Developing Usability Heuristics with PROMETHEUS: A Case Study in Virtual Learning Environments

Ismael Figueroa, Cristhy Jiménez, Hector Allende-Cid, Paul Leger

ISSN: 0920-5489
ELSEVIER
COMPUTER STANDARDS & INTERFACES
The International Journal on the Development and Application of Standards for Computers, Software Quality, Data Communications, E-topics, Interfaces and Measurement

PII: S0920-5489(18)30178-8
DOI: https://doi.org/10.1016/j.csi.2019.03.003
Reference: CSI 3335

To appear in: *Computer Standards & Interfaces*

Received date: 18 May 2018
Revised date: 2 March 2019
Accepted date: 25 March 2019

Please cite this article as: Ismael Figueroa, Cristhy Jiménez, Hector Allende-Cid, Paul Leger, Developing Usability Heuristics with PROMETHEUS: A Case Study in Virtual Learning Environments, *Computer Standards & Interfaces* (2019), doi: https://doi.org/10.1016/j.csi.2019.03.003

Highlights

- New heuristics in Virtual Learning Environments performing better Nielsen's
- Application and validation of PROMETHEUS methodology
- We propose a rigorous statistical approach for results validation

Developing Usability Heuristics with PROMETHEUS: A Case Study in Virtual Learning Environments

Ismael Figueroa[a], Cristhy Jiménez[b], Hector Allende-Cid[a], Paul Leger[c]

[a]Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso
Valparaíso, Chile
[b]Facultad de Ciencias de la Educación, Humanas y Tecnoloías
Universidad Nacional de Chimborazo
Riobamba, Ecuador
[c]Escuela de Ingeniería, Universidad Católica del Norte, Coquimbo, Chile

Abstract

Heuristic evaluation is one of the most widely-used methods for evaluating the usability of a software product. Proposed in 1990 by Nielsen and Molich, it consists in having a small group of evaluators performing a systematic revision of a system under a set of guiding principles known as *usability heuristics*. Although Nielsen's 10 usability heuristics are used as the *de facto* standard in the process of heuristic evaluation, recent research has provided evidence not only for the need of custom *domain specific* heuristics, but also for the development of methodological processes to create such sets of heuristics. In this work we apply the PROMETHEUS methodology, recently proposed by the authors, to develop the VLEs heuristics: a novel set of usability heuristics for the domain of virtual learning environments. In addition to the development of these heuristics, our research serves as further empirical validation of PROMETHEUS. To validate our results we performed an heuristic evaluation using both VLEs and Nielsen's heuristics. Our design explicitly controls the effect of evaluator variability by using a large number of evaluators. Indeed, for both sets of heuristics the evaluation was performed independently by 7 groups of 5 evaluators each. That is, there were 70 evaluators in total, 35 using VLEs and 35 using Nielsen's heuristics. In addition, we perform rigorous statistical analyses to establish the validity of the novel VLEs heuristics. The results show that VLEs perform better than Nielsen's heuristics, finding more problems, which are also more relevant to the domain, as well as satisfying other quantitative and qualitative criteria. Finally, in contrast to evaluators using Nielsen's heuristics, evaluators using VLEs heuristics reported greater satisfaction regarding utility, clarity, ease of use, and need of additional elements.

Email addresses: `ismael.figueroa@pucv.cl` (Ismael Figueroa), `cjimenez@unach.edu.ec` (Cristhy Jiménez), `hector.allende@pucv.cl` (Hector Allende-Cid), `pleger@ucn.cl` (Paul Leger)

Keywords: usability, heuristics, methodology, evaluation methods, prometheus, virtual learning environments

1. Introduction

Usability is a key aspect in the quality of a software product. According to the ISO 9241 definition, its purpose is 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"* [12]. Note that the definition makes very explicit the need to *evaluate* a software product to establish whether or not it is usable, however it does not provide any hints as to how to perform this task. Although there exist many usability evaluation methods [34] such as verbal protocols, different kinds of walkthroughs, or several ways of inquiring users, this work is focused on the method of *heuristic evaluation*. We chose this approach mainly because it is generally more affordable than extensive user tests, which may be too application-specific, and also because of a growing trend in research regarding this area, as demonstrated for instance in our previous work [13, 2]. Still, we believe that any in-depth evaluation of a system should include first an heuristic evaluation, which is followed by relevant user-tests, in order to gain direct feedback both from experts in the field of usability, as well as from real users.

Heuristic evaluation was proposed by Nielsen and Molich [24] as an affordable systematic method to detect problems in the interface of a software product. It consists in asking a small group of evaluators to examine a system to detect and classify problems based on set of guiding principles known as *usability heuristics*. Upon its introduction in 1990 heuristic evaluation has been proven as an effective mechanism for evaluating usability, traditionally using the so-called "Nielsen's Heuristics" [21] as the *de facto* standard for this kind of evaluations. However, recent research [8, 28] has provided evidence not only for the need of finer-grained *domain-specific heuristics*, but also for the need of formal methodological processes for the construction of these specially-tailored sets of heuristics.

The use of Virtual Learning Environments (VLEs) as the domain for our case study arises from the teaching activity at the ecuatorian university "Superior Polytechnic School of Chimborazo" (ESPOCH). Here, one of the authors proposed the analysis of the institutional virtual learning environment as a matter of coursework study, given the importance of user acceptance in its effect on learning and teaching processes [36]. However, we realized that common Nielsen's heuristics were not sufficiently adaptable to the specific features of VLE systems. As a consequence, and considering this situation as a good case study on the application of PROMETHEUS [13, 2], we decided to develop a new set of usability heuristics applicable to this specific domain.

To summarize, in this work we present the development of domain heuristics in the specific domain of *virtual learning environments*. To do so, we use the

PROMETHEUS methodology, recently proposed by the authors in previous work [13, 2]. More specifically, the main contributions of this work are:

1. We develop VLEs: a new set of usability heuristics for the domain of virtual learning environment, which we validate against the standard Nielsen's heuristics as a control. The empirical results show that VLEs heuristics perform better than control heuristics, mainly because they found more problems overall, and also because they found many important problems not detected by using the set of control heuristics.

2. We use the development of VLEs as a case study on the application of PROMETHEUS. Indeed, this is the first application of PROMETHEUS as a methodology for establishing domain-specific usability heuristics. This work serves as further validation that indeed PROMETHEUS is a suitable methodology for the creation of domain heuristics.

3. We perform rigorous statistical analyses to establish the validity and effectiveness of the VLEs heuristics. In addition, we complement these analyses with the quantitative analysis proposed in PROMETHEUS. To the best of our knowledge, this work presents one of the more detailed and rigorous quantitative approaches to validate new domain-specific heuristics.

As mentioned before, we validate the VLE heuristics by performing an heuristic evaluation of the ESPOCH virtual learning environment, using Nielsen's heuristics as the comparison baseline. In order to control the effect of evaluator variability, we use a large number of evaluators: for both VLE and Nielsen's heuristics there are 7 independent groups of 5 evaluators each. In other words, there are 70 evaluators in total, 35 using VLE and 35 using Nielsen's heuristics. As we discuss later, all evaluators are senior students of an HCI course, which is not too uncommon in recent research *e.g.* [11]. From the preliminary training of evaluators until the completion of heuristic evaluation, the process took around 6 months, starting in the second semester of 2017.

The rest of this article is structured as follows: Section 2 provides the theoretical background about heuristic evaluation, the methodologies for developing domain-specific heuristics, as well as background on usability evaluation in virtual learning environments. Then Section 3 describes the development of the VLE heuristics as the result of the application of PROMETHEUS, and also the empirical and statistical validation of the results. Later, Section 4 discusses relevant related work. Finally, Section 5 presents the discussion of the results, as well as the directions for future work.

2. Background

Usability is a complex discipline concerned with the activities, concepts and processes that promote and facilitate the design of human-computer interfaces that take the needs of users into consideration. As summarized by Hartson et al. [7] and elsewhere, and also as we discussed in the introduction, there are several methods to evaluate usability. Because in this work we focus exclusively in the method of heuristic evaluation, this section provides the necessary background on this technique, the recent methodological developments for the construction of domain-specific heuristics, as well as a brief overview on the usability considerations in the field of virtual learning environments.

2.1. Heuristic Evaluation and Domain Heuristics

Heuristic evaluation is a usability evaluation method developed originally by Nielsen and Molich [24] in 1990. It consists in a systematic evaluation in which several experts—usually 3 to 5—analyze the interface of an interactive system and make observations based on a set of principles known as *usability heuristics*. The goal of heuristic evaluation is to discover problems related to the usability in the interface design, in order to solve them during an iterative software development process. According to Nielsen [22], the protocol of a typical heuristic evaluation involves that each evaluator spends time inspecting the system interface in an independent manner, making note of every usability problem found. Once all evaluators have complete the evaluation, their individual findings are joined in a unique common list of usability problems associated to those heuristics that were violated. The evaluators must rate the *severity* of each problem, according to a scale of *0* (lowest severity) to *4* (highest severity) [23]. Finally, the evaluators develop a heuristic evaluation report, highlighting major usability problems and incorporating suggestions for fixing them.

In his subsequent work on heuristic evaluation Nielsen [21] proposed 10 general usability heuristics for the design and evaluation of interfaces. Known usually as “Nielsen’s heuristics”, they have become the *de facto* set of heuristics used for the application of heuristic evaluations. However, recent surveys on the development of usability heuristics [8, 28], have shown that the development of *domain-specific heuristics*—specially-tailored to the problems and needs of a given domain—is gaining traction. Indeed, in the 70 studies surveyed by Hermawati and Lawson [8] there are at least 10 different domains where new usability heuristics were devised, including for instance: virtual reality, web sites, mobile games and computing, e-learning environments, amongst many others. These surveys have also detected the need for novel methodological processes for the development of domain heuristics.

2.2. Methodologies for Developing Usability Heuristics

To contextualize the contributions of this work we describe the current and most related methodological approaches for the development of usability heuristics: the R3C methodology, the QRR methodology, and the PROMETHEUS methodology. Then, we present a side-by-side comparison between those approaches.

2.2.1. The R3C *Methodology*

Originally proposed by Rusu et al. [31] in 2011 the R3C methodology, based on the surnames of its authors, proposes 6 explicit stages for the development of domain heuristics:

1. An **Exploratory Stage:** to collect bibliographical information related to the given domain, its specific applications, and any existing heuristics, if any.

2. **Descriptive Stage:** to highlight the most important characteristics of the information previously gathered, aiming to formalize the main concepts of the research.

3. **Correlational Stage** to identify what are the characteristics that the new heuristics should have, based on traditional heuristics, and the analysis of case studies.

4. **Explicative Stage** to formally specify the set of proposed heuristics, based on a standard template.

5. **Validation Stage** to contrast the heuristics under development with the traditional ones, by means of experiments, heuristic evaluations, case studies and user tests.

6. **Refinement Stage** based on the feedback from the validation stage.

R3C has been used successfully by several researchers, specifically in domains such as: grid computing [32], virtual worlds [19], inter-cultural web sites [5] , digital television [35], mobile touch devices [10], and most recently in the domain of smartphones [11]. Most notably, the definition of R3C is very succinct, and it does not specify any specific artifacts to be produced (or required) by the given stages.

2.2.2. The PROMETHEUS Methodology

PROMETHEUS [13, 2] is a procedural methodology for developing heuristics of usability.[1] In this section we summarize the essence of PROMETHEUS, by highlighting its critical path for the development of domain-specific heuristics, and also by describing the quantitative quality indicators that are used later in the evaluation of the case study (Section 3). The complete proposal and description of PROMETHEUS is available in our previous work [2, 13].

Critical Path. PROMETHEUS is an iterative process comprised of 7 main steps, as well as an 8th step dedicated to refinement decisions. It takes the general 6 steps from R3C (Section 2.2.1) and elaborates on them by precisely defining the inputs and outputs of each step. In general terms, PROMETHEUS takes as input a given domain, *e.g.* smartphones, and yields a set of *domain heuristics* to be used in heuristic evaluations of particular applications in the domain. The 7th step of PROMETHEUS requires the evaluation of the domain heuristics being developed, against a set of *control heuristics*. By default,

[1] The original paper on PROMETHEUS is in Spanish, hence we provide a translation [2] performed by the same authors.

Nielsen's heuristics can be used as the control heuristics. As the final step, PROMETHEUS requires the computation of the quality indicators, which then guide the refinement process.

Quality Indicators. PROMETHEUS defines four quality indicators in order to quantify the evaluation between the domain heuristics H_D and the control heuristics H_C, in the context of an heuristic evaluation. The quality indicators are related to the total quantity of problems found with each set of heuristics, as well to the uniqueness, severity and criticality of the problems found. To describe the indicators we must first define the following sets of problems, based on the initial description of R3C [31]:

- *Common Problems (P^*)*: problems identified by both sets of evaluators, that is, found both by H_D and H_C.
- *Domain Problems ($\mathbb{P}_\mathbb{D}$)*: problems identified only by evaluators using the domain heuristics H_D.
- *Control Problems ($\mathbb{P}_\mathbb{C}$)*: problems identified only by evaluators using the control heuristics H_C.

Given these definitions, PROMETHEUS defines the following quality indicators. In order to provide a consistent interpretation, all indicators are defined such that a value greater than 1 means that H_D is performing better than H_C.

Unique Problems Ratio $\varPhi_P$: it quantifies which one of H_D or H_C found a greater number of unique problems. If $\varPhi_P > 1$, it means H_D found more unique problems. It is defined as:

$$\varPhi_P = \frac{\mathbb{P}_\mathbb{D}}{\mathbb{P}_\mathbb{C}} \tag{1}$$

Dispersion Ratio δ_P: it measures the distribution of problems over the heuristics of a given set of heuristics. Overall, we expect that a smooth distribution of problems over heuristics is better than having too many problems assigned to a given heuristic that would appear as too general. When $\delta_P > 1$, it implies that problems are better distributed in H_D than in H_C. It is defined as:

$$\delta_P = \frac{\delta_C}{\delta_D} \tag{2}$$

where δ_D and δ_C denote the standard deviation of the problem distribution for both the domain and control heuristics, respectively.

Severity Ratio λ_P: it quantifies the severity of problems found by H_D in relation to H_C. When $\lambda_P > 1$, it means that in general H_D found more severe problems, hence it is providing better feedback respect to issues that must be addressed. It is defined as:

$$\lambda_P = \frac{\lambda_D}{\lambda_C} \tag{3}$$

where λ_D and λ_C denote the average severity of problems for both H_D and H_C, respectively.

Specificity Ratio ε_P: similarly, this indicator measures the specificity of problems with respect to the given domain. When $\varepsilon_P > 1$, it means that in general H_D found problems more specific to the domain. It is defined as:

$$\varepsilon_P = \frac{\varepsilon_D}{\varepsilon_C} \tag{4}$$

where ε_D and ε_C denote the average specificity of problems for H_D and H_C, respectively.

Preliminary Evaluation. In our previous work [13, 2] we performed a preliminary evaluation of PROMETHEUS where we:

(a) Applied a questionnaire to researchers that used the R3C methodology.

(b) Performed a retrospective analysis of the proposed quality indicators in the context of previously developed domain-specific heuristics, following the survey done by Hermawati and Lawson [8] on 70 published articles.

As a first result, we found that most of the researchers that used R3C found PROMETHEUS to be quite applicable to their current and future research. In addition we considered the 70 studies surveyed by Hermawati and Lawson [8] and computed the quality indicators in the 21 cases where it was possible. We concluded that, although implicitly, Φ_P is by far the most important criteria used by researchers to determine when the domain heuristic performed better than the baseline comparisons of each study. In contrast, the dispersion, severity and specificity are only considered as complementary factors, if they are used at all.

2.2.3. The QRR Methodology

Recently proposed by Quiñones et al. [29], the QRR methodolodgy, based on the surnames of its authors, is similar in origin and in spirit to PROMETHEUS. Indeed, QRR also adapts and extends the R3C methodology of Rusu et al. [32] into 8 refined steps. Overall, their work is quite similar to our PROMETHEUS's proposal [13, 2]. We highlight the main similarities and differences between both methodologies:

- Both PROMETHEUS and QRR take the R3C methodology as a starting point. In hindsight it is simple to see that most similarities probably arise from this fact, and from the concurrent development and evaluation of both methodologies.

- QRR introduces the criteria of *expert judgment* for the evaluation of newly developed heuristics, in addition to the experimental validation present in PROMETHEUS.
- The refinement stage of QRR, in contrast to PROMETHEUS, does not formally define quality indicators.
- In their case study, Quiñones et al. [29] compare the output of 2 groups of evaluators, one using the control heuristics, and one using the newly developed heuristics. In our work we explicitly tackle the potential effect of evaluator bias, by performing a rigorous statistical analysis on the results of 14 groups of evaluators, with 7 groups using the control heuristics and 7 groups using the newly developed heuristics.

2.2.4. Comparison

Now we present a comparison between R3C, QRR, and PROMETHEUS. In the case of R3C we consider the most recent paper [11] that uses the methodology, which defines usability heuristics for smartphones. We consider the following points for comparison:

- Whether it is based on a previous methodology
- Quantity of steps or stages
- How many evaluators were used in the validation case study
- What kind of statistical tests were used in the validation case study
- Domain of the case study
- Whether the methodology formally defines quality indicators

As shown in Table 1, both QRR and PROMETHEUS are based on R3C, and share the same core design: 8 steps, requiring an empirical validation of the developed heuristics.

However, there are several differences in the validation case studies proposed for QRR and PROMETHEUS. For instance, in [29] the authors explain in great detail all the steps of the methodology, as well as the application of questionnaires to assess the opinion of experts regarding the utility, ease of use, clarity and other factors of the methodology. Then, they use statistical analysis to determine the correlations between the questions and the perceived dimensions. The experimental validation—by actually developing heuristics with the methodology—is briefly described in a section of the paper (see [29, Section 6]), however there are many details that are not reported, such as: quantity of problems, quantity of evaluators, expertise level of evaluators, among others. On the other hand, the evaluation presented in this paper is based on the results of the application of PROMETHEUS in the domain of virtual learning environments. A crucial difference between QRR and PROMETHEUS is that we

	R3C [11]	**QRR** [29]	**PROMETHEUS**
Based on previous method?	no	on R3C	on R3C
Stages	6	8	8
Domain of case study	Smartphones	National Parks websites	Virtual Learning Environments
Evaluation design	Heuristic evaluation control vs domain heuristics		
Quantity of evaluators	27, only using newly-developed heuristics	Not reported	**70**, split into 14 groups of 5. Each set of heuristics was assigned to 7 groups
Profile of evaluators	Students of HCI course	Not reported	Students of HCI course
Evaluation criteria	Problems found on case study	Expert judgment	Problems found on case study
Formally defined quality indicators	no	no	**yes**

Table 1: Comparison between R3C, QRR, and PROMETHEUS methodologies

define several quality indicators that are defined with formulas that are simple, yet can be interpreted more objectively to provide guidance in the refinement process of the heuristics under development. However, so far PROMETHEUS lacks the application of expert judgments as is present in QRR.

2.3. Virtual Learning Environments and Heuristics

The integration of Web in the learning processes marks a before and after in the development of eLearning platforms. Currently, educational worlwide institutions have implemented their own virtual learning environments (VLEs) for supporting learning-teaching activities by using technological tools [4]. In this sense, the virtual users' demands have significantly increased, focusing their expectations on how to get virtual learning systems with better functionality and design for allowing them to improve the levels of satisfaction and motivation during their teaching-learning activities [3].

There is a vast amount of literature regarding the study and design of VLEs, some of these works have been focused on the identification or implementation of strategies for designing the VLEs' contents, towards the construction of environments that promote the development of the connectivist learning paradigm [1, 33, 15, 39]. In other studies, authors have recognized that a critical factor to successful implementation of VLEs is *user's acceptance*, represented in most cases as the the *ease of use* attribute of an interactive system [36, 17]. Later in Section 4 we discuss about heuristic evaluation in the context of virtual learning environments.

3. New Usability Heuristics for Virtual Learning Environments

The fundamental premise underlying the evaluation of PROMETHEUS is to evaluate the quality of the methodology by measuring the quality of the domain heuristics produced by its application. In other words, if novel heuristics developed with PROMETHEUS perform better than traditional control heuristics—such as Nielsen's—then we can conclude that the PROMETHEUS methodology is working well. Following this idea, the empirical evaluation presented in this section has the following stages:

Stage 1: We apply PROMETHEUS to generate a set of *VLE heuristics*, that is, a set of heuristics specific to the domain of virtual learning environments.

Stage 2: Using Nielsen's heuristics as the set of control heuristics, we perform an heuristic evaluation of the ESPOCH[2] virtual learning environment. The system is evaluated independently by groups of 5 evaluators. There are 7 groups using VLEs and another 7 groups using Nielsen's heuristics.

Stage 3: To conclude the empirical evaluation of PROMETHEUS we perform the following validations:

(a) *Intra-heuristics validation*: to assess the variability in the groups of evaluators, regarding quantity of problems, problem dispersion and other quantitative measurements. The goal is to determine whether VLE heuristics have a better performance than Nielsen's, regarding the quantity of problems and other quantitative measurements.

(b) *Inter-heuristics validation*: here we compute PROMETHEUS' quality indicators to compare the relative quality of the results of VLEs and Nielsen's heuristics.

(c) *Perception Questionnaire*: we asked all evaluators about their perception regarding four dimensions: Utility, Clarity, Ease of Use, and Need of Additional Elements.

The following subsections describe in detail each of these stages.

3.1. Stage 1: Development of VLE heuristics

The development of the VLE domain heuristics was done by a group of 3 researchers, which applied PROMETHEUS under the guidance and supervision of the authors. All of them were, at the time, senior students of the Systems Engineering career, at the ESPOCH college, with an homogeneous background and profile, as detailed in Table 2. Prior to the application of PROMETHEUS, the researchers performed preliminary training on how to properly apply the methodology, the design philosophy behind PROMETHEUS, its stages, intermediate products, and the quality indicators (Section 2.2.2). Afterwards, they were tasked with the elaboration of VLE heuristics, within a timeframe of 2 months. Overall, the process did not have unexpected complications.

[2] ESPOCH means *Superior Polytechnic School of Chimborazo* from its original meaning in Spanish. Chimborazo is a province located in Ecuator. The inspected system is available at https://elearning.espoch.edu.ec/

Participant	Gender	Age	Occupation	Experience using VLEs
#1	Male	22	Senior student, Systems Engineering	Frequent user
#2	Male	23	Senior student, Systems Engineering	Frequent user
#3	Female	23	Senior student, Systems Engineering	Frequent user

Table 2: Profile of participants that applied PROMETHEUS to develop domain heuristics for VLEs.

H_D: VLEs Heuristics		H_C: Nielsen's Heuristics	
ID	**Definition**	**ID**	**Definition**
VH1	Visibility of system status	**NH1**	Visibility of system status
VH2	Match between system and the real world	**NH2**	Match between system and the real world
VH3	User control and freedom	**NH3**	User control and freedom
VH4	Consistency and standards	**NH4**	Consistency and standards
VH5	Error prevention	**NH5**	Error prevention
VH6	Recognition rather than recall	**NH6**	Recognition rather than recall
VH7	Flexibility and efficiency of use	**NH7**	Flexibility and efficiency of use
VH8	Aesthetic and minimalist design	**NH8**	Aesthetic and minimalist design
VH9	Help users recognize, diagnose, from errors and recover from errors	**NH9**	Help users recognize, diagnose, from errors and recover from errors
VH10	Help and documentation	**NH10**	Help and documentation
VH11	System elements consistency	**NH4**	Consistency and standards
VH12	Web standards and symbols	**-**	-
VH13	Teaching-Learning process indicator	**-**	-
VH14	Flexible configuration of resources and learning objects	**-**	-
VH15	Storage capability	**-**	-
VH16	Interactive communication	**-**	-
VH17	Multiple devices adaptation	**NH7**	User control and freedom
VH18	Measuring learning	**-**	-

Table 3: Mapping between VLE and Nielsen's Heuristics

Table 3 depicts the 18 resulting usability heuristics, which are shown as a side-by-side mapping with Nielsen's original heuristics [21]. From now on we refer to VLEs heuristics as the *domain heuristics* H_D and to Nielsen's heuristics as the *control heuristics* H_C. H_D heuristics are identified as VH1 up to VH18, whereas H_C heuristics are identified as NH1 to NH10.

As it can be seen in Table 3, the resulting VLEs heuristics keep Nielsen's heuristics unchanged and add 8 new heuristics, from VH11 to VH18, that consider features that are specific to the domain of VLEs. Still, heuristics VH11 and VH17 can be considered as refined instances of heuristics NH4 and NH7, respectively. The reason why Nielsen's heuristics appear in the new set of heuristics is because early stages of PROMETHEUS require the search and reuse of any usability heuristic related with the domain or the specific features of it. In this sense, Nielsen's heuristics are general and flexible enough to a variety of different domains. As a consequence, the new heuristics VH11 up to VH18 were developed to overcome the perceived shortcomings of Nielsen's general criteria, but there was no need to reinvent the general heuristics that are still suitable for virtual learning environments.

3.2. Stage 2: Heuristic Evaluation

For the heuristic evaluation of the ESPOCH virtual learning environment we recruited 70 students from the *Interfaces and Multimedia* course and randomly split them into 14 groups of 5 evaluators per group. Given that PROMETHEUS requires the application of least 1 heuristic evaluation, we designed our experimental setting with a large number of evaluators in order to control the variability in their application of both H_D and H_C. More specifically, our goal was for all groups to be as homogeneous as possible, which is why we: (i) recruited a large number of evaluators, and (ii) performed a random split between them. As all evaluators have similar profiles and experience levels, *i.e.* all of them are students at the same college and course, we believe that (i) and (ii) help us reduce the risk of having non-homogeneous groups.

A summary of the raw results is shown in Table 4, showing the quantity of problems found by each group of evaluators, as well as the average criticality of those problems. At a first glance it can be seen that groups using H_D found more problems than those using H_C. However, the statistical analyses performed in the following subsections will give definite evidence of whether this is supported by the data.

	Using H_D		Using H_C	
	Number of Problems	Average Criticality	Number of Problems	Average Criticality
Group 1	25	5.54	13	5.69
Group 2	30	5.64	14	4.72
Group 3	21	6.39	18	5.47
Group 4	21	5.52	13	5.85
Group 5	24	5.90	9	4.81
Group 6	26	4.43	11	6.24
Group 7	25	5.19	15	4.16
Total	172	-	93	-

Table 4: Summary of problems found by evaluator groups in the heuristic evaluation

Due to space limitations we are not able list all problems found by the evaluator groups. Nevertheless, these listings, for both VLEs and Nielsen's heuristics are available online.[3]

3.3. Stage 3 (a): Intra-Heuristics Analysis

The goal of the intra-heuristics analysis is to determine whether there is a significant effect on the heuristic evaluation when using H_D, in contrast to using H_C. Given the large number of randomly assigned evaluators, 35 per set of heuristics, we test several hypotheses regarding:

- total quantity of problems
- quantity of common problems

[3] http://zeus.inf.ucv.cl/~ifigueroa/doku.php/research/prometheus

- quantity of unique problems
- average severity and criticality of problems
- problem dispersion over the heuristics

Our results were computed based on two statistical tests: Students' t-test, and Wilcoxon's rank-sum test. We first describe the results regarding quantity of problems, severity, criticality and dispersion of problems over heuristics. Then, we provide a detailed explanation of the statistical tests applied.

3.3.1. Problem Quantity Analysis

We now describe the analysis of total problems ($\mathbb{TP}$), common problems ($\mathbb{CP}$) and unique problems ($\mathbb{UP}$), considering all the problems found by the 7 groups that worked in each set of heuristics. To visualize the obtained results, Figure 1 depicts the problem distribution for both VLE heuristics and Nielsen's heuristics. Here we can see that regarding $\mathbb{TP}$ and $\mathbb{CP}$, the sum of VLE groups outperforms the Nielsen groups, with 172 vs 93 and 147 vs 68 problems respectively. On the other hand, in both sets of heuristics the groups found the same quantity of 25 unique problems. Now we need to establish whether there are significant differences between the sets of heuristics. To this end, we consider the means $\mu^P_{H_D}$ and $\mu^P_{H_C}$, regarding the mean number of problems P for the domain and control heuristics—either $\mathbb{TP}$, $\mathbb{CP}$, or $\mathbb{UP}$—to determine whether:

- the null hypothesis $H^P_0 : \hat{\mu}^P_{H_C} = \hat{\mu}^P_{H_D}$ holds, meaning that both H_D and H_C are similarly effective at detecting problems, as the mean results of each are statistically equivalent.
- or, the null hypothesis does not hold, hence the alternative hypothesis $H^P_1 : \hat{\mu}^P_{H_C} \neq \hat{\mu}^P_{H_D}$ holds, meaning that both sets of heuristics are not similarly effective, because the mean results are statistically different.

Differences in Total Problems* $\mathbb{TP}$*. for both the t and Wilcoxon tests, the data allows us to reject the null hypothesis $H^{\mathbb{TP}}_0$ and accept the alternative hypothesis $H^{\mathbb{TP}}_1$. Put more simply, the data shows that *there exists a significant difference between VLE and Nielsen's heuristics when comparing for the mean quantity of total problems.*

Differences in Common Problems* $\mathbb{CP}$*. similar to the previous case, both statistical tests allow us to reject the null hypothesis $H^{\mathbb{CP}}_0$ and accept the alternative hypothesis $H^{\mathbb{CP}}_1$: *there are significant differences between VLE and Nielsen's heuristics when comparing for the mean quantity of common problems.*

Differences in Unique Problems* $\mathbb{UP}$*. In contrast to the previous cases, we cannot reject the null hypothesis $H^{\mathbb{UP}}_0$ because *the data shows no significant differences* between VLE and Nielsen's heuristics.

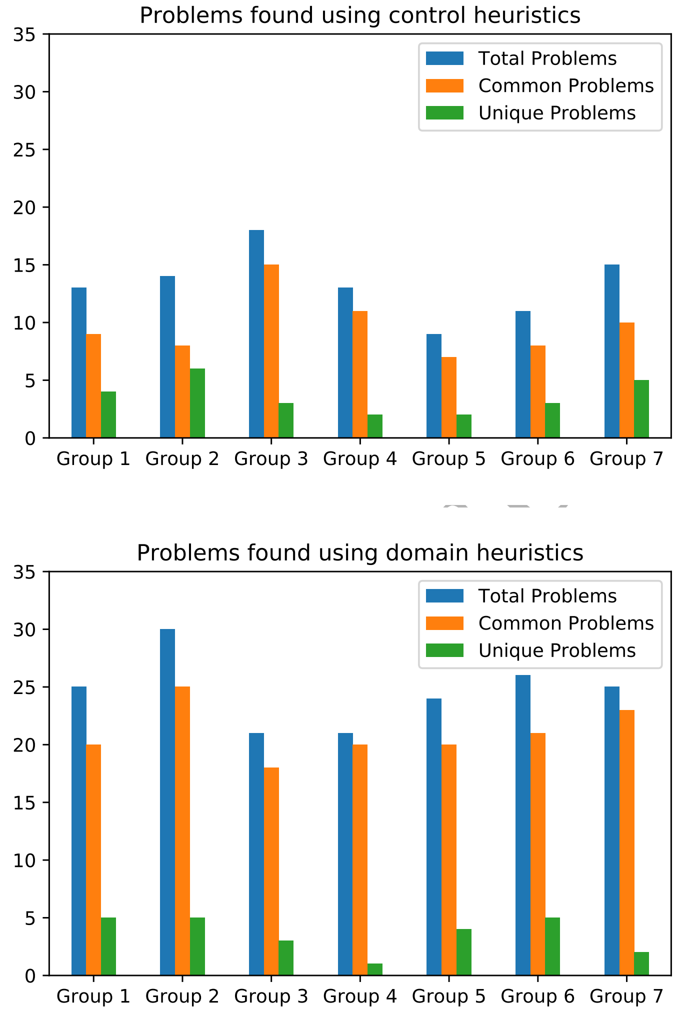

Figure 1: Quantity of problems for domain and control heuristics, regarding total, common and unique problems.

3.3.2. Severity and Criticality Analysis

Following the same approach as for the quantity of problems, we also study the mean values reported regarding the severity and criticality of the problems found by the evaluators. We also perform hypothesis testing, considering a null hypothesis H_0 where the means are equal, and an alternative hypothesis H_1 where the means are not equal. However, the evidence does not support the existence of significant differences regarding severity and criticality assessments, between the VLE and Nielsen's heuristic.

3.3.3. Dispersion of Problems by Heuristic

Understanding how problems are distributed per heuristic is another dimension to understand whether VLEs are more precise than Nielsens' heuristics. The goal is to avoid "god-heuristics" that contain most of the problems found during the heuristic evaluation. Hence, the standard deviation is a useful descriptive statistic to perform this comparison.

Dispersion data for control heuristics H_C		
ID	**AVG Problems by all groups**	**Standard deviation**
NH1	0.43	0.79
NH2	0.89	1.21
NH3	0.71	0.95
NH4	2.29	2.14
NH5	0.71	0.76
NH6	1.57	1.13
NH7	1.71	1.25
NH8	3.57	1.81
NH9	0.29	0.76
NH10	1.14	1.07
H_C **group avg**	**13.29**	**1.19**
Dispersion data for domain heuristics H_D		
ID	**AVG Problems by all groups**	**Standard deviation**
VH1	0.71	0.76
VH2	0.71	0.76
VH3	0.86	0.90
VH4	0.57	0.79
VH5	1.29	1.38
VH6	1	0.82
VH7	6.71	1.38
VH8	0.43	0.53
VH9	1.29	0.76
VH10	0.43	0.53
VH11	3.57	1.72
VH12	1.57	0.79
VH13	0.71	0.49
VH14	1.29	0.95
VH15	1	0.58
VH16	1.29	0.95
VH17	0.86	0.38
VH18	0.29	0.49
H_D **group avg**	**24.57**	**0.83**

Table 5: Dispersion data of problems by heuristics where H_C has higher average dispersion than H_D.

Table 5 summarizes the average number of problems found by each of the heuristics in the H_C and H_D. The mean standard deviation for H_C is 1.19, quite higher than the 0.83 value for H_D. This suggests that domain heuristics are more stable, given that evaluators are able to assign problems more evenly to the heuristics in the set of heuristics used in the evaluation.

It is interesting to remark that the standard deviation for heuristics NH1 to NH10 is higher than the equivalent counterpart in VLE, heuristics VH1 to VH10. This reflects the fact that some problems were transferred to the more specific domain heuristics. The only exceptions are for heuristics NH5/VH5 and NH7/VH7, where the variation rises in the domain heuristics. It is likely that this situation happens because evaluators using domain heuristics detected, on average, more problems than with control heuristics. In addition, those problems are probably more specific to the context of virtual learning environments. Finally, given that the number of heuristics is different in the control and domain groups, we did not perform any hypothesis testing, hence we use the descriptive statistics for interpretation.

3.3.4. Description of Statistical Analysis

The previous results rely on the application of the t-test and Wilcoxon's rank-sum test. Here we detail the theoretical foundation for the analysis and our results. First, let us consider Table 6, which summarizes the data and results for both statistical tests. Now, let us consider the first test, which is based on the distribution shown in Equation (5), and Student's t statistic.

$$\frac{\overline{X}_D - \overline{X}_C}{\sqrt{\left[\frac{(n_D-1)S_D^2+(n_C-1)S_C^2}{n_D+n_C-2}\right]\cdot\left[\frac{n_D+n_C}{n_D\cdot n_C}\right]}} \sim t_{n_D} + t_{n_C} - 2 \tag{5}$$

Here, X_D and X_C are random variables denoting the behavior of the domain and control heuristics, respectively, in the given metric X (quantity of problems, etc.). The hypothesis is that the behavior of X_D and X_C is similar to that of the t-statistic, based on the available n observations, and their standard deviations S^2. Again, subindices D and C denote association to the domain and control heuristics, respectively.

However, the results of the t-test are not conclusive because they rely on the assumption of normal distribution of the parameters. Therefore in order to strengthen the validity of our study we also applied the non-parametric Wilcoxon rank-sum test [40]—also known as Mann-Whitney U test—with a two-tailed significance level of $p < 0.05$, without having the assumption of normality. This design is sound because, as described in [16], when normality holds, the Mann–Whitney U test has an (asymptotic) efficiency of $3/\pi$ or about 0.95 when compared to the t-test. On the other hand, for distributions far from normal, the Mann–Whitney U test is considerably more efficient than the t-test.

We performed the t-test with a confidence of 95%, obtaining as a result that in the case of total and common problems the null hypothesis was rejected, since the t-value was greater than the t-critical value. In the case of the Wilcoxon rank-sum test, we also rejected the null hypothesis for total and common problems, since the two-tailed probability yielded lower results than the theoretical 0.05 value. Given that both tests agree in their result, that is, in the parametric and non-parametric tests the null hypothesis are rejected (or not), we can assert that the results obtained by our approach are statistically significant.

Heuristic Evaluation Group	Quantity of Problems Found						Averages Found			
	$\mathbb{TP}$		$\mathbb{CP}$		$\mathbb{UP}$		Severity		Criticality	
	H_D	H_C	H_D	H_C	H_D	H_C	H_D	H_C	H_D	H_C
Group 1	25	13	20	9	5	4	2.57	2.11	5.54	5.68
Group 2	30	14	25	8	5	6	2.77	2.36	5.64	4.72
Group 3	21	18	18	15	3	3	2.81	2.19	6.39	5.47
Group 4	21	13	20	7	4	2	2.75	2.90	5.52	5.85
Group 5	24	9	20	7	4	2	2.93	2.55	5.90	4.81
Group 6	26	11	21	8	5	3	2.16	3.01	4.43	6.24
Group 7	25	15	23	10	2	5	2.55	2.12	5.19	4.16
Descriptive values										
Mean $\overline{X}$	2.57	2.11	2.57	2.11	2.57	2.11	2.65	2.46	5.52	5.28
Variance S^2	2.57	2.11	2.57	2.11	2.57	2.11	0.06	0.14	0.37	0.54
Sample size n	7	7	7	7	7	7	7	7	7	7
t **TEST RESULTS**										
Pearson coefficient	8.93		6.29		2.45		0.10		0.45	
Degrees of freedom	12		12		12		12		12	
t **value**	7.07		8.42		0.00		1.09		0.67	
t **critical**	2.18		2.18		2.18		2.18		2.18	
Significant differences?	**yes**		**yes**		no		no		no	
WILCOXON TEST RESULTS										
***Hodges-Lehmann* means difference**	-11.75		-12		0		-0.3875		-0.385	
Confidence interval 95%	-15.5000 to -6.500		-15.0000 to -7.000		-2.0000 to 2.0000		-0.5250 to 0.2350		-1.0300 to 0.9750	
+ **differences**	0		0		3		2		3	
- differences	7		7		3		5		4	
Smallest ranges	0		0		10		8		10	
Bilateral probability	0.016		0.016		1.000		0.375		0.578	
Significant differences?	**yes**		**yes**		no		no		no	

Table 6: Results of the statistical comparison between domain and control heuristics. We first compute Student's t statistic under standard assumptions on normality. Then we apply Wilcoxon's test to verify our results under no assumptions of normality. The results of both statistical test coincide, concluding that there are statistical differences between H_D and H_C regarding the total amount of problems found (TP), and the total amount of common problems found (CP). We found no significant difference regarding unique problems (UP), severity and criticality of problems found.

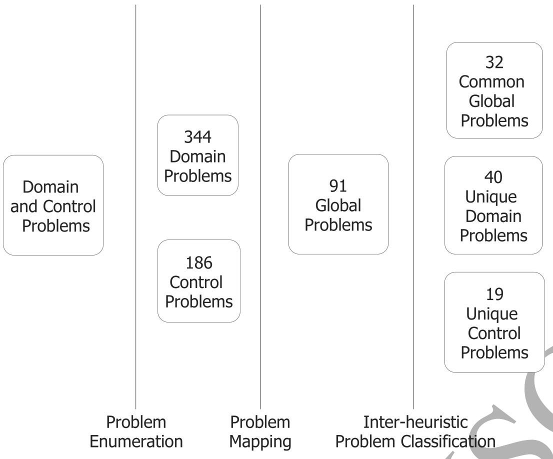
Domain and Control Problems
344 Domain Problems
186 Control Problems
91 Global Problems
32 Common Global Problems
40 Unique Domain Problems
19 Unique Control Problems
Problem Enumeration
Problem Mapping
Inter-heuristic Problem Classification

Figure 2: Application of inter-heuristic analysis procedure, steps 1 to 3

3.4. Stage 3(b): Inter-Heuristics Analysis

The second part of the validation of PROMETHEUS consists in comparing both sets of heuristics, VLEs and Nielsen's, to then compute the quality indicators proposed in PROMETHEUS. This complements the previous intra-heuristic analysis because we are not only interested in the quantity of problems, but we also want to know more about the actual problems found: did both sets of heuristic find exactly the same problems? were there several common problems? what were the problems found uniquely in one set of heuristics? are those problems more or less severe? The procedure for the inter-heuristic analysis is as follows:

1. **Problem Enumeration:** All the problems found in H_D and in H_C are uniquely enumerated, *e.g.P7V13* is problem number 13, found by group 7, in the context of VLE heuristics.

2. **Problem Mapping:** a group of researchers take all the problems, for both H_D and H_C, and creates new *global problems*. For instance, global problem GP1: "There is no way to check whether uploads succeed in the platform". The objective is to create a mapping between heuristic-specific problems and global problems.

3. **Inter-heuristic Problem Classification**: the mapping of global problems is used to identify and compute the following: total inter-heuristic problems ($\mathbb{ITP}$), common inter-heuristic problems ($\mathbb{ICP}$), unique inter-heuristic problems ($\mathbb{IUP}$). The latter is further divided into unique domain problems ($\mathbb{IUP}_D$), and unique control problems ($\mathbb{IUP}_C$).

4. **Construction of Quality Indicators**: finally, we compute all the quality indicators of PROMETHEUS previously detailed in Section 2.2.2, and proceed to interpret the results.

	Value	H_D **better than** H_C**?**
Unique problems ratio	$\Phi_P = 2.105$	yes
Dispersion ratio	$\delta_P = 1.037$	yes, slightly
Severity ratio	$\lambda_P = 1.049$	yes, slightly

Table 7: Summary of quality indicators for inter-heuristic analysis

The diagram in Figure 2 shows the results of applying steps 1 to 3 of this procedure to our case study. The final outcome is the following:

- Total inter-heuristic problems $\mathbb{ITP}$: 91
- Common inter-heuristic problems $\mathbb{ICP}$: 32
- Unique inter-heuristic problems $\mathbb{IUP}$: 59
    1. Unique domain problems $\mathbb{IUP}_D$: 40
    2. Unique control problems $\mathbb{IUP}_C$: 19

3.4.1. Inter-heuristic Quality Indicators

In the following we compute the quality indicators of PROMETHEUS, regarding the unique problems ratio, the problem dispersion ratio, and the criticality and severity ratios. Table 7 summarizes the results and the interpretation of the values.

Unique Problems Ratio Φ_P***.*** By considering the ratio between $\mathbb{IUP}_D$ and $\mathbb{IUP}_C$, we get:

$$\Phi_P = \frac{\mathbb{IUP}_D}{\mathbb{IUP}_C} = \frac{40}{19} = 2.105 \tag{6}$$

Given that $\Phi_P > 1$ we conclude that indeed the domain heuristics H_D perform better than the control heuristics H_C regarding the quantity of unique problems found. In other words, by using H_D the evaluators found around twice the amount of non-common problems than by using H_C. In fact, we found that the average *severity* of H_D (2.47) is slightly superior to the average *severity* of H_C (2.45). However, this fact seems to support our conclusion about the better performance of H_D.

Dispersion ratio δ_P***.*** Computing the dispersion rate is similar to the previous quality indicators. We take into account δ_C and δ_D, the dispersion values for both the control and domain heuristics respectively, and compute δ_P:

$$\delta_P = \frac{\delta_C}{\delta_D} = \frac{3.20}{3.09} = 1.037 \tag{7}$$

Again we get a value very close to 1, hence we consider that H_D has a slight advantage over H_C, regarding the distribution of problems over the heuristics of each set. This is also in line with the previous intra-heuristics analysis.

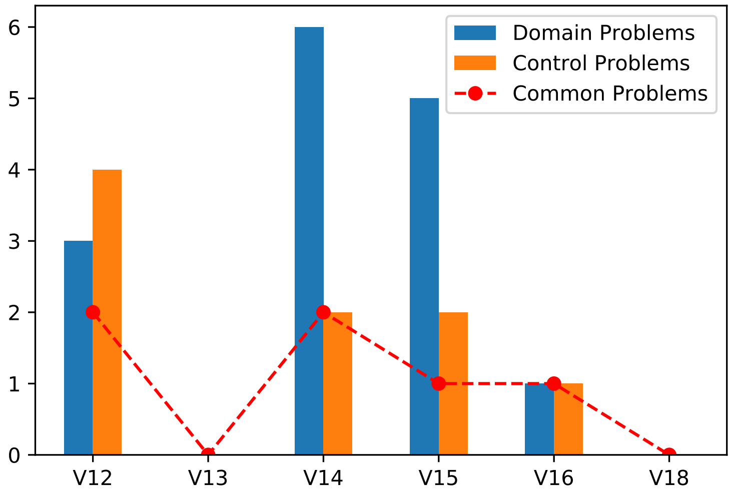

Figure 3: Distribution of problems in the new domain heuristics in H_D

Severity ratio λ_P. from the inter-heuristic problem classification we can compute the severity values λ_D and λ_C for both domain and control heuristics respectively. Then we compute the severity ratio λ_P, given by:

$$\lambda_P = \frac{\lambda_D}{\lambda_C} = \frac{2.58}{2.56} = 1.049 \tag{8}$$

Given that λ_P is very close to 1, it is difficult to say whether H_D perform better than H_C regarding the severity of found problems. Overall, and considering also de intra-heuristic analysis (Section 3.3.2) we consider that both sets of heuristics behave similarly, perhaps with a small advantage due to the use of H_D.

3.4.2. Distribution of Common Problems

Another interesting analysis regarding the dispersion and distribution of problems is to determine how the common inter-heuristics problems $\mathbb{ICP}$ are classified in both H_D and H_C, and conversely, whether the unique problems in H_D are mapped to the actual domain heuristics develop. This can provide new insights on whether and how problems that were classified in a control heuristic are transferred into domain heuristics, and into the pertinence of newly-developed domain heuristics. Put more simply, we want to assess whether domain heuristics actually promote a more fine-grained, domain-specific classification of problems. When a problem of the 91 problems belonging to the

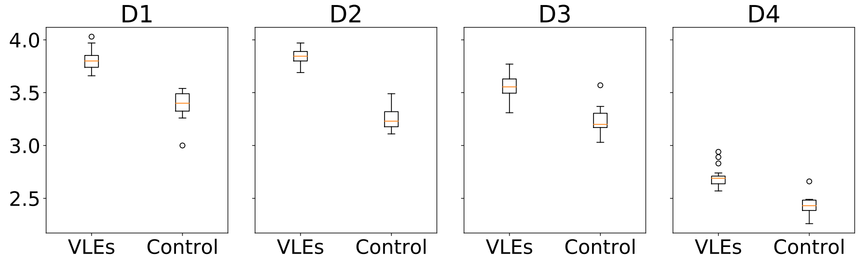

Figure 4: Descriptive comparison of post-experimental questionnaire results. In most cases the perception of VLE heuristics was better than for the control heuristics.

global list of problems was identified in both H_C and H_D, then that problem was considered as common problem.

Consequently, we consider only the novel domain-oriented heuristics VH12, VH13, VH14, VH15, VH16, and VH18 since they are not mapped to any of the control heuristics, as shown in Section 3. As shown in Figure 3, these *domain-original* heuristics vary in the distribution of common problems: VH12 and VH14 have 2 problems from $\mathbb{ICP}$, VH15 and VH16 have 1 problem from $\mathbb{ICP}$, and the rest feature 0 common problem. However in percentual terms, we see the following:

- Positive situations:
  - Only 33% of problems in VH14 are common problems.
  - Only 20% of problems in VH15 are common problems.
- Problematic situations:
  - 66% of problems in VH12 are common problems.
  - 100% of problems in VH16 are common problems.

The problematic situations help us to interpret the performance of H_D, specially given that δ_P is very close to 1. The percentages suggest that heuristics VH12 and VH16 could have been misunderstood, or that they where too general.

3.5. Stage 3(c): Post-Experimental Validation

As a complementary evaluation step, and also following previous work such as [14] and [11], we applied a questionnaire to all 70 evaluators that participated in the heuristic evaluation of the ESPOCH system. The goal is to capture and assess the perception of the evaluators regarding both sets of heuristics H_C and H_D. In particular, we consider four dimensions: Utility (D1), Clarity (D2), Ease of Use (D3), and Need of Additional Elements (D4). Each dimension is classified on a 5-points Likert scale where higher is better. Table 8 shows the values for each heuristic and each dimension, for both the domain heuristics H_D

and the control heuristics H_C. Table 8 shows the summarized results for both sets of evaluators and for each dimension under study. This data is further complemented by Figure 4, where the descriptive statistics for each set of evaluators can be easily compared.

H_D					H_C				
ID	**D1**	**D2**	**D3**	**D4**	**ID**	**D1**	**D2**	**D3**	**D4**
VH1	3.74	3.89	3.54	2.94	**NH1**	3.51	3.49	3.17	2.49
VH2	3.80	3.97	3.54	2.69	**NH2**	3.43	3.49	3.17	2.49
VH3	3.74	3.80	3.60	2.69	**NH3**	3.37	3.34	3.31	2.43
VH4	3.74	3.77	3.40	2.71	**NH4**	3.26	3.17	3.23	2.43
VH5	3.77	3.86	3.77	2.89	**NH5**	3.37	3.11	3.09	2.26
VH6	3.80	3.89	3.66	2.71	**NH6**	3.43	3.26	3.17	2.66
VH7	3.69	3.71	3.46	2.71	**NH7**	3.54	3.17	3.37	2.31
VH8	3.66	3.80	3.63	2.57	**NH8**	3.51	3.20	3.57	2.46
VH9	3.74	3.69	3.57	2.83	**NH9**	3.31	3.26	3.03	2.37
VH10	3.86	3.71	3.71	2.71	**NH10**	3.00	3.20	3.29	2.43
VH11	3.80	3.97	3.77	2.69					
VH12	3.77	3.83	3.63	2.57					
VH13	3.89	3.80	3.51	2.66					
VH14	3.80	3.80	3.31	2.66					
VH15	3.97	3.89	3.57	2.63					
VH16	3.83	3.91	3.49	2.57					
VH17	3.94	3.86	3.54	2.74					
VH18	4.03	3.91	3.37	2.63					
Mean	**3.81**	**3.84**	**3.56**	**2.69**	**Mean**	**3.37**	**3.25**	**3.25**	**2.44**
Std. dev.	**0.10**	**0.08**	**0.13**	**0.10**	**Std. dev.**	**0.16**	**0.13**	**0.15**	**0.11**

Table 8: Post-evaluation questionnaire: evaluators perceive H_D as better than H_C regarding Utility (D1), Clarity (D2), Ease of Use (D3), and Need of Additional Elements (D4).

Overall, the boxplots in Figure 4 show that evaluators using VLEs heuristics have a far better perception about these heuristics than in the case of Nielsen's heuristics. Indeed, in all dimensions—except D3—the minimum value in the VLE group is greater than the maximum value for the control group. For D3, there is a small overlap, but anyways the maximum value in the control group is below the average perception of VLEs' evaluators. Furthermore, by considering the standard deviations shown in Table 8 we see higher agreement between evaluators using VLE heuristics, *i.e.* lower standard deviation, in contrast to lower agreement between evaluators using Nielsen's heuristics.

Regarding VLEs heuristics, we can observe that D1 and D2 are the best evaluated dimensions, meaning that the heuristics are perceived as useful and with enough clarity. However D3 sits strictly below the evaluation of D2, suggesting that despite being useful and clear, the heuristics are not so easy to use as evaluators would have desired. This is reinforced by the low scores obtained in D4. Finally, even though perceptions are better regarding VLE heuristics, on average no dimension is ranked with a score of 4 or higher, which could be considered as a good outcome. This suggests there is room for improvements on the detailed explanations of each heuristic, as well as the need for complementary materials that help evaluators during their inspection.

4. Related Work

Our work is directly related to two research areas: the development of novel methodologies for constructing domain-specific heuristics, and the evaluation of usability in the context of virtual learning environments. Given that in Section 2 we already identified and compared the most relevant articles regarding the former, we now present an overview for the second area.

Usability and User Acceptance in VLEs. The perceived *ease of use* is recognized as a strong predictor of user acceptance of virtual learning environments [36, 17]. Following this line of inquiry, Parizotto-Ribeiro and Hammond [27] detected a positive relationship between interface aesthetics and perceived usability, based on a previous study in which authors stated the relevance of five design principles (unity, balance, proportion, homogeneity and rhythm) for achieving usability of VLEs' interfaces [26]. In the recent work of Wang [38], the conclusions of Parizotto-Ribeiro and Hammond [26] were, indeed, confirmed in the sense that *ease of use* and *perceived usefulness* are predictor factors of user's acceptance. Going beyond the standard methods of usability, recent work by Villareal-Freire et al. [37] propose a development guide for creating interfaces for virtual learning environments that also take into account the emotional context and responses of the students. There is a long tradition of research on usability evaluation of VLEs. For instance, Ihamäki and Vilpola [9] are early advocates for the adoption of usability methods and techniques in the context of e-learning. We refer the reader to [20], which is a recent systematic mapping of usability evaluation on virtual learning environments.

Heuristic Evaluation of VLEs. In [20] there is a considerable amount of studies that apply heuristic evaluation to VLEs, and in most cases there are domain-specific adaptations required for the successful application to the domain, such as in [25, 6, 30, 18]. In this context, the work of Reeves et al. [30] extends Nielsen's heuristics to define a novel set of 15 heuristics for the domain of e-learning programs. More recently, Mtebe and Kissaka [18] proposes a set of heuristics that consolidate interface usability, didactic effectiveness and motivation to learn, while considering the features of virtual learning environments. However, the works of [30] and [18] do not explicitly present the processes involved in the construction of the novel sets of heuristics.

Comparison of Existing Heuristics for VLEs. We perform a small comparison between the heuristics found in [30], [18], and our own. This is not meant as a full-blown comparison, which is beyond the original scope and goals of this paper, but as an exercise to explore how our developments compare with part of the state-of-the-art.

Reeves et al. [30]		Mtebe and Kissaka [18]		This work	
ID	**Definition**	**ID**	**Definition**	**ID**	**Definition**
RH1	Visibilty of system status	**MH1**	Visibility of system status	**VH1**	Visibility of system status
RH2	Match between system and the real world	**MH2**	Match between system and the real world	**VH2**	Match between system and the real world
RH3	Error recovery and exiting	**MH3**	User control and freedom	**VH3**	User control and freedom
RH4	Consistency and standards	**MH4**	Consistency and standards	**VH4**	Consistency and standards
RH5	Error prevention	**MH5**	Error prevention	**VH5**	Error prevention
RH6	Navigation support	**MH6**	Recognition rather than recall	**VH6**	Recognition rather than recall
RH7	Aesthetics	**MH7**	Flexibility and efficiency of use	**VH7**	Flexibility and efficiency of use
RH8	Help and documentation	**MH8**	Authenticity and Minimalism in Design	**VH8**	Aesthetic and minimalist design
RH9	Interactivity	**MH9**	Recognition, Diagnosis, and Recovery from Errors	**VH9**	Help users recognize, diagnose, from errors and recover from errors
RH10	Help and documentation	**MH10**	Help and documentation	**VH10**	Help and documentation
RH11	Learning Design	**MH11**	Instructional Material	**VH11**	System elements consistency
RH12	Media Integration	**MH12**	Collaborative Learning	**VH12**	Web standards and symbols
RH13	Instructional Assessment	**MH13**	Learner Control	**VH13**	Teaching-Learning process indicator
RH14	Resources	**MH14**	Feedback and Assessment	**VH14**	Flexible configuration of resources and learning objects
RH15	Feedback	**MH 15**	Accessibility	**VH15**	Storage capability
-	-	**MH16**	Motivation to Learn	**VH16**	Interactive communication
-	-	**-**	-	**VH17**	Multiple devices adaptation
-	-	**-**	-	**VH18**	Measuring learning

Table 9: Listing of 3 sets of heuristics for virtual learning environments

The comparison shown in Table 9 shows many similarities between the selected sets of heuristics. First, all of them include, either directly or with small adaptations, the whole set of Nielsen's heuristics. As explained in Section 3.1, this is due to the generality of these heuristics. Regarding the heuristics more specific to the domain of virtual learning environments, we can see that the three sets of heuristics identify crucial elements such as: tracking the learning progress, the ability to properly receive feedback, and the communication capabilities of the platforms. We believe this comparison suggests that the VLE heuristics produced using PROMETHEUS are not evidently better, nor worse, than those from the state-of-the-art. This is arguably a good thing, however we must remark that it is not a goal of this work to produce the "best" set of heuristics for virtual learning environments. Rather, we wanted to see whether the application of PROMETHEUS yielded a reasonable set of heuristics.

5. Conclusions

In this work we applied the PROMETHEUS methodology to develop the VLE heuristics for virtual learning environments. This development serves two main purposes. The first and most direct is the set of VLE heuristics itself, which was shown to perform better than Nielsen's heuristics, when applied to a specific system. The second one is related to the empirical validation of PROMETHEUS, as this is its first application outside of pilot studies. We believe that the rigorous statistical analyses performed in this work can be seen as a minimum standard to apply in the development of usability heuristics—in contrast to the arguably lax validations existing in the state of the art. We have also found new dimensions of interest regarding the validation of domain heuristics, such as the analysis on how problems are assigned to the novel domain heuristics. Indeed, it is quite fundamental to determine whether the novel domain heuristics contain the most significant problems found in heuristic evaluations, otherwise we could wonder whether the creation of domain heuristics makes sense.

Threats to validity. In the light of its contributions, this work presents some limitations and threats to validity related to the experimental design. A first consideration is the participation of students as evaluators in the case study. Although this is not uncommon, *e.g.* [11], it is valid to question whether these evaluators can be considered as "experts". This situation is mitigated by the fact that the evaluators were also regular users of the ESPOCH platform. Another consideration regarding the evaluators is the risk of non-homogeneous groups in the application of the heuristic evaluation. This is mitigated by two measures: first, we assume that all evaluators share a common profile, because they are all college students taking the same course. And second, all groups were created randomly. Finally, another threat is the application of Nielsen's heuristics as the set of control heuristics. Given the known existence of some sets of heuristics for virtual learning environments, it was quite likely that VLE heuristics would perform better than Nielsen's. However, this was a deliberate choice because we wanted to avoid as many confounding factors as possible. Hence we discarded existing sets of heuristics, considering also that most of them were developed by ad-hoc processes.

Future Work. Given that we only studied the results of the application of PROMETHEUS—as an indirect way to measure its effectiveness—it becomes necessary to also set the application of the methodology and its steps as the actual object of study. We plan to perform a future comparison between the application of PROMETHEUS, R3C, and QRR in the same domain. It is also interesting as future work to make an objective comparison of existing heuristics in the domain and try to determine the distinct tradeoffs between them. In addition, we plan to apply PROMETHEUS to several other domains such as: integrated learning environments, applications for music composition, among several others.

Conflicts of Interest

We declare the following conflicts of interest:

- Cristian Rusu, Pontificia Universidad Católica de Valparaíso: department colleague, and has previous work with author Cristhy Jiménez
- Cesar Collazos: previous work with author Cristhy Jiménez

Acknowledgments

We thank the anonymous reviewers for their comments, which have helped to greatly improve the quality of this paper. We also thank all of the participants that served as evaluators in the heuristic evaluation.

6. References

References

[1] Frances Bell. Connectivism: Its place in theory-informed research and innovation in technology-enabled learning. *The International Review of Research in Open and Distributed Learning*, 12(3):98–118, 2011.

[2] Jimenez C., H. Allende Cid, and I. Figueroa. Prometheus english version. *CoRR*, abs/1802.10121, 2015. URL http://arxiv.org/abs/1802.10121.

[3] Douglas B Clark, Emily E Tanner-Smith, and Stephen S Killingsworth. Digital games, design, and learning: A systematic review and meta-analysis. *Review of educational research*, 86(1):79–122, 2016.

[4] F. M. de Sales and M. A. S. Ramos. Technical and pedagogical usability in e-learnig: Perceptions of students from the federal institute of rio grande do norte (brazil) in virtual learning environment. In *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–4, June 2015. doi: 10.1109/CISTI.2015.7170447.

[5] Jaime Díaz, Cristian Rusu, J Antonio Pow-Sang, and Silvana Roncagliolo. A cultural-oriented usability heuristics proposal. In *Proceedings of the 2013 Chilean Conference on Human-Computer Interaction*, pages 82–87. ACM, 2013.

[6] Mary C Dyson and Silvio Barreto Campello. Evaluating virtual learning environments: What are we measuring?. *Electronic Journal of E-learning*, 1(1):11–20, 2003.

[7] H. Rex Hartson, Terence S. Andre, and Robert C. Williges. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 15(1):145–181, 2003. doi: 10.1207/S15327590IJHC1501_13.

[8] Setia Hermawati and Glyn Lawson. Establishing usability heuristics for heuristics evaluation in a specific domain: Is there a consensus? *Applied Ergonomics*, 56:34 – 51, 2016. ISSN 0003-6870. doi: http://dx.doi.org/10.1016/j.apergo.2015.11.016. URL http://www.sciencedirect.com/science/article/pii/S0003687015301162.

[9] Heli Ihamäki and Inka Vilpola. Usability of a virtual learning environment concerning safety at work. *Electronic Journal of e-Learning*, 2(1):103–112, 2004.

[10] R. Inostroza, C. Rusu, S. Roncagliolo, C. Jiménez, and V. Rusu. Usability heuristics validation through empirical evidences: A touchscreen-based mobile devices proposal. In *Chilean Computer Science Society (SCCC), 2012 31st International Conference of the*, pages 60–68, Nov 2012. doi: 10.1109/SCCC.2012.15.

[11] Rodolfo Inostroza, Cristian Rusu, Silvana Roncagliolo, Virginica Rusu, and César A. Collazos. Developing smash: A set of smartphone's usability heuristics. *Computer Standards and Interfaces*, 43:40 – 52, 2016. ISSN 0920-5489. doi: http://dx.doi.org/10.1016/j.csi.2015.08.007. URL http://www.sciencedirect.com/science/article/pii/S0920548915000926.

[12] W ISO. 9241-11. ergonomic requirements for office work with visual display terminals (vdts). *The international organization for standardization*, 45, 1998.

[13] C. Jimenez, H. Allende Cid, and I. Figueroa. Prometheus: Procedural methodology for developing heuristics of usability. *IEEE Latin America Transactions*, 15(3):541–549, March 2017. ISSN 1548-0992. doi: 10.1109/TLA.2017.7867606.

[14] Cristhy Jimenez, Cristian Rusu, Virginica Rusu, Silvana Roncagliolo, and Rodolfo Inostroza. Formal specification of usability heuristics: How convenient it is? In *Proceedings of the 2Nd International Workshop on Evidential Assessment of Software Technologies*, EAST '12, pages 55–60, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1509-8. doi: 10.1145/2372233.2372249. URL http://doi.acm.org/10.1145/2372233.2372249.

[15] Rita Kop. The challenges to connectivist learning on open online networks: Learning experiences during a massive open online course. *The International Review of Research in Open and Distributed Learning*, 12(3):19–38, 2011.

[16] E.L. Lehmann. *Elements of Large-Sample Theory*. Springer Verlag, 1999.

[17] Luis L Martins and Franz Willi Kellermanns. A model of business school students' acceptance of a web-based course management system. *Academy of Management Learning & Education*, 3(1):7–26, 2004.

[18] J. S. Mtebe and M. M. Kissaka. Heuristics for evaluating usability of learning management systems in africa. In *2015 IST-Africa Conference*, pages 1–13, May 2015. doi: 10.1109/ISTAFRICA.2015.7190521.

[19] Roberto Munoz and Virginia Chalegre. Defining Virtual Worlds Usability Heuristics. *2012 Ninth International Conference on Information Technology - New Generations*, pages 690–695, apr 2012. doi: 10.1109/ITNG.2012.138. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6209172.

[20] Walter Takashi Nakamura, Elaine Harada Teixeira de Oliveira, and Tayana Conte. Usability and user experience evaluation of learning management systems. 2017.

[21] Jackob Nielsen. 10 Heuristics for User Interface Design: Article by Jakob Nielsen, 1995. URL https://www.nngroup.com/articles/ten-usability-heuristics/.

[22] Jakob Nielsen. How to conduct a heuristic evaluation. *Nielsen Norman Group*, 1, 1995.

[23] Jakob Nielsen. Severity ratings for usability problems: Article by jakob nielsen. *Severity Ratings for Usability Problems: Article by Jakob Nielsen*, 1995.

[24] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6. doi: 10.1145/97243.97281. URL http://doi.acm.org/10.1145/97243.97281.

[25] Konstantina Orfanou, Nikolaos Tselios, and Christos Katsanos. Perceived usability evaluation of learning management systems: Empirical evaluation of the system usability scale. *The International Review of Research in Open and Distributed Learning*, 16(2), 2015.

[26] R Parizotto-Ribeiro and N Hammond. What is aesthetics anyway? investigating the use of the design principles. *Aesthetic Approaches to Human-Computer Interaction*, page 37, 2004.

[27] Rosamelia Parizotto-Ribeiro and Nick Hammond. Does aesthetics affect the users' perceptions of vles. In *12th International Conference on Artificial Intelligence in Education, Amsterdam, Denmark. Retrieved June*, volume 30, page 2006, 2005.

[28] Daniela Quiñones and Cristian Rusu. How to develop usability heuristics: A systematic literature review. *Computer Standards & Interfaces*, 53:89–122, 2017. doi: 10.1016/j.csi.2017.03.009. URL https://doi.org/10.1016/j.csi.2017.03.009.

[29] Daniela Quiñones, Cristian Rusu, and Virginica Rusu. A methodology to develop usability/user experience heuristics. *Computer Standards & Interfaces*, 59:109 – 129, 2018. doi: https://doi.org/10.1016/j.csi.2018.03.002. URL http://www.sciencedirect.com/science/article/pii/S0920548917303860.

[30] Thomas C Reeves, Lisa Benson, Dean Elliott, Michael Grant, Doug Holschuh, Beaumie Kim, Hyeonjin Kim, Erick Lauber, and Sebastian Loh. Usability and instructional design heuristics for e-learning evaluation. 2002.

[31] Cristian Rusu, Silvana Roncagliolo, Virginica Rusu, and Cesar Collazos. A methodology to establish usability heuristics. In *Proc. 4th International Conferences on Advances in Computer-Human Interactions (ACHI 2011), IARIA*, pages 59–62, 2011.

[32] Cristian Rusu, Silvana Roncagliolo, Gonzalo Tapia, Danae Hayvar, Virginica Rusu, and Dorian Gorgan. Usability heuristics for grid computing applications. In *Proceedings of The 4th International Conferences on Advances in Computer-Human Interactions ACHI*, 2011.

[33] George Siemens. Connectivism: A learning theory for the digital age. 2014.

[34] Akash Singh and Janet Wesson. Evaluation criteria for assessing the usability of ERP systems. *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, 125(October):87–95, 2009. doi: 10.1145/1632149.1632162. URL http://portal.acm.org/citation.cfm?id=1632149.1632162{&}coll=ACM{&}dl=ACM{&}CFID=65944897{&}CFTOKEN=53873172.

[35] Andrés Solano, Cristian Rusu, César A Collazos, and José Arciniegas. Evaluating interactive digital television applications through usability heuristics/evaluando aplicaciones de televisión digital interactiva a través de heurísticas de usabilidad. *Ingeniare: Revista Chilena de Ingenieria*, 21 (1):16, 2013.

[36] Erik M Van Raaij and Jeroen JL Schepers. The acceptance and use of a virtual learning environment in china. *Computers & Education*, 50(3): 838–852, 2008.

[37] Angela Villareal-Freire, Andrés F. Aguirre, and César A. Collazos. Emovle: An interface design guide. In Aaron Marcus and Wentao Wang, editors, *Design, User Experience, and Usability: Designing Pleasurable Experiences*, pages 142–161, Cham, 2017. Springer International Publishing. ISBN 978-3-319-58637-3.

[38] Minhong Wang. Effects of individual and social learning support on employees' acceptance of performance-oriented e-learning. In *E-Learning in the Workplace*, pages 141–159. Springer, 2018.

[39] Zhijun Wang, Terry Anderson, Li Chen, and Elena Barbera. Interaction pattern analysis in cmoocs based on the connectivist interaction and engagement framework. *British Journal of Educational Technology*, 48(2): 683–699, 2017.

[40] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, December 1945. ISSN 00994987. doi: 10.2307/3001968. URL http://dx.doi.org/10.2307/3001968.