Towards Meaning in User Interface Prototypes

Gretchen Torres De Macedo[1,2][a], Lucas Viana[1][b] and Bruno Gadelha[1][c]

[1]*Institute of Computing, Federal University of Amazonas - UFAM, Manaus, Amazonas, Brazil*

[2]*Amazonas State Prosecution Service - MPE/AM, Manaus, Amazonas, Brazil*

Keywords: User Interface Prototypes, Semantic Analysis, Heuristics.

Abstract: User interface prototypes aid several activities during the development process lifecycle. However, there are still many manual activities performed during this process. This research project investigates how semantic meaning identification in UI prototypes can help carry out manually performed tasks. We started by analyzing a set of UI prototypes obtained from prototyping activities with students. As a result, we created a database of 856 UI prototypes labeled semantically in 19 classes and two semantic levels: layout and functionalities. Each class of UI prototypes was analyzed to identify its distinguishing characteristics, from which we obtained a set of 19 heuristics specifications. This set of heuristics allows the development of solutions for automatic analysis of UI prototypes, thus supporting software prototyping activities.

1 INTRODUCTION

User interface prototyping is a commonly used resource to represent software features through user interface images during the software development process (Bjarnason et al., 2021). Analysts can use UI prototypes to communicate with customers, explore solutions, and present and validate ideas (Dow et al., 2011; Lauff et al., 2020). In projects that use the agile development methodology, prototyping has shown to be a critical requirements elicitation instrument that can be associated with design thinking techniques (Garcia et al., 2017; Käpyaho and Kauppinen, 2015).

However, prototyping is a challenging task (Lee et al., 2020). A designer must analyze the functionalities to represent during software prototyping and then choose design elements that characterize such functionalities. It is necessary, at the same time, to follow design rules and be creative and innovative to attract and maintain user engagement (Chen et al., 2020a). Therefore, there must be ways to assist the design of graphical user interfaces (GUI).

Some works deal with this challenge by automatically analyzing GUIs to find design problems. Examples include using computer vision techniques to check whether a GUI implementation is according to its design (Moran et al., 2018), or GUIComp (Lee et al., 2020), a tool to assist GUI design by suggesting similar GUIs, evaluating the arrangement of components, and showing attention areas on the prototype. However, GUI design checking approaches based on images do not consider the meaning of the elements and functionality represented in the prototype, which limits the types of checks made during the GUI evaluation to the arrangement of UI components on the screen.

Another way to aid the conceptualization of GUIs is through GUI searching (Huang et al., 2019), where designers can search a variety of examples of similar user interface designs (Bernal-Cárdenas et al., 2019; Chen et al., 2020b). However, GUI searching based on images only searches for GUIs visually similar, limiting the diversity of recovered examples.

Given the limitations of image-based UI design assistance methods, new approaches leverage semantic identification of interfaces to perform UI related tasks. A basis work, Liu et al. (2018) use a screenshot dataset and screen metadata to generate semantic annotations for mobile apps interfaces. Another work uses interaction data and transition components detection to predict the subsequent screens given a pair of screenshots (Zecheng et al., 2020). Chen et al. (2020a) use a combination of images and textual information to predict screen categories and suggest tags that describe UI designs.

Those examples show that semantic identification enables more detailed UI verifications and could

[a] https://orcid.org/0000-0003-1428-4924
[b] https://orcid.org/0000-0002-8154-0062
[c] https://orcid.org/0000-0001-7007-5209

De Macedo, G., Viana, L. and Gadelha, B.
Towards Meaning in User Interface Prototypes.
DOI: 10.5220/0011067100003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 235-246
ISBN: 978-989-758-569-2; ISSN: 2184-4992

therefore be used to provide better feedback during interface design by indicating design errors. An example is the analysis of recurrent features. Some features are common to different applications, such as user authentication, search, or registration forms. However, the knowledge regarding the descriptive characteristics of these recurring features is tacit: practitioners widely know it, but it is not formally specified. Therefore, an explicit description of each type of functionality can help prototyping activities by providing a means for identifying the functionality represented in a prototype and indicating possible inconsistencies between the prototype and the specification of the functionality. Furthermore, identifying semantic meaning in UI prototypes could improve the search for alternative designs for a feature (Liu et al., 2018).

Semantic identification can also support the implementation of functional prototypes. For example, developers can use functional prototypes (Bjarnason et al., 2021) to demonstrate functionalities more realistically and evolve the prototypes to become the first working version of the application, also known as a Minimum Viable Product (MVP) (Lenarduzzi and Taibi, 2016). However, this involves additional coding effort and, therefore, increased development cost. Hence, identifying the functionality represented in a prototype could also be used for suggesting code for implementation, retrieved from local or remote code repositories.

Considering these new applications of semantic identification to assist user interface design, this work seek to answer the following research question: *"How to perform semantic identification in user interface prototypes?"*. To answer this question, we addressed the problem of semantic identification through a heuristic-based approach. In this approach, we identified different categories of prototypes and described their key characteristics using heuristic specifications (Quiñones et al., 2018). Initially, to obtain a prototype base for analysis, we conducted prototyping activities with undergraduate students, resulting in 856 screens of 31 different projects from varied domains. From the iterative, open coding analysis (Felderer and Travassos, 2020) of UI prototypes, we labeled them in 19 categories, which fit in two different levels of abstraction[1]. Then, for each class, we wrote a descriptive heuristic specification. The set of heuristics and the labeled prototypes dataset can assist UI prototyping or integrate automated UI prototype analysis solutions.

This paper is organized as follows: In Section 2, we present works that identify semantics in UI prototypes or design images for different purposes; in Section 3, we describe the process of developing the design heuristics; in section 4, we present the heuristic specification of the classes identified during the analysis, with some examples; in Section 5, we present the evaluation of our set of heuristics using Expert Judgment; in Section 6, we discuss the results obtained and compare them with related works; finally, in Section 7, we present the conclusions of this work and future work.

[1]Research data available at https://doi.org/10.6084/m9.figshare.17087717

2 BACKGROUND

This section presents key concepts that form the basis of our research project: User Interface Prototypes and Heuristics. Also, we present related works that use different approaches to address the problem of UI analysis.

2.1 User Interface Prototypes

User Interface Prototyping is a practice used in the early stages of software development. Through prototypes, UI designers can demonstrate the software functionality by a visual representation of the software user interface (Deininger et al., 2017). Besides, designers can identify design problems and explore different software solutions for a problem.

The systematic mapping of Garcia et al. (2017) shows that prototypes are the most used artifact during different software development stages. They facilitate communication with customers in agile settings, serving later as a reference for development and a guide for usability tests. Käpyaho and Kauppinen (2015) also cite prototyping as a powerful artifact for requirements elicitation in agile settings, where there is usually little documentation since they are easier to create and update than written documentation.

Software interface prototyping activities can be aided by semantic identification. In this work, we approached the semantic identification problem by developing design heuristics to characterize prototype categories.

2.2 Heuristics

Heuristics are rules derived from experience used in the implementation of decision processes to find solutions to specific problems (Calle-Escobar et al., 2016). Unlike deterministic methods where all variables are involved in finding a perfect solution, heuristics ignore part of the information to find a good enough solution for solving a problem (Gigerenzer, 2008). That way, heuristics are used to reach a satisfactory solution with less computational effort (Shah and Oppenheimer, 2008).

Several works use heuristics to support products' design. This type of heuristics is known as Design Heuristics. For example, Figueiredo et al. (2012) present design heuristics to assess concern-sensitive designs, i.e., aspects of a software product that can affect the maintenance of software modules. Yilmaz et al. (2016) present a compilation of 77 design heuristics commonly used in product design.

In this work, we adopted a heuristic-based approach to develop design heuristics that can assist designers during UI prototypes development. Heuristics could focus on a UI category's most relevant and deterministic characteristics to check if it meets the specification.

2.3 Related Works

Several approaches have been used to support UI design tasks. In the field of user interface analysis, Moran et al. (2018) presents an approach to identify errors in mobile GUI implementations by comparing it to its design. Packevičius et al. (2018) leverage application code and screenshots to implement a source code analysis method based on sets of defects and rules to identify text presentation and semantic defects. Finally, GUIComp (Lee et al., 2020) is a GUI prototyping assistance tool that performs GUI checking and provides feedback by recommending similar user interfaces, evaluating visual complexity metrics, and showing points in the prototype that grab more user attention.

Other works also use the UI searching and recommendation approach to facilitate the conceptualization of user interfaces. Guigle (Bernal-Cárdenas et al., 2019) is a text-based UI search that leverages screen texts, UI components, app name, and colors to retrieve similar screens from Google Play apps. Swire (Huang et al., 2019) implements a sketch-based GUI retrieval from a sketch database built with UI examples from Rico dataset (Deka et al., 2017). Chen et al. (2020b) also perform GUI search and retrieval using a wireframe-based approach.

Several recent works adopt the semantic identification approach in user interface prototypes (Liu et al., 2018; Leiva et al., 2020; Zang et al., 2021; Chen et al., 2020a). In the work of Liu et al. (2018) , the authors describe a semantic annotation approach of elements present in images of mobile application interfaces, specifically textual buttons, and icons, in addition to some structural elements, such as menus and footers. This annotation can be used to leverage semantic based operations, such as GUI search or flow search.

ActionBert (Zecheng et al., 2020) is a methodology for identifying semantic meaning based on applications' structural information and interactive user actions' tracking. From a pair of consecutive screens, it can identify the linking component and predict the subsequent screen. The Enrico dataset (Leiva et al., 2020) is a labeled database of 1460 application screenshots, taken from the Rico (Deka et al., 2017) dataset, sorted into 20 topics. The authors also present several possible applications of Enrico to help designers' work during user interface design. Chen et al. (2020a) use semantic identification to discover missing tags used to describe images on design sharing sites. The presented solution combines application interface design images and metadata to suggest complementary tags and a visualization method to highlight which parts of images gave rise to the tag hint. The work of Li et al. (2021) presents a self-supervised technique that combines text information, design and layout patterns, application metadata, and interaction data to generate embeddings that can be used in various applications, from assistive technologies to design support of UI.

Unlike these related works, we use a method based on the thorough analysis of different types of prototypes to extract their main features. The resulting heuristics describe the different types of screens through their layout or functional characteristics. Our approach differs from others by going beyond the recognition of prototypes' categories, once the checklist specified in each heuristic allows us to verify whether the design meets the requirements described in the heuristic specification.

3 METHOD

In this work, we present a set of design heuristics to identify semantics in user interface prototypes. To develop the design heuristics, we first identified categories of prototypes from a base of student-developed UI prototypes and related work. Then, we identified the defining characteristics of each prototype category, which make up the heuristics specification.

To achieve this set of prototype categories and their predominant characteristics, we adapted the methodology proposed by Quinones et al. (2018) to the context of design heuristics specification. This methodology is composed of 8 steps, which can be performed sequentially or iteratively, in cycles involving one or more steps. Figure 1 shows how we performed the methodology steps. In the Exploratory and Experimental stages, we gathered elements for the Descriptive stage. Also, in the Correlational stage, we changed the prototypes categories obtained during the Experimental stage. In this section, we describe each of them and how we adapted it to the context of this work.

Step 1, the Exploratory stage, initiates by defining a specific application domain to develop usability and UX heuristics. In our case, we aim to assist UI design activities, so we developed characterization heuristics for UI prototypes from different domains and platforms. Therefore, our research domain consists of user interface prototype analysis in the context of user interface design. In this stage, we also must collect three types of data: information about the domain, usability/ UX attributes under evaluation by the set of new heuristics, and existing heuristics, principles, guidelines, and patterns. First, as presented in Section 2, we performed a literature review, where we found works related to GUI design assistance, semantic identification in UI prototypes, and GUI search. In those works, the UI design and prototypes classification is approached at different semantic levels: at the component level, by identification of UI components and interactive elements like buttons and icons (Liu et al., 2018); at the structural level, where the layout of its components characterizes the semantic of UI designs (Leiva et al., 2020); and at the functional level, describing more complex functionalities (Chen et al., 2020a).

In our context, we collected the following types of information from related works:

- Components identification heuristics, text buttons concepts, and icons classes from Liu et al. (2018);
- UI designs topics from Leiva et al. (2020);
- UI semantic categories (platform, color, app functionality, screen functionality, screen layout) from Chen et al. (2020a);
- User interface design patterns from UI Patterns website (Toxboe, 2022).

In Step 2, the Experimental stage, we performed an experiment to create a user interface prototypes dataset. We needed this dataset to analyze and obtain a categorization of UI prototypes. Thus, we performed prototyping activities with undergraduate students. These activities resulted in 31 projects and 856 UI prototypes for mobile, web, and desktop platforms.

Then, we analyzed the prototypes to obtain different categories of prototypes. Before starting the prototypes' analysis, one of the researchers removed duplicated prototypes and descriptive UI elements. Then, this researcher started labeling UI prototypes using an open coding approach, where the labels of UI prototypes were created and modified iteratively during the analysis. The researcher replaced some prototypes labels with more representative ones in this process. Next, the researcher reviewed the labeling performed in the first step, unifying distinct labels assigned to prototypes with the same semantic meaning and splitting labels for prototypes with different semantic meanings. This researcher also grouped very similar labels with few elements. After the review stage, another experienced researcher reviewed the categorization. The disagreement points were discussed until they reached a consensus.

The labeling process resulted in 23 prototypes categories, where 47 UI prototypes received more than one label. Table 1 presents the prototype categories obtained during the Labeling process. It is worth noticing that only the quantities of single-label prototypes are listed.

In step 3, the Descriptive stage, we must select and prioritize the topics collected in Steps 1 and 2. Therefore, we analyzed and selected heuristics and UI categories from related works. First, from Liu et al. (2018), we selected all concept buttons and synonyms definitions, as we will use them in our heuristics definition. However, their heuristics were not selected because they detect UI components and icons, and our work goal is to identify semantics for the entire UI prototype. Next, from the UI Patterns website (Toxboe, 2022), Leiva et al. (2020), and Chen et al. (2020a), we selected only the categories that matched the ones we found during the prototypes labeling process. From Chen et al. (2020a) we also selected the app categories to classify our prototype projects. We also adopted their screen functionality and layout UI categories to group our heuristics. Table 2 presents the information selected from each related work.

In step 4, the Correlational stage, the researchers must match features and existing heuristics collected in previous steps. We have not selected the pre-existing heuristics in step 3, so we mapped the prototype categories resulting from the labeling process with the categorization lists from related works. The Table 3 presents this mapping. Also, we grouped our categories into two groups: Layout and Functionalities. The Layout categories refer to generic software features, such as registration forms and lists of objects. The Functionalities categories are related to more complex features, such as authentication forms and chat. The Layout categories identification starts with the *HEU1* prefix and the Functionalities categories, with the *HEU2* prefix.

In Step 5, the Selection stage, we should determine which heuristics to keep, adapt, create and elim-

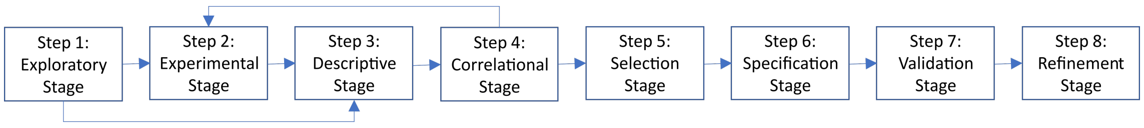

Step 1: Exploratory Stage
Step 2: Experimental Stage
Step 3: Descriptive Stage
Step 4: Correlational Stage
Step 5: Selection Stage
Step 6: Specification Stage
Step 7: Validation Stage
Step 8: Refinement Stage

Figure 1: Methodology to develop usability and user experience heuristics (Source: Quiñones et al. (2018)).

Table 1: Prototypes categogies obtained during the Experimental stage.

#	Label	Quantity	Description
01	Account Registration	57	Creating a user account to access an application features
02	Authentication	40	Credential information form for accessing application resources
03	Chat	24	Functionality for exchanging instant messages between users of the application
04	Checkout	17	Information on payment, shipping, and billing options in e-commerce applications
05	Content Filter	7	A set of fields that filter the displayed information
06	Date Picker	8	Query based on dates presented as a calendar
07	File Upload	6	A file upload form
08	Grid	20	A set of instances of an entity from the application domain displayed in a grid format
09	Home	11	Application home page presenting the most relevant information in a wide variety of formats.
10	List	123	Set of instances from a domain entity
11	Master-Detail	28	Displaying attributes of an instance of a domain entity followed by a list of instances of another entity related to the first
12	Menu	82	Application feature options
13	Modal	21	A pop-up window asking for a user action.
14	Notification	48	System status information
15	Product Page	20	Detailed information on a product
16	Registration	80	Form for registering an instance of an entity in the application domain
17	Search	50	Search engine
18	Shopping Cart	12	List of products or services and their quantities, to be purchased in an e-commerce application
19	Terms of Use	10	Application terms of use
20	Update	34	A form that presents the attribute values of an entity instance so that they can be modified
21	User profile	32	User account information
22	Videoconference	20	Videoconference functionality
23	View	59	Displays attribute values of an object from an entity in the application domain

Table 2: Information selected from different sources during step 3, Descriptive stage.

Source	Collected information	Selected information
Liu et al. (2018)	28 UX concept buttons and synonyms (no, login, back, go, ok, all, next, add, create, more, retry, skip, set, delete, Facebook, view, book, continue, list, save, search, finish, agree, show, share, buy, update, edit); Heuristics for: extracting text buttons, extracting icons, code-based heuristics to identify components	All concept buttons and synonyms were selected. None of the heuristics were selected.
Leiva et al. (2020)	Enrico dataset topics (bare, dialer, camera, chat, editor, form, gallery, list, login, maps, media player, menu, modal, news, other, profile, search, settings, terms, tutorial	Chat, form, list, login, menu, modal, profile, search, terms.
Chen et al. (2020a)	Semantic UI categories (platform, color, app functionality, screen functionality, screen layout)	All app categories; screen functionality categories: login, signup, checkout, search, profile; layout categories: form, list, grid.
UI Patterns website (Toxboe, 2022)	User interface design patterns	Modal, notifications, table filter, product page, shopping cart, chat, account registration.

Table 3: Match between categories from the labeling process and related works.

ID	Category	Enrico (Leiva et al., 2020)	Chen et al. (2020a)	UI Design Patterns (Toxboe, 2022)
Layout categories				
HEU1-01	Content Filter			Table Filter
HEU1-02	File Upload			
HEU1-03	Grid		Grid	
HEU1-04	List	List	List	
HEU1-05	Master-Detail			
HEU1-06	Menu	Menu		
HEU1-07	Registration	Form	Form	
HEU1-08	Update	Form	Form	
HEU1-09	View			
Functionalities categories				
HEU2-01	Account Registration		Signup	Account Registration
HEU2-02	Authentication	Login	Login	
HEU2-03	Chat	Chat		Chat
HEU2-04	Checkout		Checkout	
HEU2-05	Home			
HEU2-06	Product Page	Profile		Product Page
HEU2-07	Search	Search	Search	
HEU2-08	Shopping Cart			Shopping Cart
HEU2-09	User profile	Profile	Profile	
HEU2-10	Videoconference			

inate. We analyzed the prototype categories in this work to identify their characteristics that represent prototype design requirements. This analysis verified that four categories do not represent software features: Date Picker, Terms of Use, Notification, and Modal. Date Picker is a differentiated data entry component for dates selection used in various situations. Terms of Use have no specific design requirements. Finally, the Notification and Modal categories represent situations in which texts are shown to the user and may require user action. These categories can be used in a wide variety of situations and also do not represent software features. Therefore, we did not select these four categories for the specification stage.

Step 6 is the Specification stage, where the heuristics are formally specified. In this work, we specified heuristics for the 19 remaining categories from the labeling process. We adapted the heuristics definition standard template proposed by Quiñones et al. (2018) to the context of this work. As we are not dealing with usability heuristics, we removed the fields *Application Feature*, *Benefits*, *Usability Attributes*, and *UX Factors*.

Also, we added the fields *Synonyms* and *Additional Information*. The field *Additional Information* in the heuristics specification lists additional sets of information a heuristic needs in its checklist items. For example, some heuristics, such as Registration, Update, and List, are related to the application domain entities. Besides, the heuristics use terminological and visual characteristics, mapped as terms and icons sets, to describe the prototype elements. For instance, a set of terms used to label buttons on authentication forms is (login, connect, sign in). We present the heuristics specification in Section 4.

In Step 7, the Validation stage, we performed a validation of the proposed design heuristics through an expert judgment experiment (Quiñones et al., 2018). We proposed an app development scenario where the participants designed UI prototypes for the software requirements. In this study, we assessed the utility, ease of use, and improvement recommendations for the heuristics. Section 5 presents the experiment results in detail.

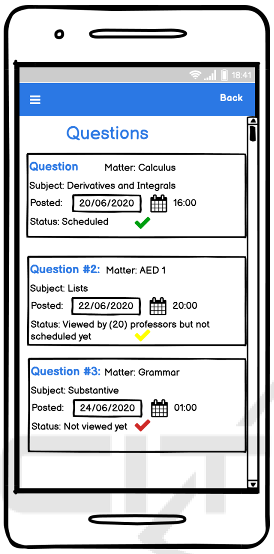
In step 8, the Refinement stage, we performed refinements in the set of heuristics using the validation stage results. We merged the heuristics *Content Filter* and *Search*. We also merged the Grid and List heuristics because they represent groups of objects with different arrangements. We also added the carousel as a form of presenting objects in this heuristic. The expert analysis also suggested the inclusion of heuristics to design prototypes with accessibility requirements and evaluation forms.

4 UI PROTOTYPE DESIGN HEURISTICS

This section presents characterization heuristics for the categories defined during the labeling process. Due to space limitation, we present three heuristics from the Layout and three from the Functional categories. We chose the more representative heuristics from each category, i.e., with the higher number of instances. The remaining heuristics are available in the technical report[2].

4.1 Layout Heuristics

Layout heuristics comprise UI prototypes whose predominant characteristic is to present general application features. Figure 2 shows a List prototype example. In this section, we detailed three layout heuristics: List, Menu, and Registration.


18:41

Back

Questions

Question Matter: Calculus
Subject: Derivatives and Integrals
Posted: 20/06/2020 16:00
Status: Scheduled ✓

Question #2: Matter: AED 1
Subject: Lists
Posted: 22/06/2020 20:00
Status: Viewed by (20) professors but not scheduled yet ✓

Question #3: Matter: Grammar
Subject: Substantive
Posted: 24/06/2020 01:00
Status: Not viewed yet ✓

Figure 2: Level Layout example – List prototype.

ID: HEU1-04
Name: **List**
Synonyms: Vertical list, horizontal list
Definition: List of objects from a domain entity
Explanation: A list presents a set of objects from an entity in the application domain.
Examples: A user views a movies list on a streaming service application.
Additional information:

List of domain entities and attributes.

Checklist:

1. The prototype presents a list of objects from an entity in the application domain.

2. Each item in the list presents values of attributes from the object.

Related heuristics: Search, Videoconference, Master-Detail, Grid

[2]Technical report available at https://doi.org/10.6084/m9.figshare.17087717

ID: HEU1-06
Name: **Menu**
Synonyms: Toolbar, drop-down menu, top bar, bottom bar
Definition: Application features options.
Explanation: A menu displays the actions a user can perform on a screen.
Examples: The user browses departments of an online clothes store.
Additional information:

1. Application features.
2. Application features labels.
3. Application features icons.

Checklist:

1. The prototype presents a set of elements in a side list, top bar, bottom bar, or buttons.

2. Elements labels and icons represent the functionalities offered by the application.

Related heuristics: none

ID: HEU1-07
Name: **Registration**
Synonyms: Record
Definition: Form for registering an object of an entity in the application domain.
Explanation: A registration form illustrates user input information that is stored in the application's database.
Examples: A news website employee publishes information about the municipal elections.
Additional information:

1. List of domain entities and attributes.

2. RegBtnLblSet – set of labels to describe registration buttons. Examples: create, send, save, request inclusion, add, forward, finalize, confirm, register, generate.

3. RegIconSet – set of icons to represent registration buttons.

Checklist:

1. The prototype presents a data entry form with one or more fields whose values are not filled in.

2. The form fields represent attributes of an object in the application domain.

3. The form presents a button whose label belongs to a set of labels used to describe registration buttons (RegBtnLblSet) or whose icon belongs to a set of icons used to represent registration buttons

(RegIconSet).
Related heuristics: Account Registration, File Upload

4.2 Functional Heuristics

Functional heuristics comprise UI prototypes that represent high level functionalities that are frequent in many applications. In this section, we present the heuristics specification for Account Registration, Authentication and Search prototypes categories. Figure 3 shows an Account Registration prototype example.

ID: HEU2-01
Name: **Account Registration**
Synonyms: Sign up, sign-up, signup, user registration
Definition: Creates a user account to access the application's features.
Explanation: A user from the application creates an account to access personalized or restricted features.
Examples: A user registers for a restaurant website to order a pizza.
Additional information:

1. List of user entities and attributes from the application domain.

2. UserIdLblSet – a set of labels to describe user identification fields. Examples: id, login, email, identification, name, user, username, registration number, phone number.

3. PwdLblSet – a set of labels used to describe password fields. Examples: password, keyword.

4. RetPwdLblSet – a set of labels used to describe retype-password fields. Examples: confirm password, retype your password.

5. RegBtnLblSet – a set of labels to describe registration buttons. Examples: register, confirm, confirm registration, finalize, continue, create account, finalize registration, complete, register.

6. RegIconSet – a set of icons to represent registration buttons.

Checklist:

1. The prototype presents a data entry form with one or more fields whose values are not filled in.

2. The form fields represent attributes of a user entity in the application domain.

3. The form has a user identification, a password and a retype-password fields.

4.The label of the user identification field belongs to a set of labels used to describe user identification fields (UserIdLblSet).

5. The label of the password field belongs to a set of labels used to describe password fields (PwdLblSet).

6. The label of the retype-password field belongs to a set of labels used to describe retype-password fields (RetPwdLblSet).

7. The form has a registration button whose label belongs to a set of labels used to describe registration buttons (RegBtnLblSet), or whose icon belongs to a set of icons used to represent registration buttons (RegIconSet).
Related heuristics: Registration

ID: HEU2-02
Name: **Authentication**
Synonyms: Login, log in
Definition: Credential information form for accessing application resources.
Explanation: A user types in identification and a password to access restricted application features.
Examples: A user access a social media application.
Additional information:

1. UserIdLblSet – a set of labels to describe user identification fields. Examples: id, login, email, identification, name, user, username, registration number, phone number.

2. PwdLblSet – a set of labels used to describe password fields. Examples: password, keyword.

3. AuthBtnLblSet – a set of labels to describe authentication buttons. Examples: login, log in, connect, enter, log in with google, log in with facebook, log in with gmail account, continue.

4. AuthIconSet – a set of icons to represent authentication buttons.

Checklist:

1. The prototype presents a data entry form with a user identification field and a password field.

2. The label of the user identification field belongs to a set of labels used to describe user identification fields (UserIdLblSet).

3. The label of the password field belongs to a set of labels used to describe password fields (PwdLblSet).

4. The form presents a button whose label belongs to a set of labels used to describe authentication

buttons (AuthBtnLblSet) or whose icon belongs to a set of icons used to represent authentication buttons (AuthIconSet).

5. Optionally, the prototype presents: a *Forgot Password* link; a *Login with another login system* option.
Related heuristics: none

ID: HEU2-07
Name: **Search**
Synonyms: Find
Definition: Search engine
Explanation: A Search prototype illustrates how users enter keywords to find information in a software application.
Examples: A user looks for a book by its name in an online bookstore.
Additional information:

1. List of user entities and attributes from the application domain.

2. SrchLblSet – a set of labels to describe search term fields. Examples: search by name, type the ID, search product, what are you looking for.

3. SrchBtnLblSet – a set of labels to describe search buttons. Examples: search, find.

4. SrchIconSet – a set of icons to represent search buttons.
Checklist:

1. The prototype presents a form with a text field related to a domain entity to inform the search terms.

2. The text field label belongs to a set of labels used to describe search term fields (SrchLblSet).

3. The prototype presents a Button whose label belongs to a set of labels used to describe search buttons (SrchBtnLblSet) or whose icon belongs to a set of icons used to represent search buttons (SrchIconSet).

4. Optionally, the prototype presents a list of objects from a domain entity illustrating the search results.
Related heuristics: List, Content Filter

5 HEURISTICS VALIDATION

This section presents the validation experiment results of our design heuristics set. We validated the set of heuristics through an experiment that analyzed the perception of UI design experts. First, we selected two HCI experts and two software development practitioners for the experiment. Next, the experts evaluated the heuristics through a survey and a focus group. The following sections present the details of experiment preparation and analysis of the results.

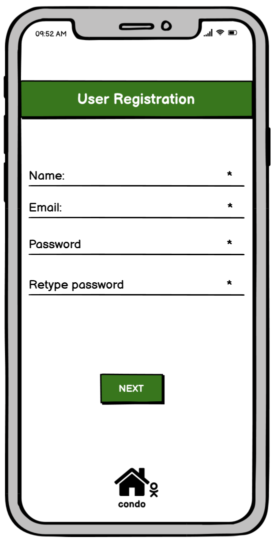


Figure 3: Example of an Account Registration prototype.

5.1 Preparation

To validate the heuristics, we conducted a Validation through an Expert Judgment experiment (Quiñones et al., 2018). As it was the first experiment to evaluate the heuristics, the study aimed to assess the general perception of specialists concerning the dimensions proposed by Quinones et al. (2018) , which consist of Utility, Clarity, Ease of Use, and Necessity of an Additional checklist.

The study consisted of UI prototyping activities for a fictitious e-commerce website. We asked the experts to design UI prototypes for three freely-chosen requirements from those presented in the problem specification.

The researchers recommended that the experts use the set of heuristics during prototyping. To facilitate using the heuristics set, we created a website with the heuristics specifications accompanied by examples from the prototype base developed by the students.

At the end of the prototypes' design, the experts filled out a questionnaire to evaluate the heuristics. Table 4 presents the questions of this questionnaire. After completing the questionnaire, we conducted a focus group with the experts, where they evaluated the heuristics and the prototyping experience using them.

Table 4: Survey on the experts' judgment.

#	Question
1	Were any of the heuristics useful in developing the prototype? Please check the ones that were used below.
2	Have you made any modifications to your prototypes after consulting the heuristics? If yes, please detail these changes.
3	Would you like to suggest modifications to existing heuristics? What modifications?
4	Would you like to suggest new heuristics? Which ones?
5	Would you like to suggest removing heuristics that you think are unnecessary? Which ones?

5.2 Results

We performed the analysis of the experiment results according to the aspects of Utility, Clarity, Ease of Use, and Necessity of an Additional Checklist, presented by Quiñones et al. (2018) .

Regarding the Utility dimension, all participants claimed to have used heuristics to develop or improve their prototypes. Specifically, the evaluators mentioned using the List, Search, View, Grid, Registration, Product Page, Chat, Shopping Cart, and Content Filter heuristics. One of the practitioners (P1) reported that during the design of a Registration prototype (HEU1-07), which he often implements, he included elements that were not in the checklist and were, in fact, unnecessary. Likewise, he missed several elements when designing a shopping cart prototype, which he often does not implement. He noted the missing items only after using the Shopping Cart heuristic (HEU2-08). The other practitioner (P2) mentioned that heuristics could be used alone or combined to form more complex prototypes.

Regarding Clarity's dimension, the experts reported some problems. The two HCI specialists and one of the practitioners indicated that they were confused about the differences between the Search (HEU2-07) and Content Filter (HEU1-01) heuristics and between the List and Grid heuristics. Regarding the Search and Content Filter heuristics, the experts reported that the specifications are complementary and could be unified. Regarding the List and Grid heuristics, they reported that they are different ways of presenting a group of objects, suggesting the inclusion of a carousel display. One of the HCI specialists did not understand the meaning of the Master-Detail heuristic (HEU1-05). However, the practitioners reported it as a heuristic of great relevance in the projects they develop. Also, one of the evaluators did not understand the relationship between the View and User Profile heuristics, suggesting that the most appropriate relationship would be with the Registration heuristic.

As for the Ease of Use dimension, the evaluators reported some issues. For example, one of the HCI experts found it difficult to memorize the name of the heuristics. He also used the View heuristic (HEU1-09) in the product page prototype design because he did not find the corresponding heuristic (HEU2-06). Despite this, the evaluators also reported ease in applying the heuristics to the modeled prototype and praised the navigation between heuristics through links in the Related Heuristics section.

Regarding the Necessity of an Additional Checklist dimension, the evaluators suggested the addition of two new design heuristics: for the design of rating and accessibility UIs. The suggestion of these new categories showed that functionality heuristics need to cover more diverse aspects of prototype design. The specification of new heuristics demands new analysis activities and will be the subject of future work.

In addition, the discussion on the problems reported by the evaluators resulted in the following changes in the heuristics set:

- Addition of an indication to the accepted password format to the Account Registration heuristic checklist (HEU2-01);
- Unification of Content Filter (HEU1-01) and Search (HEU2-07) heuristics;
- Unification of List (HEU1-04) and Grid (HEU1-03) heuristics. Addition of a Carousel as an option for displaying an object list.
- Addition of a relationship between User Profile (HEU2-09) and Registration (HEU1-07) heuristics.

The heuristics presented in this work specify what distinguishes a prototype of a particular category. However, the evaluators confused their purpose in several situations. They understood that the heuristics indicated how to design prototypes concerning the arrangement of components on the screen. For instance, one of the HCI experts (P3) reported that a Content Filter (HEU1-01) example used bright red and green colors, suggesting it as a design recommendation. The other HCI expert (P4) reported that he decided to design a Registration prototype (HEU1-07) in multiple pages, requiring too many user clicks, by looking at examples of this heuristic.

When analyzing the experiment data, we verified that the availability of design examples from the prototype base obtained in Step 2 induced the participants to understand that the prototypes should have a similar appearance to the examples. Another factor we identified was that the Grid (HEU1-03) and List (HEU1-04) heuristics contained indications of how the objects should be arranged in their specification, in disagreement with the other heuristics and the catalog's purpose.

6 DISCUSSIONS

In this work, we presented a set of design heuristics for UI prototypes. These heuristics and their framing at different semantic levels can be used to assist in user interface design tasks, such as searching for interfaces with similar functionality or verifying design adequacy to predefined patterns in the heuristics.

Compared to image-based UI designs classification methods, the heuristic-based approach has the advantage of being based on the semantic characteristics of the UI design. Therefore, it is not sensitive to the placement of UI components or interfered with by overlayed menus and notifications. Despite that, approaches based on images, image metadata, and user interaction data (Zecheng et al., 2020) can be combined with the heuristic-based approach during the classification process, as indicated by Li et al. (2021), or complemented after classification, where the heuristics would work as a checklist to verify the adequacy of the UI design.

Regarding the aspects that are semantically analyzed, the work by Liu et al. (2018) classifies structural components, text buttons, and icons, located at a semantic level prior to those presented in this article, which classifies interfaces as a whole and not isolated elements. The Enrico dataset (Leiva et al., 2020), in turn, filtered and classified a subset of the Rico dataset, finding 20 categories, although it does not consider semantic levels. Their work lists several applications of the labeled dataset, some of which apply to the dataset we present in this work: UI description based on the identified label, UI classification, listing the most likely labels for a UI, and optimized search of rank-based GUIs.

Like our work, the work of Chen et al. (2020a) also identified tags categories at different semantic levels: platform, color, app functionality, screen functionality, and screen layout. In their work, each UI can receive up to one tag from each of the identified categories. In our approach, most UI prototypes fit into only one category or class, except for prototypes labeled with more than one class.

Some of the classes identified in this work have few instances, limiting the relevant features found in the examples to define the respective heuristics and making it difficult to develop a UI prototype classifier in the following steps. To mitigate this problem, we can extend our dataset by incorporating labeled databases as the Enrico dataset (Leiva et al., 2020).

7 CONCLUSION

This research presented an approach to classifying interface prototypes based on heuristics. First, we presented the classes of UI prototypes identified during the labeling process and the respective set of identification heuristics. We also identified three semantic levels in the prototypes, which supported the creation of heuristics.

The heuristics we present in this article are not a closed set, as we created them on the basis of our set of UI prototypes. In the future, the dataset can be extended by incorporating prototypes of new prototyping activities with students, resulting in new categories and heuristics.

The next stage of our work consists of using the set of heuristics to develop a UI prototype classification and analysis tool. Our goal is to support the interface design process, offering deeper analysis than those performed in similar works (Lee et al., 2020), by incorporating semantic information into the analysis process.

ACKNOWLEDGEMENTS

We are grateful to the Amazonas State Prosecution Service – MPE/AM for the incentive to this research provided under the Program for the Functional Improvement of Employees.

This research, carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree nº 6.008/2006(SUFRAMA), was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law nº 8.387/1991, through agreement 001/2020, signed with Federal University of Amazonas and FAEPI, Brazil. This research was also supported by the Brazilian funding agency FAPEAM through process number 062.00150/2020, the Coordination for the Improvement of Higher Education Personnel-Brazil (CAPES) financial code 001 and, the São Paulo Research Foundation (FAPESP) under Grant 2020/05191-2.

REFERENCES

Bernal-Cárdenas, C., Moran, K., Tufano, M., Liu, Z., Nan, L., Shi, Z., and Poshyvanyk, D. (2019). Guigle: A gui search engine for android apps. In *2019 IEEE/ACM 41st ICSE-Companion*, pages 71–74. IEEE.

Bjarnason, E., Lang, F., and Mjöberg, A. (2021). A model of software prototyping based on a systematic map. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–11.

Calle-Escobar, M., Mejía-Gutiérrez, R., Nadeau, J.-P., and Pailhes, J. (2016). Heuristics-based design process. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 10(4):369–386.

Chen, C., Feng, S., Liu, Z., Xing, Z., and Zhao, S. (2020a). From lost to found: Discover missing ui design semantics through recovering missing tags. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW2).

Chen, J., Chen, C., Xing, Z., Xia, X., Zhu, L., Grundy, J., and Wang, J. (2020b). Wireframe-based ui design search through image autoencoder. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 29(3):1–31.

Deininger, M., Daly, S. R., Sienko, K. H., and Lee, J. C. (2017). Novice designers' use of prototypes in engineering design. *Design Studies*, 51:25–65.

Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., and Kumar, R. (2017). Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 845–854.

Dow, S., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, D., and Klemmer, S. (2011). Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2807–2816.

Felderer, M. and Travassos, G. H. (2020). *Contemporary Empirical Methods in Software Engineering*. Springer.

Figueiredo, E., Sant'Anna, C., Garcia, A., and Lucena, C. (2012). Applying and evaluating concern-sensitive design heuristics. *Journal of Systems and Software*, 85(2):227–243.

Garcia, A., Silva da Silva, T., and Selbach Silveira, M. (2017). Artifacts for agile user-centered design: a systematic mapping. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Gigerenzer, G. (2008). Why heuristics work. *Perspectives on psychological science*, 3(1):20–29.

Huang, F., Canny, J. F., and Nichols, J. (2019). Swire: Sketch-based user interface retrieval. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–10.

Käpyaho, M. and Kauppinen, M. (2015). Agile requirements engineering with prototyping: A case study. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 334–343.

Lauff, C. A., Knight, D., Kotys-Schwartz, D., and Rentschler, M. E. (2020). The role of prototypes in communication between stakeholders. *Design Studies*, 66:1–34.

Lee, C., Kim, S., Han, D., Yang, H., Park, Y.-W., Kwon, B. C., and Ko, S. (2020). Guicomp: A gui design assistant with real-time, multi-faceted feedback. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Leiva, L. A., Hota, A., and Oulasvirta, A. (2020). Enrico: A dataset for topic modeling of mobile ui designs. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'20 Extended Abstracts)*.

Lenarduzzi, V. and Taibi, D. (2016). Mvp explained: A systematic mapping study on the definitions of minimal viable product. In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 112–119. IEEE.

Li, T. J.-J., Popowski, L., Mitchell, T., and Myers, B. A. (2021). *Screen2Vec: Semantic Embedding of GUI Screens and GUI Components*. Association for Computing Machinery, New York, NY, USA.

Liu, T. F., Craft, M., Situ, J., Yumer, E., Mech, R., and Kumar, R. (2018). Learning design semantics for mobile apps. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pages 569–579. ACM.

Moran, K., Li, B., Bernal-Cárdenas, C., Jelf, D., and Poshyvanyk, D. (2018). Automated reporting of gui design violations for mobile apps. In *Proceedings of the 40th International Conference on Software Engineering*, pages 165–175.

Packevičius, Š., Barisas, D., Ušaniov, A., Guogis, E., and Bareiša, E. (2018). Text semantics and layout defects detection in android apps using dynamic execution and screenshot analysis. In *International Conference on Information and Software Technologies*, pages 279–292. Springer.

Quiñones, D., Rusu, C., and Rusu, V. (2018). A methodology to develop usability/user experience heuristics. *Computer Standards & Interfaces*, 59:109–129.

Shah, A. K. and Oppenheimer, D. M. (2008). Heuristics made easy: an effort-reduction framework. *Psychological bulletin*, 134(2):207.

Toxboe, A. (2022). User interface design pattern gallery. http://ui-patterns.com/.

Yilmaz, S., Daly, S. R., Seifert, C. M., and Gonzalez, R. (2016). Evidence-based design heuristics for idea generation. *Design Studies*, 46:95–124.

Zang, X., Xu, Y., and Chen, J. (2021). Multimodal icon annotation for mobile applications. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*, pages 1–11.

Zecheng, H., Sunkara, S., Zang, X., Xu, Y., Liu, L., Wichers, N., Schubiner, G., Lee, R., and Chen, J. (2020). Actionbert: Leveraging user actions for semantic understanding of user interfaces. *arXiv preprint arXiv:2012.12350.*