen’s heuristics (or others if appropriate) as control.

With this sorting, R3C establishes that the new heuristics work well when $P^*$ and/or $P_D$ include the highest percentage of problems. In practice, the severity of specific and general problems is also considered. Finally, note that R3C is a simple methodology, which has been applied successfully and offers a framework that encourages experimental validation, thus avoiding some of the problems described in Section II. It is precisely because of this that we take R3C as starting point for the development of PROMETHEUS.

IV. PROMETHEUS

PROMETHEUS is a PROcedural METhodology for developing HEuristics of USability, which has emerged as a refinement of R3C. We say it is a procedural methodology as it accurately describes the steps to follow—and the artifacts to construct—during the elaboration process of new domain heuristics. PROMETHEUS originates from a previous investigation by Jiménez et al. [11] to determine the usability of R3C. Therefore, they conducted a questionnaire with the researchers who developed heuristics using R3C, in the domains of grid computing [12], interactive television [13], virtual worlds [14] and touchscreen devices [15]. On the quantitative side, the aggregated results are inconclusive, because the ease of use was evaluated as “neutral”. However, there were specific difficulties in the explanatory, experimental and refinement stages. This was confirmed by the comments made by those surveyed, which indicate lack of clarity in these stages, and especially in refinement.

Next, we present PROMETHEUS, starting with its contextualization regarding R3C, and then describing the critical path for the methodology. A full-fledged description of PROMETHEUS is presented in Section IV.

A. Refinements to R3C

Figure 1 presents a diagram with the R3C stages and the PROMETHEUS stages. At a first glance, the main difference is that PROMETHEUS breaks up the exploratory and correlational stages, each into two new stages. In addition, the descriptive and explanatory stages have been renamed, and the validation and refinement stages have been refined. The specific contributions of PROMETHEUS to the R3C stages are:

- **Exploratory**: PROMETHEUS specifies the stages search for specific information and search for heuristics of usability. The first considers 4 characteristic dimensions to describe the specific domain: context of use, logical devices, physical devices, and user profiles. The second stage provides guides for a systematic literature review that produces a set of heuristics applicable to the domain.

- **Descriptive**: PROMETHEUS specifies an encoding table of specificity indices, based on the domain’s characteristic dimensions.

- **Correlational**: PROMETHEUS proposes two stages: normalization of heuristics and prioritization of heuristics. In the first stage, cases of duplication or overlap found in the heuristics are resolved. The second case creates a specificity ranking for each heuristic, also considering the characteristic dimensions, among other factors.

- **Explanatory**: the explanatory stage is kept from R3C, although recent evidence suggests it is possible to improve this stage [16], e.g. considering novice evaluators.

- **Validation**: we have kept the idea from R3C to perform experimental validation. However, PROMETHEUS requires at least one heuristic evaluation using the domain heuristics and a group of control heuristics. In addition, PROMETHEUS specifies quality indicators for the heuristics, based on this experiment.

- **Refinement**: in PROMETHEUS, the quality indicators obtained during the validation stage are used to suggest in which stages changes must be made or which specific problems need to be solved.

B. Overview of the critical path

Figure 2 illustrates the 8 stages of PROMETHEUS to develop domain heuristics. The figure shows the required artifacts as input, and the produced artifacts as output for each stage. It also shows potential intermediate outputs, when suitable heuristics are found, thus making the development of new heuristics unnecessary. However, the methodology’s critical path consists of all the steps taken by a researcher for the effective development of new usability heuristics for a specific domain. That is to say, despite considering situations of intermediate outputs, PROMETHEUS is focused on situations where it is necessary to create new domain heuristics.

For starters, in the first iterative application of the methodology, all stages should be applied sequentially. Then, based on feedback from the refinement stage, greater adaptability and flexibility are possible in the stages performed during the progressive improvement of the new heuristics. Obviously, the number of iterations and the stages that have to be repeated will depend on the researcher’s needs, the results of the validation, and other factors specific to each project. However, the indicators generated in the validation stage allow us to make an informed decision whether or not to continue with the refinement. The critical path is summed...
up in four major phases:

1) Systematic search: The process starts by selecting the domain for which a set of specific usability heuristics is required. Stages 1 to 5 consist in a systematic search process, specificity encoding, and prioritizing one or more sets of existing heuristics, based on the characteristic domain dimensions, using a standard template.

2) Defining the domain heuristics: At the end of stage 5, the researcher will have a set of heuristics applicable to the domain, where there should be no duplication or overlapping problems. Crucially, at this stage the researcher can create new heuristics based on existing data. Then, stage 6 describes the heuristics using a standard template, like in R3C. The detailed description aims to facilitate the evaluators’ task of performing the experimental validation. The result of stage 5 and/or 6 is the new domain heuristics.

3) Experimental validation: Subsequently, stage 7 is to experimentally validate the domain heuristics. Unlike R3C, and in line with the suggestions of Hermawati and Lawson [8], PROMETHEUS requires at least one heuristic evaluation in a domain-specific case study. This evaluation must consider the domain heuristics, and a set of control heuristics, which by default can be Nielsen’s 10 heuristics, so as to generate the indicators that are detailed below. It would also be ideal to have several groups of evaluators for each set of heuristics, to minimize the differences between groups. Another difference from R3C is that the evaluators that use the control heuristic must assign a specificity score to each problem found concerning the domain. This score uses the same scale used by the researcher in stage 3.

The validation phase culminates with the calculation of the following indicators:

- **Rate of unique problems**, \( \Phi_P = P_D / P_C \): This represents which group of heuristics found more unique problems. If \( \Phi_P > 1 \), more problems were found in the domain heuristics.
- **Rate of problem dispersion**: To measure the distribution of problems in the groups of heuristics, we consider the values \( \delta_C \) and \( \delta_D \), which represent the standard deviation of problem distribution for the control heuristics and domain heuristics respectively. Given the above, we define the rate of dispersion \( \delta_P = \delta_C / \delta_D \). If \( \delta_P > 1 \), problems are better distributed in the domain heuristics than in the control.
- **Rate of severity**: Similar to the rate of dispersion, we define the rate of severity \( \lambda_P = \lambda_D / \lambda_C \), which represents the relationship between the average severity \( \lambda_D \) of the problems encountered with the domain heuristics, versus the average severity \( \lambda_C \) of the problems encountered with the control heuristics. If \( \lambda_P > 1 \) it means that on average, the domain heuristics found more severe problems than the control heuristics.
- **Rate of specificity**: Finally, we define the rate of specificity \( \varepsilon_P = \varepsilon_D / \varepsilon_C \), which relates specificity averages \( \varepsilon_D \) and \( \varepsilon_C \) of the problems encountered with the domain and control heuristics respectively. If \( \varepsilon_P > 1 \), then the problems found by the domain heuristics are, on average, more specific that those found by the control heuristics.

It is important to remark that the indicators are uniform i.e. when its value is greater than 1, it is considered positive, and we consider that there are potential efficiency problems when its value is less than 1. This is useful, for example, to construct visualizations like Figure 3, which compares the indicator values from various existing studies.

4) Refinement: As shown in Figure 2, the refinement stage takes the heuristic quality indicators \( \Phi_P, \delta_P, \varepsilon_P \) and \( \lambda_P \) into consideration, so as to propose possible refinements to the domain heuristics. Such refinements involve returning to work in one of the specific stages of PROMETHEUS, and then continuing with the critical path as needed. The need for
refinement was born, at least, from the following problematic scenarios, or any combination of them.

- \( \Phi_P < 1 \), that is, the domain heuristics found fewer unique problems. This may indicate that the heuristics are not well adjusted to the domain, that the application examined has mainly general problems, or else that the detailed explanations for the domain heuristics were not understood well by the evaluators.

- \( \delta_P < 1 \), that is, problems are worse distributed in the domain heuristics than in the control. This can be a symptom of overlapping heuristics, too many problems, or even overly specific heuristics, lack of problems, or that they are difficult to implement for evaluators.

- \( \varepsilon_P < 1 \), that is, the problems found by the control heuristics are considered more specific to the domain than those found by the domain heuristics. This problem can indicate problems with the prioritization of domain heuristics, or problems in the experiment design, for example selecting groups of evaluators.

- \( \lambda_P < 1 \), that is, the problems found by the control heuristics are more severe than those found by the domain heuristics. In general, this may indicate problems in the experiment design, in particular with the selection of the groups of evaluators.

The end decision determining when the domain heuristics are considered validated depends ultimately on the researchers, the domain, and the specific applications being studied. However, PROMETHEUS provides a precise methodological process with objective quality indicators, which promote continuous feedback and provide quantitative support to make such a decision.

V. INITIAL VALIDATION

For the initial validation of PROMETHEUS we have used a qualitative approach via a short questionnaire, taken by researchers who used R3C in their work, and a quantitative approach by calculating quality indicators (described in Section III-B) from the existing results mentioned in Section II as far as the data allowed.

A. Questionnaire for investigators that used R3C

We gave a questionnaire (see Section III) to 5 investigators that used R3C in cases studies related to domains of: U-Learning, applications for tablets, transactional web applications, and smartphone applications. At the time the questionnaire was taken, four of the interviewees were senior students in Computer Engineering at the Pontificia Universidad Católica de Valparaíso, and one, Inostroza, the author of [17], was a doctoral student in the final stages of his thesis. Two interviewees worked on tablet domains, but are considered as experts individually, for the purposes of this validation. The objective of the questionnaire was to have expert evaluation of the following criteria:

- Objective of the study
- Applicable stages
- Stages that appear easiest to apply
- Stages that appear hardest to apply
- Recommendations for adding new stages
- Recommendations for eliminating stages
- General comprehension of PROMETHEUS
- Specific comprehension of each stage
- Pertinence of quality indicators
- Ease of calculation of quality indicators
- Other recommendations

We gave each participant a document with a preliminary description of PROMETHEUS, so they could evaluate the methodology’s applicability to their case studies. We also carried out individual sessions to clarify any doubts. After these sessions, the participants had three months to evaluate the applicability of PROMETHEUS.

Results. Table I summarizes the main results from the questionnaires taken. The following information is noteworthy:

- 3 of the 5 participants had the objective of constructing new heuristics, while the other 2 were looking to refine existing heuristics. This has an effect on the potential number of applicable stages, as the investigators from the first group considered more stages of PROMETHEUS applicable to their projects. In the case of U-Learning, 6 of 8 stages were considered applicable. In contrast, for transactional web and smartphone applications, only stage 6 was considered applicable.

- In general, participants considered that the applicable stages would also be easy to apply. The only explicit case of a stage found hard to apply is stage 4 (heuristic normalization), in the U-Learning domain.

- All participants indicated they did not consider it necessary to add or eliminate stages within the methodology, that the general application of the methodology was clear and that the methodology was useful. Regarding comprehension of each particular stage, there were only issues with understanding stage 6 in the case of transactional web applications.

- Unfortunately, the quality indicators were only considered applicable, pertinent and easy to calculate in the case of U-Learning. In general, the other investigators were not interested in calculating them.

- Among the recommendations given, the following stand out:
  - Include an example of how to fill the template in for the “detailed heuristic description” stage.
  - Better explain the calculations included in the methodology.
  - Include a real application example.

B. Validation of quality indicators

Preliminaries. Before showing the obtained results, let us recall the classification of problems proposed in R3C (Section III):

- Common problems, \( P^* \): Common problems identified by both groups of evaluators.
- Domain problems, \( P_D \): Problems only identified by the group of evaluators that used the new heuristics.

...
- **Control problems, \( P_C \):** Problems only identified by the group of evaluators that used Nielsen’s heuristics (or others if applicable) as control.

Now, let \( P \) be the total set of problems found when applying both the control heuristics and the domain heuristics; we have that:

\[
\begin{align*}
P_D^* &= P_D \cup P^* - (\text{domain heuristics problems}) \\
P_C^* &= P_C \cup P^* - (\text{control heuristics problems}) \\
P^* &= P_D \cup P_C \cup P^* = P_D^* \cup P_C^* - (\text{total problems})
\end{align*}
\]

Similarly, in addition we consider the average severity \( \lambda_D^* \) of the problems in \( P_D^* \) and the average severity \( \lambda_C^* \) of the problems in \( P_C^* \). Considering all of the aforementioned, we define the indicators \( \Phi_P = \frac{P_D^*}{P_C^*} \) and \( \lambda_P = \frac{\lambda_D^*}{\lambda_C^*} \) that approximate the uniqueness and severity rates, respectively, in cases where the common problems \( P^* \) are not separated between the two groups of heuristics.

Considering the previous definitions, the second point of the initial PROMETHEUS validation consists in analyzing the data from the 19 studies identified by \[8\] that attempt to determine the effectiveness of the new heuristics (Section \[2\]), and considering two additional recently-published studies \[5\], \[18\]. We calculate the heuristic quality indicators for each of them as possible according to the published information, and we underscore the conclusions regarding whether or not the developed heuristics are more effective than the control heuristics. Table \[1\] summarizes the methodology, the criteria used by the authors, the indicators that could be calculated, and whether or not the authors conclude that the domain heuristics developed are more effective than the control heuristics used. Here we synthesize the main results obtained.

**Regarding the number of problems.** Of the 21 studies analyzed, we were able to calculate an indicator for 17 of them. In these 17 cases, the most commonly used indicator corresponds to the uniqueness rate \( \Phi_P \), used in 9 cases, or the approximate uniqueness rate \( \Phi_P^* \), used in 4 cases. Two other cases only considered the number of problems based on the domain heuristics; in other words, only \( P_D^* \) was considered. In the remaining two cases, no indicator based on the number of detected problems could be calculated.

**Regarding the dispersion of problems.** In a very distant second place, calculating the dispersion rate \( \delta_P \) was possible in four cases, while two cases only had information regarding the problem dispersion for the domain heuristics; in other words, \( \sigma(P_D^*) \).

**Regarding the severity and specificity of the problems.** Similarly, it was possible to calculate the severity rate \( \lambda_P \) in four cases. In 2 cases it was possible to calculate an approximation \( \lambda_P^* \), and in another 2 cases only the problem severity of the domain heuristics, i.e. \( \lambda_D \), was considered. Moreover, it was not possible to calculate the problem specificity rate \( \varepsilon_P \).

**Domain heuristics vs control heuristics.** In 20 of the 21 cases, the authors conclude that the domain heuristics developed (or refined) are more efficient than then control heuristics. The case wherein this does not occur is in the domain of security management applications \[19\], in this case only \( \Phi_P = 1.03 \) could be calculated, which is difficult to interpret without additional information.

**Relation between indicators and positive conclusions.** Figure \[3\] shows the indicators calculated in Table \[1\] breaking each sub-case down as an independent input in the graph. Each vertical line represents a case study involving more than one indicator. A fundamental observation is that the condition \( \Phi_P > 1 \), or its approximation \( \Phi_P^* > 1 \), seems to be the best predictor to validate the domain heuristics’ efficiency. This is not so in only two cases, for \( \Phi_P = 0.93 \) and \( \Phi_P^* = 0.73 \). Under the PROMETHEUS scheme, these cases are difficult to interpret and would require a refinement, as well as calculating the other specified indicators. Furthermore, other qualitative factors may contribute to positive evaluation in these case studies. Regarding the distribution and severity rates, we observed values close to 1, and which complement the uniqueness rates. In just 3 cases, only severity was considered, due to a lack of other quantitative data.

To conclude this section, we observe that the retrospective analysis of the selected cases in which domain heuristics have been developed supports the pertinence of the indicators proposed in PROMETHEUS. Particularly, it is clear that the

---

**TABLE I**

**SUMMARY OF THE INITIAL VALIDATION RESULTS FOR PROMETHEUS**

| Case Study | Goal of the study | Applicable | Easiest | Hardest | Would add? | Would remove? | Understands in general? | Understands each stage? | Useful? | Indicators are pertinent? | Indicators easy to compute? |
|------------|-------------------|------------|---------|---------|------------|--------------|------------------------|------------------------|--------|------------------------|--------------------------|
| U-Learning | Obtener nuevas heurísticas | 1, 2, 3, 4, 7, 8 | 1, 2, 3, 4, 7, 8 | 1 | No | No | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tablets #1 | Obtain new heuristics, refine existing heuristics, describe heuristics in detail | 1, 2, 6, 7 | 6, 7 | – | No | No | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tablets #2 | Obtain new heuristics, refine existing heuristics, describe heuristics in detail | 1, 2, 6, 7 | 1, 2, 6, 7 | – | No | No | ✓ | ✓ | ✓ | ✓ | ✓ |
| Transactional web applications | Refined new heuristics, describe heuristics in detail | 6 | 6 | – | No | No | ✓ | ✓ | ✓ | ✓ | ✓ |
| Smart-phones | Refine existing heuristics, describe heuristics in detail | 6 | 6 | – | No | No | ✓ | ✓ | ✓ | ✓ | ✓ |
Fig. 3. Visualization of quality indicators in selected studies where they could be calculated. Each sub-case is considered individually, thus 23 cases in total. In each case the authors conclude that the domain heuristics are better than those the control. An extreme case with $\Phi_P = 9.1$ has been omitted. The dotted line indicates the value 1, which differentiates between a positive or negative interpretation of the indicator.

most important indicator is the uniqueness rate, and that the other indicators are more complementary and can help discriminate in less conclusive cases. We also observed existent studies that base their positive conclusions on a rather weak quantitative argument, which suggests that the refinement recommendations in PROMETHEUS are on the right track.

VI. CONCLUSIONS AND FUTURE WORK

PROMETHEUS is a PROcedural METHodology for developing HEuristics of USability proposed to increase the robustness of investigations that aim to develop new domain-specific usability heuristics. Based on Hermawati y Lawson’s study [8], and using the R3C methodology as a base [9], we propose a process consisting in 7 principal stages, and a continuous refinement cycle. The main contributions of PROMETHEUS are: (i) it specifies each artifact consumed or generated in detail for each stage, (ii) it defines quantitative quality indicators, and (iii) it defines the refinement based on feedback provided by the indicators. Based on a questionnaire completed by expert investigators, and a retrospective analysis on relevant, state-of-the-art case studies, we have performed an initial validation on PROMETHEUS which is, in our opinion, very positive and suggests new and immediate lines of future work for the additional and complete validation of PROMETHEUS.

In essence, it is necessary to apply PROMETHEUS using real case studies that aim to develop new domain heuristics, and follow the critical path, so as to obtain specific feedback in each of the proposed stages. To this end, we propose applying PROMETHEUS to two new case studies:

1) Virtual Learning Environments, with the case study in question to be found at http://elearning.espoch.edu.ec
2) E-Government Pages. Also based on public services available in Ecuador.

Both of these cases are expected to have at least two groups of evaluators in the experimental validation stage. In addition, semi-structured interviews and questionnaires will be conducted in order to obtain the opinion of everyone involved in these case studies.
| Domain                        | # Hs | Criteria                                                                 | Methodology | Indicators | $H_D$ better than $H_C$? |
|------------------------------|------|---------------------------------------------------------------------------|-------------|------------|-------------------------|
| Grid Computing               | 12   | Quantity of unique, specific and general problems. Severity of specific vs general problems | RJC         | Case 1: $\Phi_D = 2.00, \delta_D = 0.97$, $\lambda_D = 1.16$   | ✓           |
|                             |      |                                                                           |             | Case 2: $\Phi_D = 1.83, \delta_D = 1.11$                     | ✓           |
| Virtual worlds              | 16   | Quantity of unique, specific and general problems. Severity of specific vs general problems | RJC         | Case 1: $\Phi_D = 1.94, \delta_D = 0.71, \lambda_D = 0.91$ | ✓           |
|                             |      |                                                                           |             | Case 2: $\Phi_D = 2.75, \delta_D = 0.78, \lambda_D = 1.07$ | ✓           |
| Inter-cultural web sites    | 13   | Quantity of unique, specific and general problems. Severity of specific vs general problems | RJC         | Case 1: $\Phi_D = 1.97, \lambda_D = 1.11$                     | ✓           |
|                             |      |                                                                           |             | Case 2: $\Phi_D = 1.71, \lambda_D = 1.25$                    | ✓           |
|                             |      |                                                                           |             | (based on percentages)                                       | ✓           |
| Digital Televisio            | 14   | Quantity of unique, specific and general problems. Severity of specific vs general problems | RJC         | $\Phi_D = 5.00, \delta_D = 1.28$                            | ✓           |
| Noticication systems         | 8    | Quantity of problems found on 3 interfaces used as case study. Level of compliance of the heuristics developed | Based on usability problems related to notification systems | Case 1: $\Phi_D = 0.93$                                     | ✓           |
|                             |      |                                                                           |             | Case 2: $\Phi_D = 1.64$                                     | ✓           |
|                             |      |                                                                           |             | Case 3: $\Phi_D = 1.23$                                     | ✓           |
| Assential Robotics           | 9    | Quantity of total and unique problems                                       | Adaptation of existing heuristics                      | $\Phi_D = 4.33$                                           | ✓           |
| Security management          | 7    | Quantity and severity of problems, effectiveness, meticulousness, validity and reliability of the heuristics | It uses grounded theory analysis to generate the heuristics | $\Phi_D = 1.03$                                           | ✓           |
| applications                 |      |                                                                           |             | $\Phi_D = 0.57$                                           | ✓           |
| Educational web sites        | n/a  | Quantity and severity of problems, effectiveness, meticulousness, validity and reliability of the heuristics, time used and associated cost | Own 4 steps methodology that analyzes the characteristics of the domain and evaluates cases through experts and users | $\Phi_D = 9.17$                                           | ✓           |
| Mobile-based applications    | 11   | Quantity and severity of problems                                          | Extends Nielsen’s heuristics                          | $\Phi_D = 1.9$                                           | ✓           |
| Mobile computing             | 8    | Quantity and severity of problems, quantity of problems per heuristic, time used by evaluators | Own 3 steps methodology based on the analysis of 3 experts | Case 1: $P_D^C = 26$, $P_C^D = 22$ $\Phi_D = 1.18$ | ✓           |
|                             |      |                                                                           |             | Case 2: $P_D^C = 38$, $P_C^D = 28$ $\Phi_D = 1.36$         | ✓           |
|                             |      |                                                                           |             | Combined cases: $\delta_D = 0.57$                          | ✓           |
| Learning based in cases      | 22   | Quantity and severity of problems in specific vs general heuristics         | It uses grounded theory analysis to generate the heuristics | Case 1: $P_D^C = 27$, $P_C^D = 37$ $\Phi_D = 0.73$, $\lambda_D^* < \lambda_D^C$ | ✓           |
|                             |      |                                                                           |             | Case 2: $P_D^C = 74$, $P_C^D = 56$ $\Phi_D = 1.32$, $\lambda_D^* > \lambda_D^C$ | ✓           |
|                             |      |                                                                           |             | $\lambda_D^* = 2.39, \lambda_D^C = 2.39, \Phi_D = 0.96$ $\sigma(P_D^C)$ = 3.16 | ✓           |
| Online games                 | 10   | Quantity and severity of problems, quantity and severity of problems per heuristic | Based on the usability analysis on games | $P_D^C = 19$, $P_C^D = 15$, $\Phi_D = 1.27$ | ✓           |
| Mobile map applications      | 10   | Quantity and severity of problems                                          | Adaptation of Nielsen’s heuristics using a theoretical-conceptual focus | $P_D^C = 19$, $P_C^D = 15$, $\Phi_D = 1.27$ | ✓           |
| Smartphones                  | 12   | Quantity of unique, specific and general problems. Severity of specific vs general problems | RJC         | $P_D^C = 28.33$, $\lambda_D^C = 2.28$ $\sigma(P_D^C) = 2.36$ | ✓           |
| Groupware                    | 24   | Problems found in case study and their severity                            | Based on design patterns                              | $P_D^C = 39$, $\sigma(P_D^C) = 1.85$, $\lambda_D^C = 2.15$ | ✓ Without $H_C$, positive conclusions |
| Mobile touch devices         | 12   | Quantity of unique, specific and general problems. Severity of specific vs general problems | RJC         | Case 1: $\lambda_D = 1.37$ | ✓           |
| ERP Application              | 5    | Mean, median and mode of problem severity per heuristic                    | Based on 5 evaluation criteria for ERP | $\lambda_D^* = 1.76; \lambda_D^C = 0.88, \lambda_D^* = 2$ | ✓           |
| Ambient display              | 12   | Percentage and severity of problems identified by both groups of heuristics Porcentaje y | Adaptation of Nielsen’s heuristics taking into account characteristics of the domain and the opinion of experts and designers | The presented results do not permit us to compute the indicators | ✓           |
| Computer games               | 18   | Kinds of problems, quantity and severity                                   | Based on the literature and the review of experts in gameability and game designers | The presented results do not permit us to compute the indicators | ✓           |
| Information display          | 8    | Quantity of problems, quantity of real problems, menticulousness, validity, effectiveness and reliability of the heuristics | It uses critical parameters using scenario-based designs | The presented results do not permit us to compute the indicators | ✓           |
| Web Sites                    | 13   | Quantity of problems, time used by evaluators                              | Adaptation of Nielsen’s heuristics                     | The presented results do not permit us to compute the indicators | ✓           |

1 No comparison against control heuristics
In this section we present a detailed explanation of each of the 8 stages of PROMETHEUS. We describe the purpose of each stage, specify input and output artifacts, and discuss their relevance. As shown in Figure 2, the eight stages of PROMETHEUS are:

1) Search for specific information
2) Search for usability heuristics
3) Heuristic specificity
4) Heuristic normalization
5) Heuristic prioritization
6) Detailed description of heuristics
7) Validation
8) Refinement

A. Stage 1: Search for specific information

The first stage is exploratory in nature, and its main objective is to determine the characteristic dimensions of a specific domain, D. The characteristic dimensions proposed are the following:

1) Usage contexts, UC.
2) Interactive logic devices, LD.
3) Interactive physical devices, PD.
4) User Profiles, UP.

**Input.** The name or description of domain D is required to begin to determine the characteristic dimensions. For example: “grid computing”, “e-learning”, etc.

**Output.** This step produces two artifacts. The first is a list of relevant keywords for the bibliographic search in the next step (Section A-B). The second is the list of characteristic dimensions and their initial specificity indices, which is summarized in tables, such as Table III, one for each characteristic dimension. The specificity values follow a standard 5-level Likert scale, where 0 is not specific and 4 is completely specific. These specificity scores are used in step 5 (Section A-E) for calculating the final specificity index.

| Usage Context | Specificity |
|---------------|-------------|
| Indoor        | 3           |
| Outdoor       | 2           |
| Noisy         | 2           |
| Quiet         | 4           |

**TABLE III**

**INITIAL SPECIFICITY EXAMPLE TABLE FOR USAGE CONTEXT**

An immediate goal for future work is to make a list of pre-selected choices for each characteristic dimension. For example: suggest physical devices such as desktops, tablets, phones, touchscreen, etc., to help complete this stage.

B. Stage 2: Search for usability heuristics

The objective of this stage is to conduct a bibliographical search to identify sets of usability heuristics related to the domain D under study. In general, and as suggested research practice, it is recommendable to perform a systematic mapping or a systematic review of literature [35], [36].

**Input.** At this stage the keywords related to the domain are used, from Stage 1 (Section A-A).

**Output.** At this stage sets of heuristics are obtained $S_1, \ldots, S_n$, which are potentially applicable to the domain. A unique identifier must be assigned to each heuristic, for example, $H_{1S_1}$ represents the first heuristic of the set $S_1$. If there is a set of heuristics validated for the domain and which suits the researchers’ needs, it is possible to terminate the application of PROMETHEUS at this point. These heuristics can also serve as control heuristic in Stage 7 (Section A-G), in the event that new heuristics for the domain are defined.

In general, Stages 1 and 2 specify how to carry out the process of extracting information, as reported by Hermawati and Lawson [8], giving specific recommendations on how to conduct the search, and taking into account the determining aspects of the domain under study. As mentioned in Section VI, and in this appendix, there is considerable potential for improvement in these stages.

C. Stage 3: Heuristic specificity

In order to perform an initial filtering, at this stage we assign an initial specificity index ISI, associated with each heuristic found in the previous stage.

**Input.** Heuristic sets $S_1, \ldots, S_n$, identified in the previous stage, are required.

**Output.** At this stage a tabulation of heuristics is obtained with their initial specificity indices, as shown in Table IV. We say that these heuristics are denormalized, since they don’t resolve duplication and/or overlap conflicts between the heuristics yet. However, it is possible to rank the heuristics and eliminate those considered unlikely to be applied to the domain.

| Heuristic ISI |
|---------------|
| $H_{1S_1}$    | 3            |
| $H_{kS_k}$    | 4            |
| $H_{nS_n}$    | 5            |

**TABLE IV**

**SPECIFICITY INDICES FOR DENORMALIZED HEURISTICS**

D. Stage 4: Heuristic normalization

The objective of this stage is to resolve possible cases of overlap or duplication in the heuristics obtained so far.

**Input.** The denormalized heuristics are received as input, along with their initial specificity indicators.

**Output.** At this stage a preliminary set of heuristics applicable to the domain is produced, ensuring that there are no overlapping or duplication problems between heuristics. Each heuristic will have an ISI score, possibly revised after normalization.

In general, cases of multiplicity can be solved through any of the following:
• Keep one of the heuristics that have similarity and discard the others.
• Discard similar heuristics and reformulate a new heuristic that combines the characteristics of the heuristics that make it up.

Thus, cases of overlapping can be solved by any of the following strategies:
• Maintain a general heuristic that groups together the overlapping heuristics.
• Separate the heuristic overlapping into several individual heuristics.

This process of resolving overlap and multiplicity should be carried out iteratively until obtaining a set of normalized heuristics, in other words, a set without any overlaps or multiplicities. Note that if new heuristics are created, following the aforementioned strategies, these must also have a unique identifier. Finally, it is recommended to review and reconsider the values of the ISI specificity for each one the normalized heuristics.

E. Stage 5: Prioritization of Heuristics

The objective of this stage is to synthesize the applicability of the normalized heuristics, considering the specificity of each one with respect to context of use, as well as the initial specificity indicators, to have a rankable list of heuristics that can be applied to the domain.

Input. This stage receives the normalized heuristics obtained in the previous stage, together with the respective ISI indicators. In addition, it works with characteristic dimensions, UC, PD, LD, UP; obtained in Stage 1.

Output. The principal product for this stage is the specificity matrix, as per the example in Table VI, which facilitates a final process of ranking and selecting the heuristics that can be considered most applicable for the domain.

### Table VI
**Specificity for Context of Use and GSI<sub>UC</sub> Calculation**

| Heuristic | Indoor | Outdoor | Noisy | Quiet | GSI<sub>UC</sub> |
|-----------|--------|---------|-------|-------|------------------|
| H<sub>23</sub> | 4      | 4       | 4     | 4     | 4                |
| H<sub>23</sub> | 4      | 0       | 0     | 0     | 1                |
| H<sub>24</sub> | 4      | 2       | 2     | 4     | 3                |
| :         | :      | :       | :     | :     | :                |

Finally, the final specificity index FSI synthesizes the specificity and applicability of the different heuristics to the domain in a single value, considering the initial evaluation and the evaluation of each characteristic dimension. For a heuristic H<sub>j</sub>, the index is calculated with the following formula:

\[
FSI_{H_j} = 4 \times \frac{ISI_{H_j} \times \sum GSI_{(UC,LD,PD,UP)}}{64}
\]

The final value of FSI varies between 0 and 4; in general, a high value of FSI is expected to indicate high applicability of that heuristic, although the final selection criterion is always within the discretion of the researchers involved, with the methodological backing granted by PROMETHEUS.

F. Stage 6: Detailed description of heuristics

At this stage the description of the heuristics selected in the previous stage are formalized, in order to design the experimental validation for Stage 7. This formal description can provide information necessary to understand and apply the domain heuristics when performing a heuristic evaluation, particularly for inexperienced evaluators, as evidenced in [16]. Table VII presents a template format that can be used at this stage. This template takes the basis proposed by R3C [9], and adds a new field, Checklist, with the intention to further facilitate its application, following the suggestions in [16].

Input. The selected set of heuristics for application to the domain.

Output. The formal description of each heuristic, based on the standard template.

### Table VII
**Standard Template to Describe Usability Heuristics.**

| Heuristic Identifier | Name | Name that identifies the heuristic. |
|----------------------|------|-------------------------------------|
| Description          | Detailed explanation of the heuristic. |
| Benefits             | Expected usability benefits when there is compliance with the heuristic. |
| Problems             | Expected problems that can arise if the heuristic is misunderstood during heuristic evaluation. |
| Application context  | Additional information regarding the applicability of the heuristic. |
| Related heuristics   | References to other heuristics related to the (non-)compliance of this heuristic. |
| Checklist            | Detailed operational steps and criteria to be used when applying this heuristic. |

This matrix contains, for each heuristic H<sub>j</sub>, a summary of the following specificity indicators:
• ISI: initial specificity indicators, created in Stage 1 and refined in Stage 4.
• GSI<sub>UC</sub>, GSI<sub>PD</sub>, GSI<sub>LD</sub> and GSI<sub>UP</sub>: global specificity indicators of each characteristic dimension. To exemplify the calculation of these indicators, we consider the UC dimension, for which we created a tabulation such as Table VII. For each usage context, a specificity score was assigned, and the GSI<sub>UC</sub> score is the average of each row. Similarly, tables were created for the other characteristic dimensions.
G. Stages 7 & 8: Validation and Refinement

The ultimate purpose of PROMETHEUS culminates in the stages of validation and refinement, in which it seeks to empirically validate that the selected domain heuristics are more effective than a control group of heuristics. As explained in Section IV, PROMETHEUS defines quantitative quality indicators for this comparison, and to guide the process of iterative refinement if necessary. Of course, it is always advisable to perform additional validations, especially qualitative; PROMETHEUS provides a solid foundation for validating domain heuristics.

**Input.** The selected domain heuristics, along with their formal description. Additionally, the set of control heuristics to be used. By default Nielsen’s heuristics [7] should be used as a control group, unless there are more specific heuristics and they are already validated.

**Output.** After the experimental evaluation, the following artifacts are obtained:

- Set of problems detected in the heuristic evaluation(s). The problems are classified into common problems, $P^*$, problems specific to the domain heuristics, $P_D$, and problems specific to the control heuristics, $P_C$. Each problem is associated with a particular heuristic.
- Each problem’s severity is specified, using a 5-level Likert scale.
- For each problem in $P^* = P^* \cup P_C$, a specificity coefficient, with the same kind of scale as for severity.

In addition, with this information the following quality indicators are calculated:

- Unique problem rate $\Phi_P$.
- Problem dispersion rate $\delta_P$.
- Problem severity rate $\lambda_P$.
- Problem specificity rate $\varepsilon_P$.

Regarding the indicators, those were described in Section IV and we only need to define the rate of specificity. Let us recall that this indicator is defined as follows:

$$\varepsilon_P = \frac{\varepsilon_D}{\varepsilon_C}$$

where $\varepsilon_D$ and $\varepsilon_C$ are the specificity averages for the problems found, respectively, by the domain heuristics and the control heuristics.

The point we must explain is that the $\varepsilon_D$ and $\varepsilon_C$ calculations are different, although they aim to quantify the same phenomenon: how specific the problems encountered are. The problem is that when the control heuristics are very general, for example Nielsen’s, it is difficult to decide when a problem is domain-specific. For that reason, the problems in $P_C^*$ need to be qualified according to specificity. That is to say, $\varepsilon_C$ corresponds to the simple specificity average for each problem. Nevertheless, on the other hand, the domain heuristics have been designed and selected based on their specificity indices – in particular $FSI_j$, which synthesizes this characteristic on a heuristic level. Therefore, $\varepsilon_D$ corresponds to a weighted sum of the number of problems in each heuristic, multiplied by the $FSI_j$ associated with it. In summary, we have that:

$$\varepsilon_D = \sum_{H_j} \left( \frac{|P \in H_j^D| \times FSI_j}{|P_D^*|} \right)$$

$$\varepsilon_C = \sum_{i=1}^n \frac{\varepsilon(P_i)}{|P_C^*|}$$

for all domain heuristics $H_j$, where:

- $|P \in H_j|$ is the number of problems associated with said heuristic.
- $|P_D^*|$ and $|P^*|$ correspond to the number of domain and control problems, respectively.
- $FSI_j$ is the final specificity index for said heuristic $H_j$.
- $n$ is the total number of problems in $P_C^*$.
- $\varepsilon(P_i)$ is the individual specificity of a given problem $P_i$, and associated with the control group problems.

**APPENDIX B**

**INTERVIEW FOR THE INITIAL EVALUATION OF PROMETHEUS**

In this section we present the interview conducted as part of the initial validation of PROMETHEUS, designed to collect information in respect to the perception of PROMETHEUS as a methodology to create usability heuristics, based on the following aspects and whose questions are summarized in Table VIII.

- Method of application
- Interest or final product
- Stages applied
- Easiest and/or most difficult stages
- Usefulness
- Quantification of application time
- Validation mechanisms of the generated products
- Future plans for application of the process
### TABLE VIII
QUESTIONS FROM THE INTERVIEW EXECUTED IN THE INITIAL VALIDATION OF PROMETHEUS

| Question                                                                 | Purpose                                                                 |
|--------------------------------------------------------------------------|Adam| Date: 28/04/2022 |
| ¿What is your subject of study?                                          | Determine the specific domain in which the researcher is working. |
| What product or products do you expect to produce in your research?     | Understand whether the researcher is developing new heuristics, refining existing ones, or making a detailed description of heuristics. |
| Which stages of PROMETHEUS do you think are applicable to your research? | Determine the potential applicability of stages in PROMETHEUS to the particular project of the researcher. |
| Considering all potentially applicable stages, which ones do you consider would be the easiest to apply? | Determine the ease of application of applicable stages of PROMETHEUS. |
| Considering all potentially applicable stages, which ones do you consider would be the hardest to apply? | Determine which applicable stages of PROMETHEUS are considered as difficult to apply. |
| Do you consider it necessary to add new stages or activities to the methodology? | Obtain experts' opinion about the proper quantity of stages or activities in PROMETHEUS. |
| Do you consider it necessary to remove or consolidate stages or activities in the methodology? | Obtain experts' opinion about the proper quantity of stages or activities in PROMETHEUS. |
| Do you understand in general terms how to apply the methodology?        | Identify the perception regarding the general clarity of the process. |
| Do you understand how to apply each individual stage?                   | Identify the perception regarding the clarity of the description of each stage of the process. |
| Do you consider in general terms that the methodology is useful, as well as each of its stages? | Identify the perception regarding the usefulness of the methodology. |
| Do you consider that the quality indicators defined in PROMETHEUS are relevant and points towards an effective quantitative validation? | Obtain experts' opinion regarding the pertinence of the quality indicators. |
| Do you consider that the quality indicators defined in PROMETHEUS are easy to compute? | Obtain experts' opinion regarding the ease of computation of the quality indicators. |