

Article

A Hybrid Recommender System for HCI Design Pattern Recommendations

Amani Braham ^{1,2,*} , Maha Khemaja ³ , Félix Buendía ⁴  and Faiez Gargouri ⁵

¹ Department of Computer Engineering, Universitat Politècnica de Valencia, Camino de Vera S/N, 46022 Valencia, Spain

² ISITCOM, University of Sousse, Sousse 4011, Tunisia

³ PRINCE Research Lab, ISITCOM, University of Sousse, Sousse 4011, Tunisia; maha_khemaja@yahoo.fr

⁴ Escuela Técnica Superior de Informática (ETSINF), Universitat Politècnica de Valencia, Camino de Vera S/N, 46022 Valencia, Spain; fbuendia@disca.upv.es

⁵ MIRACL Laboratory, University of Sfax, Sfax 3029, Tunisia; faiez.gargouri@usf.tn

* Correspondence: amanibraham@gmail.com; Tel.: +34-600-21-44-52

Abstract: User interface design patterns are acknowledged as a standard solution to recurring design problems. The heterogeneity of existing design patterns makes the selection of relevant ones difficult. To tackle these concerns, the current work contributes in a twofold manner. The first contribution is the development of a recommender system for selecting the most relevant design patterns in the Human Computer Interaction (HCI) domain. This system introduces a hybrid approach that combines text-based and ontology-based techniques and is aimed at using semantic similarity along with ontology models to retrieve appropriate HCI design patterns. The second contribution addresses the validation of the proposed recommender system regarding the acceptance intention towards our system by assessing the perceived experience and the perceived accuracy. To this purpose, we conducted a user-centric evaluation experiment wherein participants were invited to fill pre-study and post-test questionnaires. The findings of the evaluation study revealed that the perceived experience of the proposed system's quality and the accuracy of the recommended design patterns were assessed positively.

Keywords: HCI; design patterns; design problems; semantic similarity; ontology models; recommender system



Citation: Braham, A.; Khemaja, M.; Buendía, F.; Gargouri, F. A Hybrid Recommender System for HCI Design Pattern Recommendations. *Appl. Sci.* **2021**, *11*, 10776. <https://doi.org/10.3390/app112210776>

Academic Editors:
Alessandro Micarelli,
Giuseppe Sansonetti and
Giuseppe D'Aniello

Received: 20 September 2021
Accepted: 11 November 2021
Published: 15 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The continuous advance in the development of Information Technology (IT) is currently witnessing a rapid growth of platforms, devices, and environments [1]. This has promoted an increase in design possibilities and a widespread interest in the study of User Interface (UI) [2] within the HCI research community to satisfy user requirements. In this context, the development of adaptive applications is attracting increasing attention, causing developers and designers to face great difficulties in designing and implementing applications that meet the dynamics of their environment. Hence, these applications open up new challenges as users need adaptive UIs that can cope with their corresponding preferences, surrounding context, and specific requirements. In general, adaptive UIs are supposed to adapt interaction contents and information processing modes automatically to deal with changing context and users' needs and disabilities at any time [3]. Nevertheless, the main drawback of these interfaces is their developmental complexity, which requires significant efforts. Recently, adaptive UIs have made tremendous progress regarding the big evolution of technology. This fact makes the task of development even more complex by requiring extra knowledge and expertise. Therefore, UI developers need to be assisted in designing and developing adaptive interfaces. One method to assist and help developers is to use design patterns so that these interfaces are developed using reusable design solutions, rather than from scratch.

The cornerstone of the design pattern concept was laid down in the architecture domain by Alexander [4]. This concept was initially meant to focus on frequent problems faced by designers in order to offer a correct solution within a particular context. The design pattern concept was later transferred to software design when Gamma et al. [5] introduced design patterns as a way to share the design solutions of experienced developers. After that, design patterns emerged in the HCI domain to capture HCI knowledge [6]. They have sparked interest in various areas, including UI and Web design [7–9]. In this sense, developers can take advantage of freely reusing existing design knowledge to elaborate efficient and adaptive UIs and save on development time. In recent decades, an increasing number of design patterns in the HCI domain have been noticed. Moreover, several design pattern repositories and catalogues have been organized and published [10–12]. The sheer amount of available design patterns offers good design solutions to recurring design problems. Nevertheless, it is difficult for developers or designers to follow all available HCI design patterns during the development of UIs and to select the right design patterns when a design problem is tackled. This is especially true when design patterns are stored within more than one repository. To overcome these issues, a supporting system that recommends appropriate design patterns is required.

Recommender systems have become an emerging research area where information overload is a major problem. In general, recommender systems are filtering systems [13] that handle the problem of information overload by retrieving the most relevant elements and services from a large amount of data. In recent decades, various approaches have been proposed for developing recommender systems. In this sense, there are a significant number of studies that introduce different recommendation approaches for selecting relevant design patterns, including text classification [14], case-based reasoning [15], and ontology-based [16] approaches. Fortunately, various recommender systems consider these approaches to (semi-)automatically select appropriate design patterns. Nevertheless, design patterns in the HCI domain have not been well-adopted within existing systems.

In this work, we propose a different approach to address the challenge of exploring design pattern recommendations in the HCI domain. The main contribution of the present approach is twofold. The first contribution of this paper relates to the development of a recommender system for selecting the most relevant HCI design patterns for a given design problem. To achieve this, we propose a hybrid recommender system based on well-accepted recommendation techniques that combines text-based and ontology-based methods and is aimed at considering semantic similarity and ontology models for retrieving relevant HCI design patterns. Moreover, the purpose of the second contribution is to validate the proposed system in terms of participant acceptance intention towards our system by assessing the perceived experience of the recommender system and the perceived accuracy of the recommended design patterns.

The remainder of this paper is organized as follows. Section 2 reviews related work on design pattern recommender systems. Section 3 introduces the proposed recommender system. Section 4 outlines the implementation of the recommender system. Section 5 presents the design of the experiment. The statistical results extracted from the experiment and the discussion of the obtained results are provided in Sections 6 and 7, respectively. Finally, a conclusion and discussion on future research work are drawn in Section 8.

2. Related Work

Recommender systems have become an emerging research area in different domains. In this context, several studies have presented recommender systems for retrieving design patterns. This section reviews several significant works related to design pattern recommendation systems and provides a critical analysis of the discussed works.

2.1. Design Pattern Recommender System

In the literature, several research studies have been carried out on the recommendation of relevant design patterns. Each of these studies adopted different recommendation

techniques. Some of the existing works developed recommender systems for selecting relevant design patterns based on a text-based technique. These recommender systems are generally based on two main methods, including (i) text retrieval and text classification for natural language processing, and (ii) similarity measures between design patterns and design problem descriptions. In this sense, Hamdy and Elsayed [17] proposed a Natural Language Processing (NLP) recommender approach that was applied on a collection of 14 different software design patterns. This collection was created using pattern definitions from the *Gang-of-Four* (GoF) book, represented with a vector space model, and ranked according to similarity scores. From the collection of design patterns, retrieving the most suitable design patterns for a given design problem is based on the degree of similarity by adopting cosine similarity. Likewise, Hussain et al. [18] presented a framework that aids the classification and selection of software design patterns. Unsupervised learning and text categorization techniques were used to exploit their proposed framework. More specifically, these techniques were applied to perform the classification and the selection of software design patterns through the specification of design problem groups. This framework selects the right design pattern class for a given design problem based on the use of text classification technique and cosine similarity.

Other recommendation techniques are based on questions from which the appropriate design patterns are selected according to the answers provided by the user. The following are a couple of question-based approaches that make use of questionnaires to recommend design patterns. For instance, Youssef et al. [19] proposed a recommendation system based on the use of question-based techniques to recommend the appropriate design pattern category. This system examines the Goal Question Metric (GQM)-based tree model of questions. These questions are first answered by software engineers considering the user requirements. Then, the answers' weights are measured and the system recommends appropriate design pattern categories accordingly.

In the last few years, semantic technologies have been successfully applied in recommender systems. In particular, ontologies have been used in recommender systems to define and find relevant design patterns. In this context, Abdelhedi and Bouassidar [20] developed an ontology-based system for recommending Service-Oriented Architecture (SOA) design patterns. This recommender provides a questionnaire to users to retrieve their requirements. Using these requirements, an ontology that represents the different SOA pattern problems and their corresponding solutions is considered for recommending design patterns. This ontology was interrogated by SPARQL queries to search for the appropriate SOA design pattern. Similarly, Naghdipour et al. [21] proposed an ontology-based approach for selecting appropriate software design patterns to solve a given design problem. The presented method is based on interrogating an ontology of software patterns using queries to select the most suitable software design patterns according to the given design problem.

Recently, hybrid approaches that combine two or more recommendation techniques have come into focus. In this context, Celikkan and Bozoklar [22] have provided a recommendation tool that considers three main recommendation techniques, including text-based, case-based reasoning, and question-based technique. This tool aims to recommend adequate software design patterns for design problems whose description is text-based. To this end, the cosine similarity metric is computed to compare the design problem against design patterns, and to provide a ranked list of design patterns according to similarity measures. This list is then filtered to enhance recommendation results and to provide a refined list of design patterns considering the answers provided by designers.

2.2. Critics and Synthesis

Table 1 illustrates a comparison of the studied works with regard to the following criteria:

- Domain: design patterns have emerged out of different domains, such as software design patterns, SOA design patterns, and HCI design patterns.

- Problem input format: recommender systems require different problem input formats such as full-text, keywords, or questionnaires.
- Recommendation method: recommender systems consider various recommendation methods, namely text-based, case-based, question-based, and ontology-based methods.
- Degree of automation: the recommendation phase may be carried out semi-automatically when the role of users is required to some extent, or fully automatically without any human expert intervention for the selection of design patterns that ought to be recommended.
- Similarity approach: such recommender systems are based on the similarity of semantic or syntactic across a range of design pattern descriptions and problem scenarios.
- Knowledge support: recommender systems could support the reuse of knowledge by integrating ontology models.

Table 1. Comparison of the proposed work with existing design pattern recommendation systems.

Work	Domain	Problem Input Format	Recommendation Method	Degree of Automation	Similarity Approach	Knowledge Support
[17]	Software design patterns	Full-text	Text-based	Automatic	Syntactic (CS)	–
[18]	Software design patterns	Full-text	Text-based	Automatic	ntactic (CS)	–
[19]	Software design patterns	Questionnaire	Question-based	Semi-automatic	–	–
[20]	SOA design patterns	Questionnaire	Ontology-based	Semi-automatic	–	+
[21]	Software design patterns	Full-text	Ontology-based	x	–	+
[22]	Software design patterns	Full-text	Text-based, Case-based, Question-based	Semi-automatic	Syntactic (CS)	–
Our work	HCI design patterns	Full-text, Keywords	Text-based, Ontology-based	Automatic	Semantic	+

+ supported, – not supported, x not specified.

Although there have been many advances in the design pattern recommendation field, there are still problems to be dealt with, as can be seen in the comparative table (Table 1). For instance, we noticed that the recommendation domain covered in the aforementioned works concerns either software design patterns [17–19,21,22] or SOA design patterns [20]; nevertheless, they do not consider the HCI domain and tend to overlook HCI design patterns in practice despite the increase in design pattern collections in this emerging domain. In the present work, we provide a recommender system that covers the HCI domain by selecting the most relevant HCI design patterns according to specific problems.

Moreover, among the weaknesses that exist in previous works, one of them is the fact that the majority of these works rely on low-quality design problem input. For example, the approach presented in [17] recommends design patterns for predefined design problems that are written briefly. This fact may limit the set of real design problems in the sense that it restricts end-users' choices regarding design problem scenarios. On the contrary, the aim of the current work is to provide a more flexible recommender system that uses real design problem scenarios by offering end-users the ability to interact with the system and input the design problem, which could be based on full-text or keywords.

Furthermore, various existing works [17,18,22] adopted text-based recommendation techniques based on NLP methods and syntactic similarity. The syntactic similarity measurements aim at calculating the number of identical words using cosine similarity scores. In contrast, we propose a semantic similarity, which focuses more on the meaning and the interpretation-based similarity between design patterns and problem scenarios since it allows the integration of semantic information into the recommendation process [23]. Thus, the use of semantic similarity can greatly improve the text-based recommenda-

tion technique and, accordingly, the recommendation results. Other works are based on semi-automatic recommendation strategies. For instance, recommendations require the intervention of users to answer questionnaires [19]. Another work [20] invites users to select the appropriate design pattern category to get the recommended SOA patterns. This makes their methods rather semi-automatic. Alternatively, we propose a fully automatic recommender system that does not require any human intervention to retrieve HCI design patterns.

The use of ontology-based techniques can enhance the overall quality of recommender systems. However, limited research in this area has taken place in recommending design patterns. Existing ontology-based approaches [20,21] extract design patterns by means of queries, which are not sufficient for getting the appropriate ontology instances. On the contrary, we propose to improve ontology-based techniques by expanding ontology with inference rules together with SPARQL queries, allowing relevant design patterns to be deduced from the ontology model. Apart from queries, we consider the use of inference rules to enhance ontology's capabilities for revealing implicit knowledge and filtering the obtained recommendation results that better fit with the given design problem.

To address the gap within the existing research, this work proposes a novel recommender system that follows a hybrid method. This method combines text-based and ontology-based techniques to provide an automatic recommendation of relevant HCI design patterns. More specifically, the text-based technique uses NLP methods and semantic similarity measures, while the ontology-based technique relies on an ontology of HCI design patterns enriched with a set of SPARQL queries and inference rules.

3. Proposed Recommender System

The present work focuses on the recommendation of relevant HCI design patterns. To address this purpose, we propose a hybrid recommender system, named User Interface DDesign Patterns Recommender (IDEPAR), which is part of the global Adaptive User Interface Design Pattern (AUIDP) framework [24]. As illustrated in Figure 1, the global AUIDP framework incorporates two main systems, the IDEPAR system (Figure 1a) and the User Interface Code Generator using DDesign Patterns (ICGDEP) system (Figure 1b). While the IDEPAR system concerns the recommendation of relevant HCI design patterns, the ICGDEP system covers the implementation of design patterns recommended by the IDEPAR system to generate the final user interface to the end-user. In this work, we only describe the IDEPAR system to focus on the automatic recommendation of HCI design patterns. As shown in Figure 1a, the IDEPAR system requires a design problem as input to retrieve the most relevant design patterns. In the following subsections, we introduce the representation of design problems along with a detailed description of the IDEPAR system's architecture.

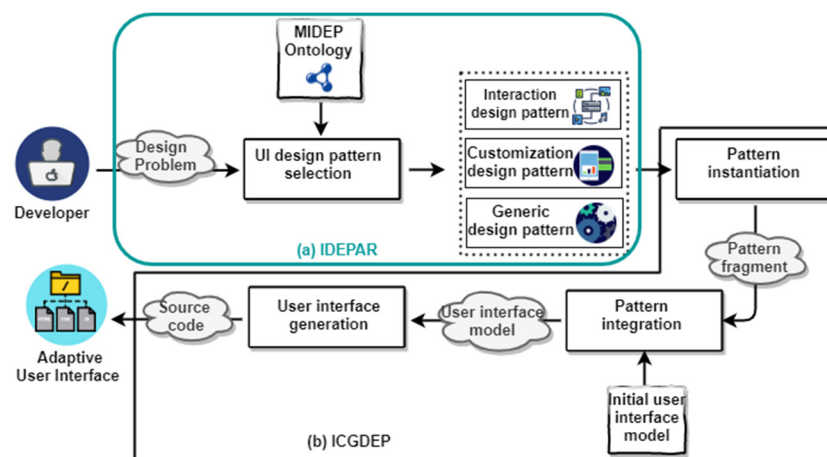


Figure 1. IDEPAR system within the AUIDP framework (a) IDEPAR system, (b) ICGDEP system.

3.1. Design Problem Representation

The IDEPAR system provides the possibility for developers to input their design problems in natural language to specify their requirements. Therefore, understanding design problems and investigating how such problems can be represented is crucial for providing recommendations that match with the given design problems. In this context, we propose an approach to design problem representation that (i) identifies the main elements that compose a design problem, and (ii) is formally represented via ontology models of design problems that will be used by the IDEPAR system. In addition, we classify design problems into atomic problems that are the smallest sub-design problems and composite problems that refer to problems that can be decomposed into simpler problems. Furthermore, we relate design problems to additional concepts, as illustrated in Table 2. The ontology model of design problem concepts within Protégé is displayed in Figure 2.

Table 2. Design problem concepts description.

Concept	Description
Design problem	A design problem can be atomic or composite
Overall description	A set of information that describes the design problem
Problem concept	Issues that constitutes a design problem (e.g., user interface issue, user characteristic, source code constraint, application functionality)
Problem category	A category associated with a design problem such as design time or runtime.

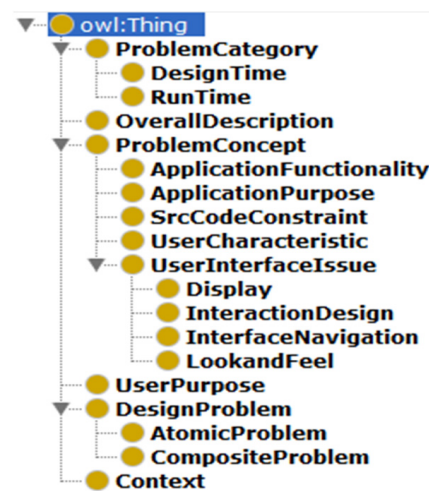


Figure 2. Design problem ontology model within Protégé.

3.2. Overall Architecture of the IDEPAR System

The IDEPAR system entails strategies to deal with design pattern recommendations regarding the text-based technique and the ontology-based technique by supporting a hybrid recommendation approach. As depicted in Figure 3, the IDEPAR system includes two main modules that interact among them, including the NLP module and the semantic module. A brief description of each module is introduced in the following subsections.

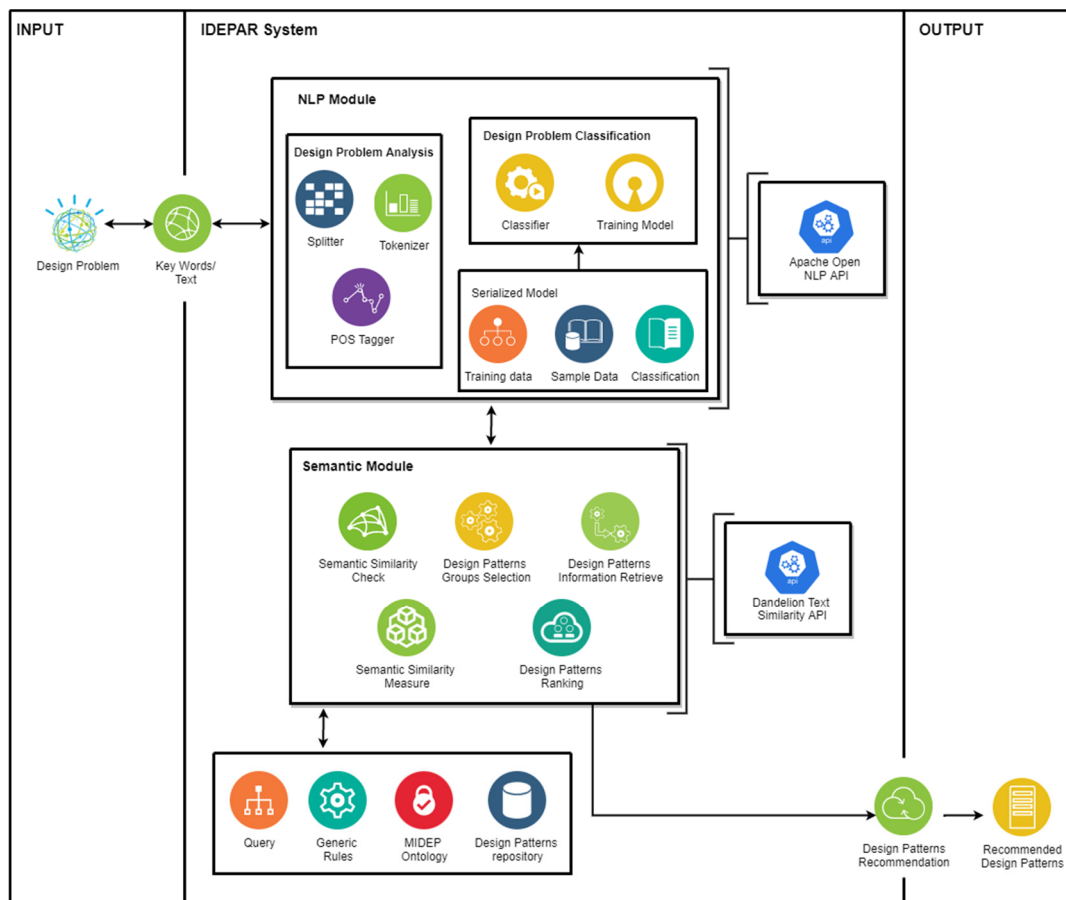


Figure 3. IDEPAR system architecture.

3.2.1. NLP Module

The NLP module is in charge of preprocessing the given design problems using a text-based technique. As input, it takes the definition of the design problem, which could be based on a set of key words or full-text format, and generates categories for each atomic problem. This module covers two main phases:

- Design problem analysis phase: At this phase, the NLP module preprocesses the given design problem. Then, it decomposes composite design problems into atomic ones. In particular, the NLP module applies three main strategies that consider the standard information retrieval method, including sentence split, tokenization, and Part of Speech (POS) tagging. The first strategy consists of splitting the composite design problems into atomic design problems. The second focuses on turning atomic design problems into small textual fragments, called tokens. The third strategy annotates tokens by assigning each token to its corresponding tag.
- Design problem classification phase: At this phase, the NLP module performs an automatic classification of atomic design problems based on the NLP auto-categorization method. It affects the categories of the design problem(s) retrieved from the previous phase. This phase mainly requires a training model generated from a set of training data. The training data can be presented in a sample data document, which includes classification samples of design problems.

3.2.2. Semantic Module

The main target of the semantic module is to perform an automated reasoning over the MIDEP ontology [24,25] and to select the most relevant design patterns based on an

ontology-based technique. In the following, we present an overview of the MIDEP ontology and describe the workflow of the semantic module.

- The MIDEP ontology:

The MIDEP ontology is a modular ontology that is built using the NeOn methodology [26]. This ontology presents a modeling solution for tackling recurring design problems related to user interfaces. As depicted in Figure 4, we distinguished three main modules that constitute the MIDEP ontology, including the design pattern module, the user profile module, and the user interface module. The proposed IDEPAR system considers a collection of 45 HCI design patterns that are formalized within the MIDEP ontology. A partial list of these design patterns, along with their corresponding design pattern category, group, problem, and solution, is illustrated in Table 3.

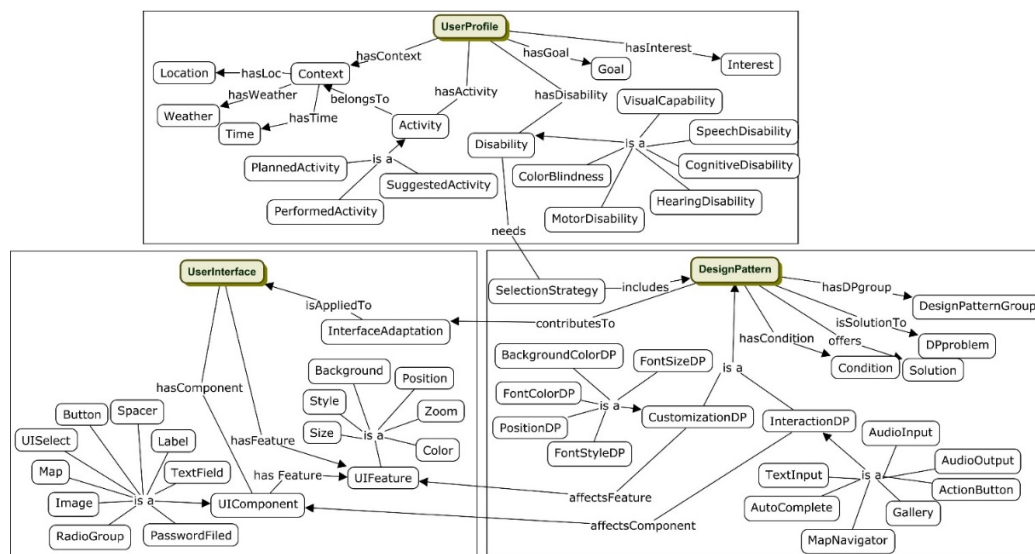


Figure 4. MIDEP ontology model.

Table 3. Partial list of HCI design patterns.

Design Pattern Group	Design Pattern Category	Design Pattern Name	Design Pattern Problem	Design Pattern Solution
Navigation	Interaction	MapNavigator	User needs to find a location of choice on a map	Display map navigator element
		Menu	User needs to access the main navigation	Repeat the main navigation on the bottom of the page
		NavigationTab	Content needs to be separated into sections and accessed using a flat navigation structure	Display a horizontal bar containing the different sections or categories
FontColor	Customization	LightFont DarkFont ColoredFont	User has difficulties perceiving font color	Set light font color Set dark font color Set colored font color

- Semantic module workflow:

The semantic module workflow, depicted in Figure 5, takes design problem categories affected by the NLP module as input and outputs a list of the recommended design patterns. A detailed description of each phase is provided below.

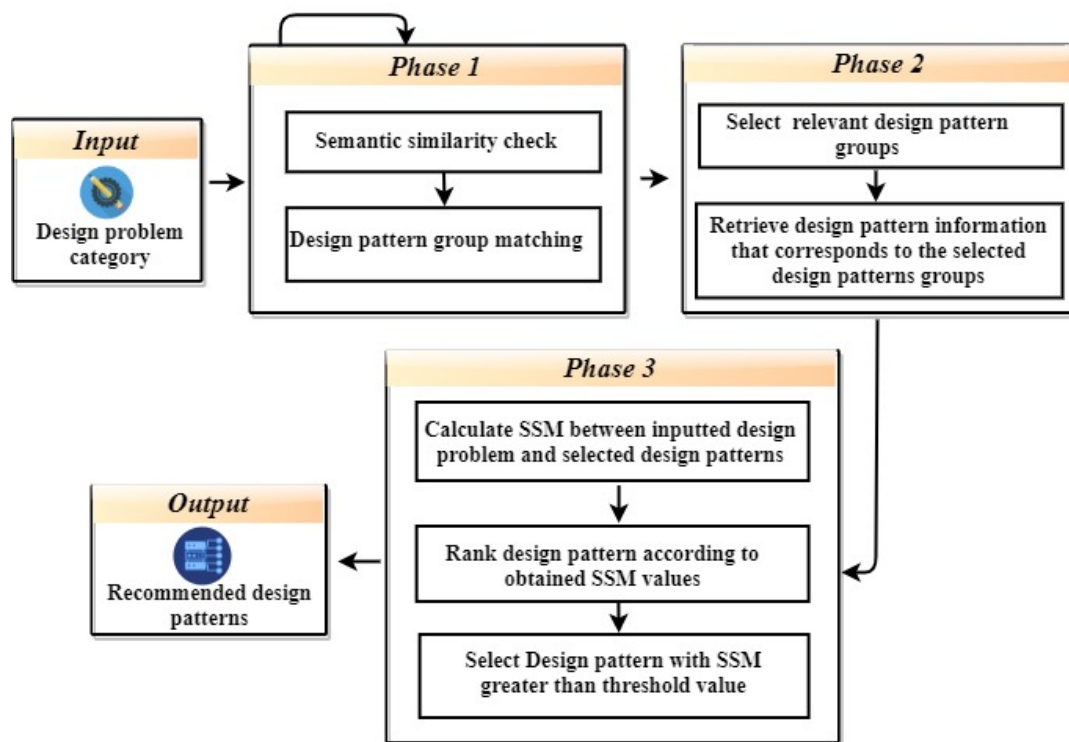


Figure 5. Semantic module workflow.

1. Phase 1: The workflow starts by checking the semantic similarity between design problem categories and design pattern group. First, the semantic module calculates the Semantic Similarity Measures (SSM) and creates relationships between concepts that are semantically similar. In Figure 6, we provide an algorithm that illustrates this process in more detail. After that, the semantic module performs a matching between MIDEP ontology instances, including the design problem concepts and design pattern group. At this level of matching, the present module applies generic rules using the inference engine. These inference rules include an antecedent and consequent part; whenever the “conditions” presented in the antecedent part hold, the “facts” specified in the consequent part must also hold. An example of a matching rule applied in this phase is illustrated in Table 4.
2. Phase 2: The second phase within the semantic module addresses the selection of the design pattern group and their corresponding design patterns. In particular, the present module makes inferences on the MIDEP ontology using a reasoning mechanism based on the “hasDPgroup” relationship between the ontological concepts.

Table 4. A rule example for matching design pattern groups.

Rule	$\text{DesignProblem}(?x) \wedge \text{isComposedofProblemConcept}(?x,?a) \wedge$ $\text{DesignPatternGroup}(?y) \wedge \text{hasContext}(?y,?b) \wedge$ $\text{isSemanticallySimilarTo}(?a,?b) \rightarrow \text{matcheWith}(?y,?x)$
Description	<p>Design problem “x” is composed of problem concept “Conceptx”, design pattern group “y” has context “a”, “Conceptx” is semantically similar to “b”, then “y” matches with “x”.</p>

```

Initialize x, y, ont ; // x: design problem , y: design pattern group,
                    // ont: MIDEP ontology
M ← readModel(ont);
Cx ← getConcept(M,x);
Cy ← getContext(M,y);
N ← numberDPgroupInstance(M);
for i to N do
  SSM(Cx,Cy) ← semanticSimilarity(Cx,Cy);
end
foreach SSM(Cx,Cy) do
  if (SSM(Cx,Cy)≠ null do
    createObjectProperty(M, isSemanticallySimilarTo,x,y);
  end if
  else do
    z ← newIndividual(M);
    createObjectProperty(M, isSemanticallySimilarTo,x,z);
  end else
end
end
end

```

Figure 6. Semantic similarity check algorithm.

- Phase 3: After retrieving an initial list of design patterns, the last phase computes the SSM between the design problem categories, affected in the NLP module, and the descriptions of design patterns. Then, the semantic module ranks design patterns using the obtained SSM and selects the most relevant design patterns for the given design problem, following Equation (1):

$$SSM(A, B) > \alpha, \quad (1)$$

where “A” and “B” are the text of the design problem category and design pattern condition, respectively. “ α ” is a threshold value for the similarity measures. As part of the design pattern ranking process, we note that design patterns with an SSM value below 0.4 are not relevant to the design problems. Therefore, a threshold of 0.4 is considered.

4. Recommender System Implementation

4.1. Implementation of Server-Side System

The IDEPAR system was implemented as a Web service that can be operated using a RESTful API and thus can be deployed on any Java application server that is able to run services packages as jar files. Moreover, the IDEPAR system consists of seven micro-services that communicate via REST calls, as illustrated in Figure 7.

For developing the environment in which the IDEPAR system is exposed as a Web service, this work leveraged different tools and technologies, including (i) Jersey as a RESTful Web service container that provides Web services, (ii) Apache Tomcat as a Web server to host Jersey and RESTful Web services, (iii) Spring Framework for dependency injection, (iv) Apache Jena for reasoning over the MIDEP ontology and processing SPARQL queries, (v) Apache OpenNLP API for processing natural language text, and (iv) Dandelion Text Similarity API for identifying the semantic relationships between texts.

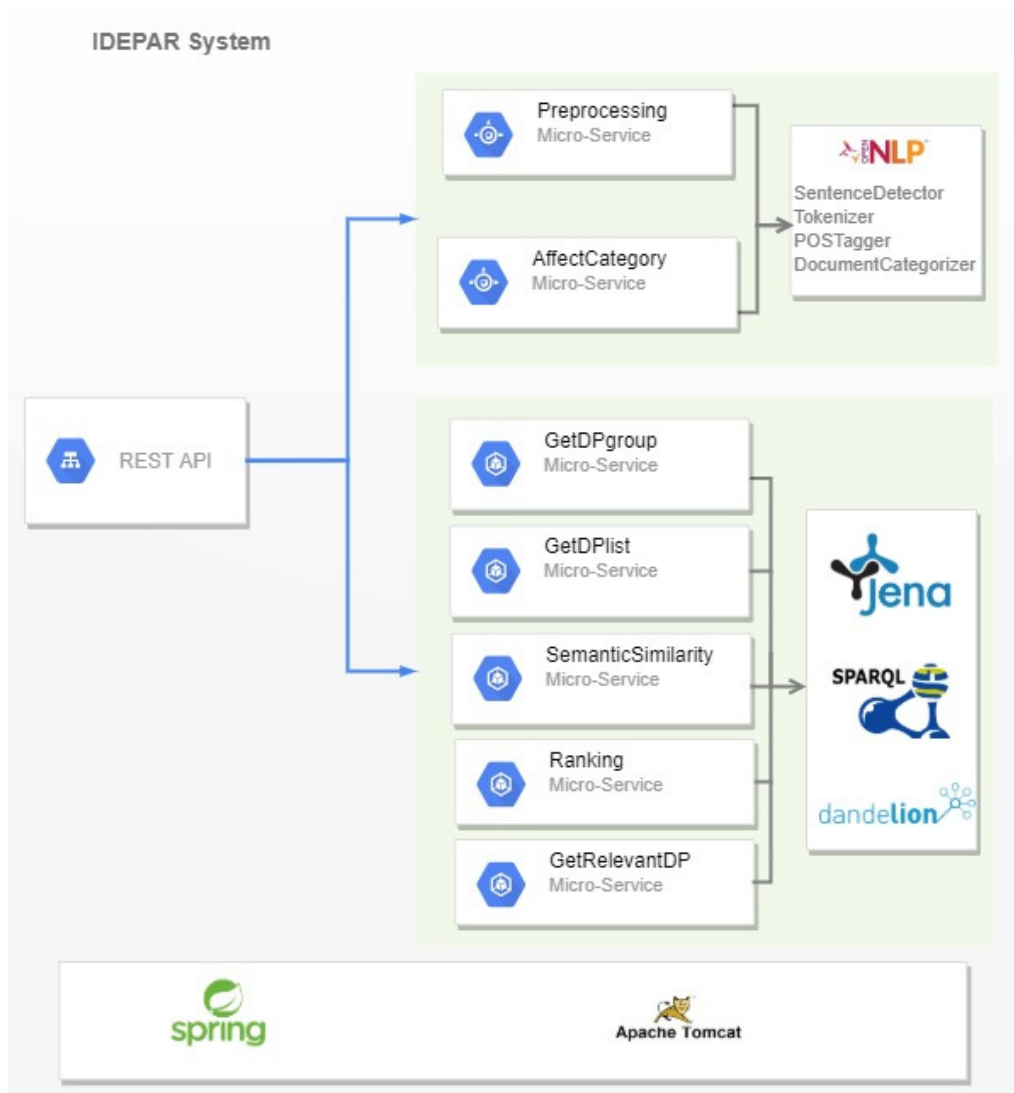


Figure 7. IDEPAR system—server-side implementation.

4.2. Design Pattern Recommendation Example

In order to illustrate a recommendation example, we describe how the IDEPAR system is applied for a particular design problem scenario. As an example of a design problem, we considered the following scenario (DPS-1): “The user cannot perceive colors, The user needs to find the location of a point of interest”. A detailed description of the results of each module within the proposed recommender system is further illustrated.

The given design problem (DPS-1) was processed through various steps in the NLP module. First, in the splitter step, the design problem (DPS-1) was divided into sentences using the Sentence Detection API so that different design problem sentences could be extracted. Individual sentences were identified in the given scenario and long sentences were split into short sentences with the aim of identifying atomic design problems. As a result, the design problem (DPS-1) was split into two atomic problems: “the user cannot perceive colors” (DPS-1-1) and “The user needs to find the location of a point of interest” (DPS-1-2). In Figure 8, we provide the results of the splitter step.

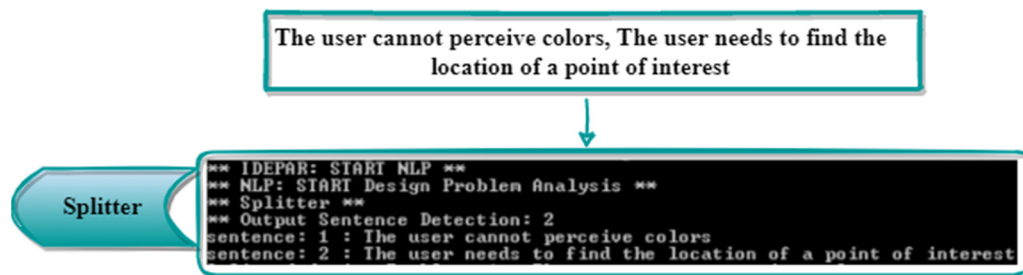


Figure 8. Splitter results for DPS-1.

Then, in the tokenizer step, the two atomic design problems were tokenized using the Tokenizer API. As illustrated in Figure 9a, each sentence was transformed into an object wherein each word was represented as a small fragment, called a token. Next, in the POS tagger step, the NLP module assigned POS tags to tokens obtained from the tokenizer step. All tokens were marked with their POS tags, as shown in Figure 9b.

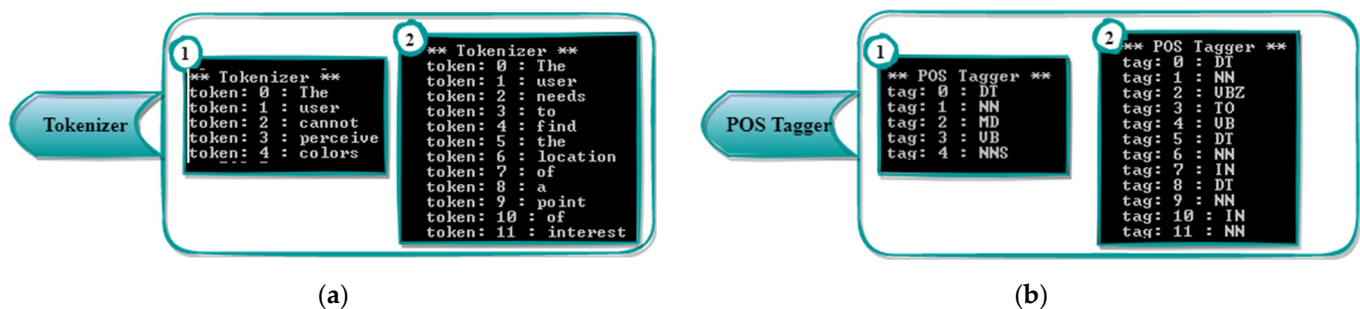


Figure 9. Results for DPS-1: (a) tokenizer results; (b) POS tagger results.

Finally, based on the tags assigned in the previous step, only nouns and verbs were part of the classifier step. The Document Categorizer API was considered to affect categories for each atomic design problem. More specifically, a training model was used to identify the appropriate categories by providing the nouns and verbs of each atomic problem. As illustrated in Figure 10, the categories “Colorblindness” and “NavigateToMap” were assigned to DPS-1-1 and DPS-1-2, respectively.

The result of the NLP module for DPS-1 is depicted in Figure 11. The given design problem (DPS-1) was passed as input parameters to the “getNLPmoduleResult” service, which communicates with the “Preprocessing” and “AffectCategory” micro-services, presented in Figure 7. The response body of the developed service was provided in a string format (Atomic design problem => Category) that would be used in the semantic module.

The output from the NLP module was used in the semantic module to retrieve the most relevant HCI design patterns for the given design problem (DPS-1). The “Colorblindness” and “NavigateToMap” categories were passed as input parameters to the “getSemanticmoduleResult” service that communicates with the micro-services, which considered Apache Jena, SPARQL queries, and Dandelion API, as presented in Figure 7. The response body of the “getSemanticmoduleResult” service was provided in JSON response format. An excerpt of the recommended design pattern list for the given design problem (DPS-1) with a description of their problems and solutions is shown in Figure 12.

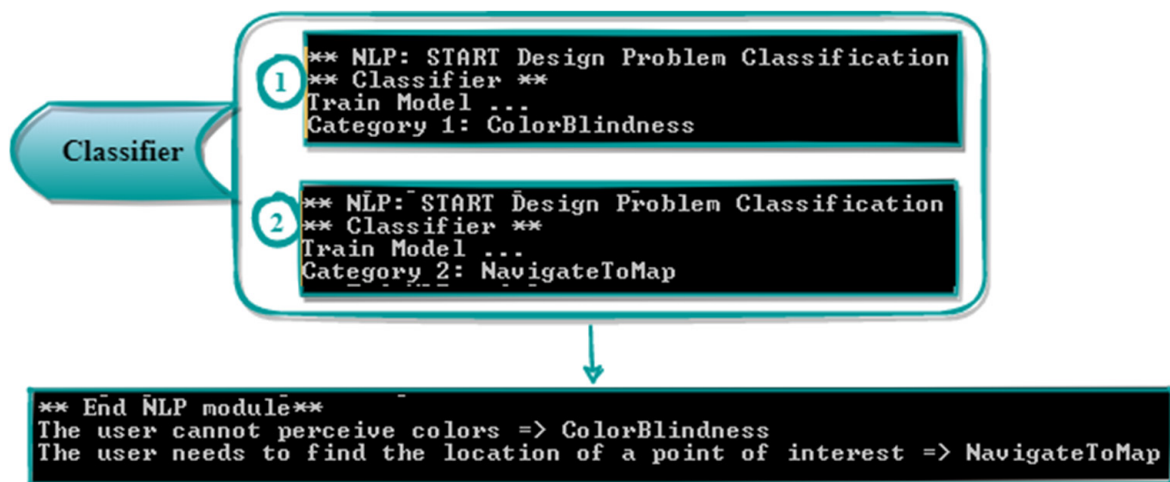


Figure 10. Classifier results for DPS-1.

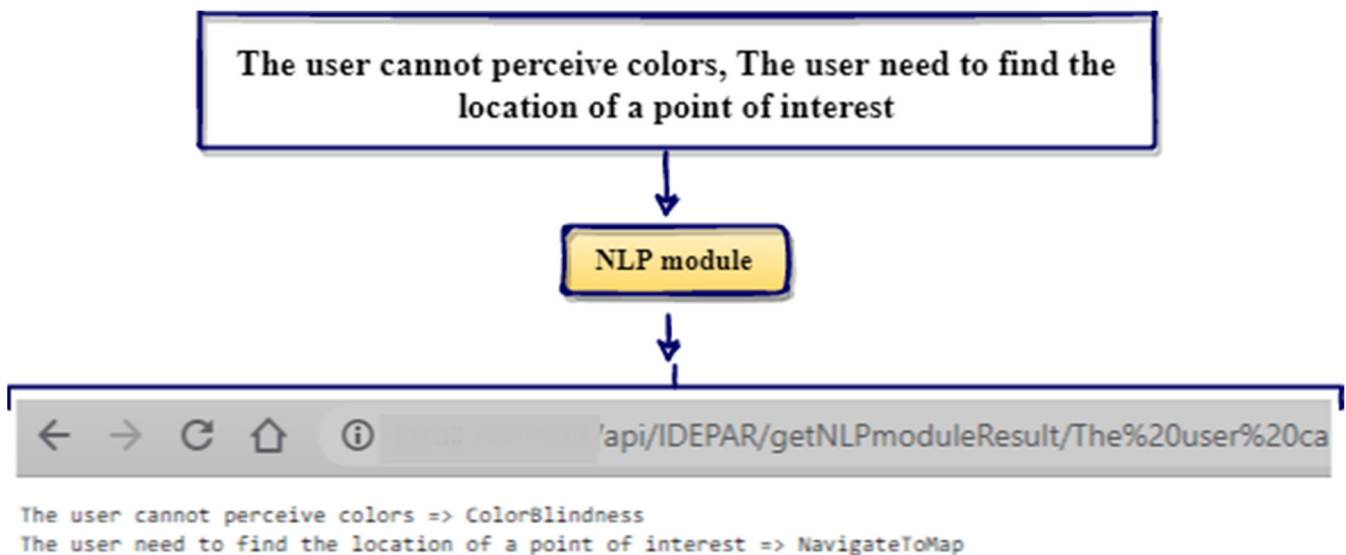


Figure 11. NLP module results for DPS-1.

4.3. Web Application Development

In order to process the design pattern recommendation requests received from developers and designers, we presented a Web application that communicates with the aforementioned REST Web services provided by the IDEPAR system. This application was developed using Spring Boot, Angular, and other technologies. Figure 13 illustrates the repository of HCI design patterns considered in the IDEPAR system.

In order to show the accomplishment of the IDEPAR system regarding various design problems, we considered the following two design problem scenarios in which keywords and text descriptions are considered, respectively.

- DPS-1: “The user cannot perceive colors, The user needs to find the location of a point of interest”.
- DPS-2: “LowVision Disability”.



Figure 12. Semantic module results for DPS-1.

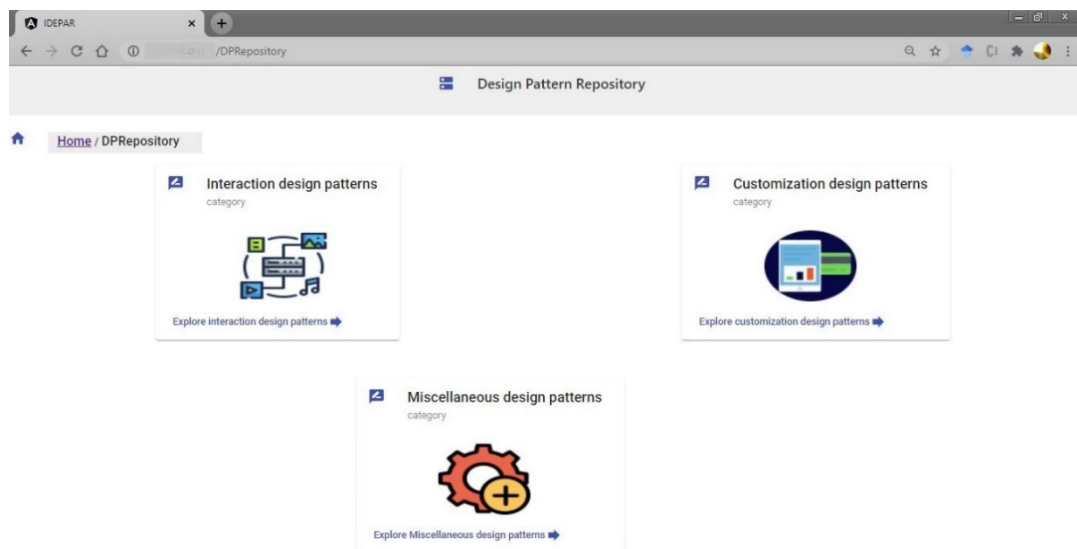


Figure 13. Design pattern repository interface.

First, to deal with the design problem (DPS-1), we present the obtained results in Figures 14 and 15. In particular, Figure 14 shows the selection of the text description relating to the first design problem (DPS-1) and Figure 15 illustrates the list of the recommended

HCI design patterns retrieved by the proposed IDEPAR system for the given design problem (DPS-1).

Second, to solve the design problem (DPS-2), Figure 16 outlines the interface for choosing DPS-2 using the user characteristic option, and Figure 17 presents the list of HCI design patterns that are recommended by the IDEPAR system to solve DPS-2. Each design pattern item is displayed with its name and problem. In this example, a list of four design patterns were recommended to solve DPS-2. As illustrated in Figure 17, by clicking on one of the recommended design patterns (e.g., FontSizeLarge) the present interface expanded the displayed item to show further information regarding the design pattern group and solution, as well as the following two actions: “choose design pattern” and “rate design pattern”.

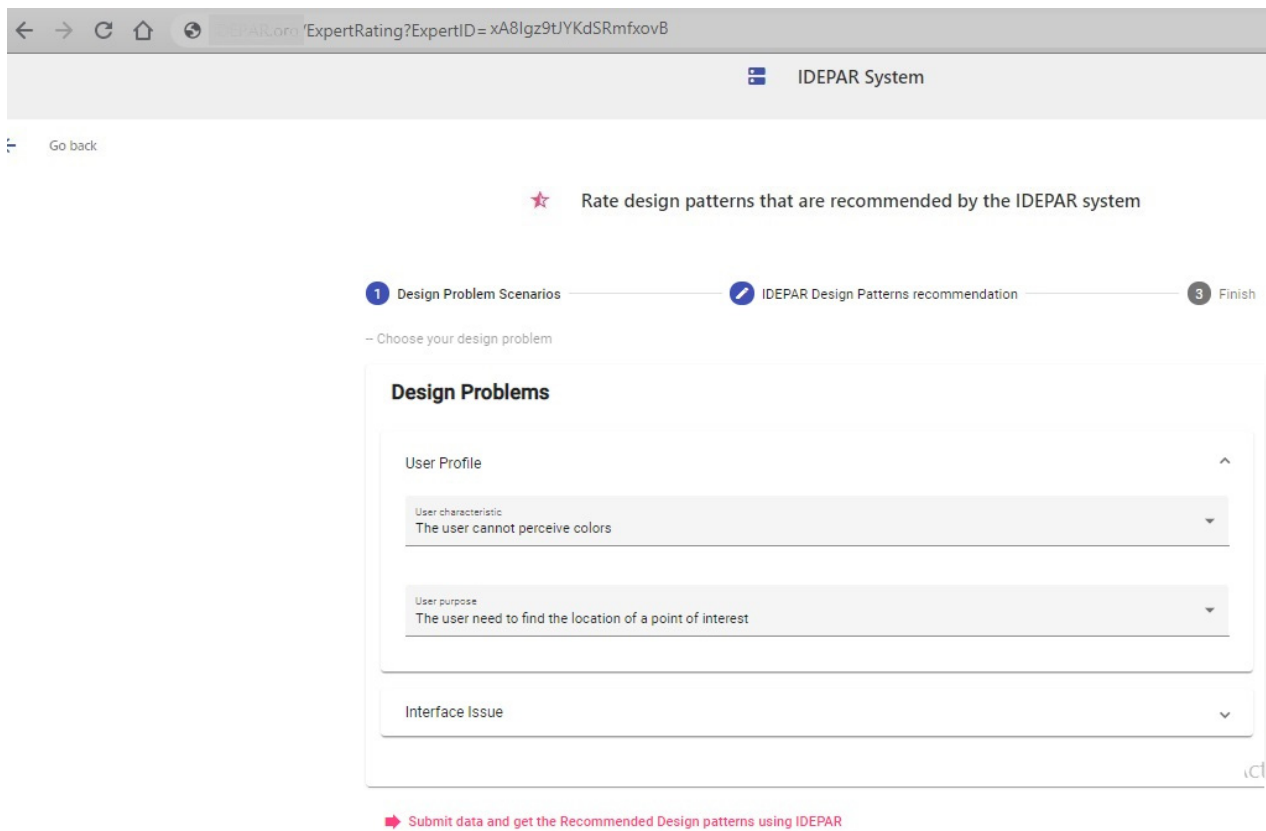


Figure 14. Selection of design problem DPS-1.

★ Rate design patterns that are recommended by the IDEPAR system

1 Design Problem Scenarios — 2 IDEPAR Design Patterns recommendation — 3 Finish

– Rate the relevance of the recommended design patterns

Nbr	DesignPatternName	DesignPatternProblem
1	ZoomIn	Different users need different zoom level
2	ZoomOut	Different users need different zoom level
3	MapNavigator	User needs to find a location of choice on a map
4	DarkBackgroundColor	User has difficulties perceiving background color
5	WhiteColor	User has difficulties perceiving images and text

< Back > Next

Figure 15. List of recommended design patterns for DPS-1.

★ Rate design patterns that are recommended by the IDEPAR system

1 Design Problem Scenarios — 2 IDEPAR Design Patterns recommendation — 3 Finish

– Choose your design problem

Design Problems

User Profile

User purpose

Interface Issue

Submit data and get the Recommended Design patterns using IDEPAR

Figure 16. Selection of design problem DPS-2.

★ Rate design patterns that are recommended by the IDEPAR system

1 Design Problem Scenarios 2 IDEPAR Design Patterns recommendation 3 Finish

-- Rate the relevance of the recommended design patterns

Nbr	DesignPatternName	DesignPatternProblem
1	FontSizeLarge	Different user need different font size for comfortable perception
2	FontSizeMedium	Different user need different font size for comfortable perception
3	ZoomIn	Different users need different zoom level
4	FontSizeSmall	Different user need different font size for comfortable perception

Design pattern Group: FontSize

Design pattern solution: Maximize Text font size and interaction element size for comfortable perception

Choose design pattern

Submit Rating

-- 1 = Not relevant at all -- 2 = To a small extent -- 3 = Somewhat -- 4 = To a large extent -- 5 = To a very large extent

< Back > Next

Figure 17. List of recommended design patterns for DPS-2.

5. Experimental Evaluation

We conducted an experimental study in order to achieve a comprehensive evaluation of the proposed IDEPAR system, which was designed to recommend HCI design patterns, along several relevant dimensions. To that purpose, we performed a user-centric evaluation study.

5.1. Hypotheses

The main objective of this evaluation was to figure out the impact of the recommendations on the participants' acceptance intention towards the proposed IDEPAR system. Therefore, the experiment was performed from a research perspective focused on recommending the most relevant HCI design patterns from participants' perspective interests in finding design patterns that fit with their design problem. In this context, two main research questions were formulated:

(RQ1): What is the participants' perceived experience of the IDEPAR system? To tackle this research question, we wanted to test the following hypothesis: H1 = Participants' perceived experience of the proposed system is positive.

(RQ2): What is the participants' perceived accuracy of the recommended HCI design pattern? In order to address this research question, we wanted to test the following hypothesis: H2 = Participants consider the recommended HCI design patterns as relevant and matching well with the given design problem.

5.2. Study Design

Users from different sources with a minimum experience in the HCI field were invited via mailing lists to participate in this experiment. Among the participants, 67% were female and 33% were male, with the majority being between ages 25 and 40 years (75%). Concern-

ing the participants' academic disciplines, this study was conducted on researchers (58%), software developers (33%), and computer science students (9%). After accepting the invitation, they were informed about the steps of the evaluation study. At first, they were given a guide describing how to use the IDEPAR system through a document. After that, they were asked to access the application developed to test the proposed recommender system.

5.3. Study Protocol

In order to verify the previously mentioned hypotheses, participants were asked to carry out two main tasks. The first task was to fill the pre-study questionnaire, while the second was focused on answering the post-test questionnaire. More specifically, the pre-study questionnaire was oriented towards gathering participants' information regarding their knowledge about recommender systems and their level of expertise with HCI design patterns. Concerning the post-test questionnaire, it was mainly aimed at evaluating the quality of user experience with the IDEPAR system and the relevance of the recommended design patterns. This questionnaire was prepared based on the ResQue framework, which is a well-known user-centric evaluation recommender system for assessing user's experience and their acceptance [27]. The ResQue framework provided a wide variety of question statements that were categorized into the following four layers:

- Perceived system quality: refers to questions that assess the participant's perception of the objective characteristic related to the recommender system.
- Belief: concerns a higher level of the participant's perception of the recommender system.
- Attitude: includes questions that assess the participant's overall feeling regarding the system.
- Behavioral intention: includes questions that assess the recommender system's capability to engage participants to use it regularly.

Questions that belong to these layers mainly address participants' perceived experiences of the recommender system and accuracy of design patterns. Indeed, these questions answered the two hypotheses (H1 and H2). From the questions provided by the ResQue questionnaire, 13 questions were considered in the post-test questionnaire. In this questionnaire, the five-point Likert scale (ranging from 1 to 5) was considered as the measurement scale used to assess the degree of participants' answers, with 1 signifying "strongly disagree" and 5 signifying "strongly agree". The selected questions and their categories are presented in Figure 18. The full version of the post-test questionnaire is available in Table A1 in Appendix A.

1. Perceived System Quality	2. Belief	4. Behavioral Intention
<p>Perceived Accuracy: Q1- The recommended design patterns are relevant and match the given design problem.</p> <p>Novelty: Q2- The design patterns recommended to me are novel. Q3- The IDEPAR system helped me discover new design patterns.</p> <p>Diversity: Q4- The recommended design patterns are diverse.</p> <p>Information Sufficiency: Q5- The information about the IDEPAR system are sufficient for me. Q6- The information provided for the recommended design patterns are sufficient for me.</p>	<p>Perceived Usefulness: Q7- The IDEPAR system helped me to find the relevant design patterns.</p> <p>Perceived Ease of Use: Q8- I became familiar with the IDEPAR system very quickly.</p>	<p>Use Intention: Q12- I will use the IDEPAR system again. Q13- I will tell my colleagues about this recommender.</p>
	3. Attitude	
	<p>Overall Satisfaction: Q9- Overall, I am satisfied with the IDEPAR system</p> <p>Confidence & Trust: Q10- I am convinced of the design patterns recommended to me. Q11- The IDEPAR system can be trusted</p>	

Figure 18. Representative Questions from each ResQue layer.

5.4. Statistical Analysis

In order to perform the statistical analysis of the data collected from the two questionnaires, we used IBM SPSS version 28.0 [28]. Descriptive analyses were substituted for all data. Particularly, measures of frequency (percent), central tendency (mean), and measures of dispersion (standard deviation) were used. In addition, the reliability of the post-test questionnaire's layers was assessed by Cronbach's alpha [29]. Finally, Pearson correlation was considered for identifying the correlation between the experience of the participants and their answers. For testing such a correlation, a p -value of ≤ 0.05 was considered to be statistically significant.

6. Results

6.1. Pre-Study Questionnaire Results

A total of 12 participants completed their tasks and were involved in the present experiment. The responses to the demographic data of the pre-study questionnaire were as follows: not familiar with recommender systems (16%), whereas the remaining participants possessed medium (42%) or high (42%) knowledge about recommender systems. Concerning the level of expertise with HCI design patterns, the majority of participants (more than 90%) had experience with HCI design patterns, wherein 8% were novice, 33% were intermediate, and 59% were advanced. Table 5 illustrates the descriptive statistics regarding the demographic data.

Table 5. Pre-study questionnaire results.

Questionnaire	Item	Percent
Knowledge of recommender systems	Low	16%
	Medium	42%
	High	42%
Level of expertise with HCI design patterns	Novice	8%
	Intermediate	33%
	Advanced	59%

6.2. Post-Test Questionnaire Results

The participants' results from the post-test questionnaires were collected and analyzed. We provide the descriptive statistics concerning the 13 questions of the post-test questionnaire in Table 6. Along with Cronbach alpha, mean, and standard deviation (SD) values, the distribution of answers for each question item was also calculated. Figures 19–22 show a divergent stacked bar that illustrates the distribution of answers provided by the participants to perceived system quality, belief, attitude, and behavioral intention layer, respectively.

Table 6. Post-test questionnaire results.

Layer	Question	Mean	SD
Perceived system quality	Q1	3.91	0.95
	Q2	3.41	1.25
	Q3	3.33	1.31
	Q4	4.33	0.74
	Q5	2.41	0.64
	Q6	2.41	0.64
Belief	Q7	3.91	0.95
	Q8	4.33	0.84
	Q9	4	1.08
Attitude	Q10	3.58	0.75
	Q11	3.83	0.68
Behavioral intention	Q12	3.33	0.74
	Q13	3.16	0.68

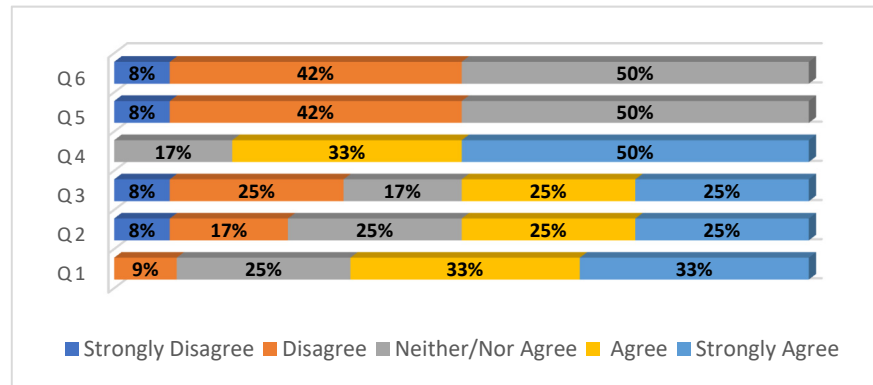


Figure 19. Distribution of answers to post-test questionnaire: perceived system quality layer.

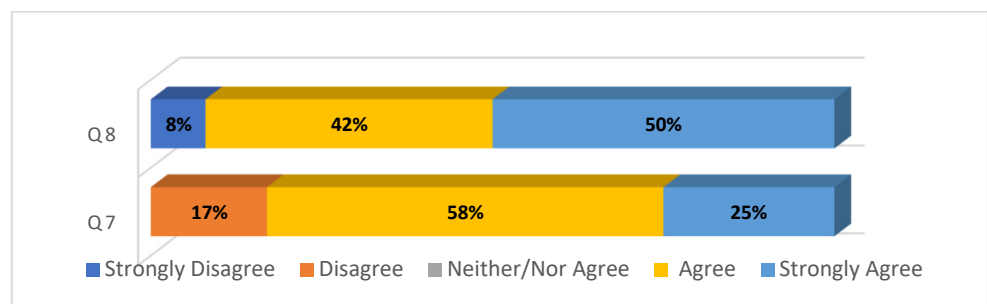


Figure 20. Distribution of answers to post-test questionnaire: belief layer.

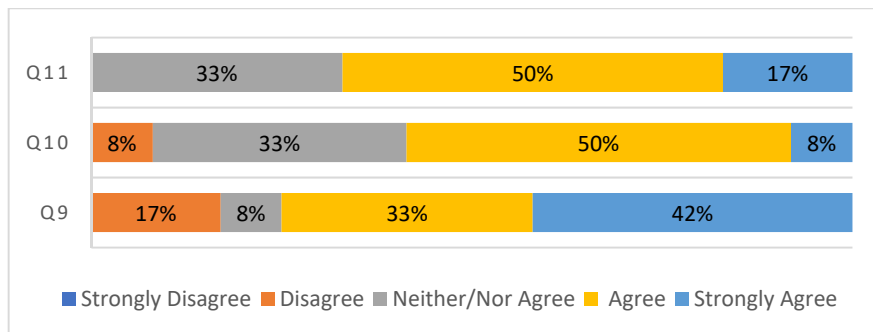


Figure 21. Distribution of answers to post-test questionnaire: attitude layer.

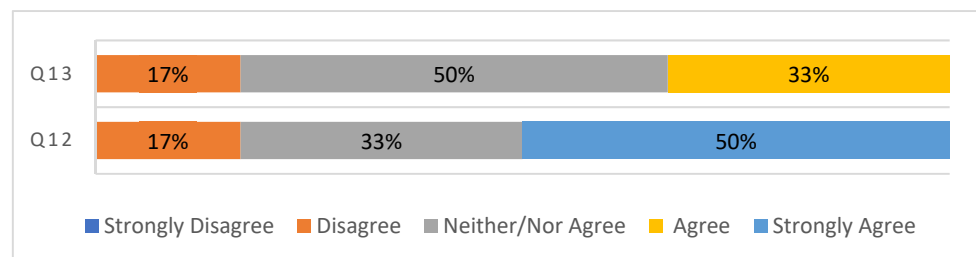


Figure 22. Distribution of answers to post-test questionnaire: behavior layer.

According to the results presented in Table 6, we observed that the mean value for many questions was above the median, with SD values below 1. More specifically, answers for the first question, with a mean value of to 3.91 (SD = 0.95), and the fourth question, with a mean value of 4.33 (SD = 0.74), reveal that participants believed that the IDEPAR system recommended relevant and diverse HCI design patterns. For the second and

third questions, roughly 50% of participants perceived the novelty of the recommended design patterns (Figure 19). Meanwhile, the information sufficiency questions, including Q5 and Q6, received the lowest scores. Among all participants, 42% disagreed and 8% strongly disagreed with Q6 and Q5. The mean value for these questions was equal to 2.41 (SD = 0.64). The results indicated that participants were not well-satisfied with the sufficiency of the information about the system (Q5) and the information provided for the recommended design patterns (Q6). Additionally, Figure 20 shows that a minority of participants were satisfied with the information sufficiency, in which 8% of participants did not agree and 42% disagreed. Moreover, the mean value for the belief layer was high, being equal to 4.12. The answers of this layer reveal that more than 90% of all participants agreed that “the IDEPAR system helped them to find the relevant design patterns”, involving 50% strongly agreeing for Q7. For Question 8, answers varied between 25% strongly agree and 58% agree. Thus, more than 80% of participants considered the proposed system as useful. In contrast, a minority of participants did not perceive it as useful. Concerning the attitude layer, participants’ overall satisfaction was high with a mean value equal to 4 (SD = 1.08), according to the answers of Q9. Among all participants, 60% were satisfied with the recommender system. Furthermore, the mean values for questions Q10 and Q11 were equal to 3.58 (SD = 0.75) and 3.83 (SD = 0.68), respectively. Finally, the mean value for the behavioral intention was equal to 3.24 (SD = 0.71), which reveals that participants found the proposed system moderately acceptable in terms of use intentions. More specifically, the mean values for Q12 and Q13 were equal to 3.33 (SD = 0.74) and 3.16 (SD = 0.68), respectively. Figure 22 shows that a minority of participants were satisfied with the use intention; their answers vary between 33% neither/nor agree and 17% disagree for Q12, and 50% neither/nor agree and 17% disagree for Q13.

In order to verify whether the internal consistency test provided reliable results or not, we considered Cronbach’s alpha criterion. This criterion has to meet a minimum threshold of 0.7 [30]. As presented in Table 7, the results of the measurements of Cronbach’s alpha met the required minimum threshold for perceived system quality, attitude, and behavioral intention layers, except for the belief layer.

Table 7. Cronbach’s alpha results of the post-test questionnaire layers.

Layer	Cronbach Alpha
Perceived system quality	0.78
Belief	0.61
Attitude	0.86
Behavioral intention	0.82

Moreover, we investigated the correlation between the participants’ expertise and answers based on the Pearson’s rank correlation coefficient. We relied on this coefficient as it provides values in the range from -1 to 1 , therefore it is suitable for detecting negative correlations. In Tables 8 and 9, we provide the correlations that we found. Table 8 illustrates a significant correlation between participants’ knowledge regarding recommender systems and the response of Q7 ($r = 0.873$, $p < 0.001$) and Q10 ($r = 0.711$, $p = 0.010$). Differently, Table 9 shows a significant correlation coefficient between participants’ expertise with design patterns and the answers of Q1 ($r = 0.080$, $p < 0.001$), Q7 ($r = 0.744$, $p = 0.005$), Q9 ($r = 0.598$, $p = 0.040$), and Q10 ($r = 0.595$, $p = 0.041$). Analysis of the obtained Pearson coefficient results revealed that participants with good knowledge of recommender systems and a high level of experience with design patterns found that the IDEPAR system was helpful for retrieving relevant HCI design patterns ($p < 0.001$; $p = 0.005$) and the recommended design patterns were convincing ($p = 0.005$; $p = 0.041$). Among participants who had high experience with HCI design patterns, the relevance of the recommended patterns ($p < 0.001$) and their satisfaction regarding the proposed system ($p = 0.040$) was confirmed.

Table 8. Correlations between participants' knowledge about recommender systems and answers of Q7 and Q10.

	Q7	Q10
Pearson correlation	0.873 **	0.711 **
Sig. (1-tailed)	<0.001	0.010

** Correlation is significant at the 0.01 level (1-tailed).

Table 9. Correlations between participants' level of expertise with HCI design patterns and answers of Q1, Q7, Q9, and Q10.

	Q1	Q7	Q9	Q10
Pearson correlation	0.880 **	0.744 **	0.598 *	0.595 *
Sig. (1-tailed)	<0.001	0.005	0.040	0.041

* Correlation is significant at the 0.01 level (1-tailed). ** Correlation is significant at the 0.05 level (1-tailed).

7. Discussion

In this section, we discuss the interpretation of the obtained results. Firstly, regarding the perceived system quality layer, the results reveal that the majority of participants (66%) confirmed that “the recommended design patterns are relevant and match with the given design problem”. Additionally, according to participants' responses to Q2 and Q3, 50% of participants agreed with the novelty of the IDEPAR system. Moreover, the overall mean of the belief layer was equal to 4.12, and thus exceed the “Agree” value. Indeed, participants generally believed that the IDEPAR system helped them to find relevant HCI design patterns and perceived the ease of use of the provided system. Furthermore, the mean value for the attitude layer was equal to 3.80 (SD = 0.83), which reveals overall satisfaction of the participants and a high trust of the IDEPAR system. Concerning the behavioral intention layer, results indicate that 50% of participants strongly agreed that they would use the IDEPAR system again, and 30% of them agreed that they would recommend the system to their colleagues. Overall, we observe that participants assigned relatively low rates, especially for the information sufficiency (Q5, Q6) and for the use intentions (Q12, Q13). These results may come from the difficulty of understanding the information provided by the system. Therefore, richer information regarding the recommended design patterns is needed. We consider this as a stimulus for the future enhancement of the proposed IDEPAR system.

Secondly, the reliability of items was conducted with Cronbach's alpha. The obtained alpha was about 0.78, 0.86, and 0.82 for all items of perceived system quality, attitude, and behavioral intention, which exceed the minimum threshold of 0.7. Indeed, the reliability was deemed good for all items, except for the belief items, for which it is considered acceptable.

After that, the Pearson coefficient was applied for the target to test the statistical significance of the correlation. In this sense, knowledge of recommender systems and level of expertise with HCI design patterns appeared to be positively correlated with the answers of perceived usefulness, confidence, and trust. In addition, the results of correlation analysis reveal that experience with HCI design patterns has a positive relationship with the perceived accuracy, as denoted with p -value < 0.001. Overall, correlation analysis indicated that several factors influence participant attitudes regarding perceived accuracy, perceived usefulness, satisfaction, confidence, and trust. The selected factors were mainly concerned about participants' knowledge about recommender systems and their level of experience with HCI design patterns.

In brief, the research and development of the presented IDEPAR system allow us to answer the two previously mentioned research questions, RQ1 and RQ2, and thus support the two hypotheses, H1 and H2. More specifically, the findings of the evaluation study reveal that (i) participants had a positive experience regarding IDEPAR system quality

(H1), and (ii) the participants' perceived accuracy of the recommended HCI design patterns was assessed positively (H2).

8. Conclusions

In this work, we have proposed the IDEPAR system, which is a hybrid recommender system aimed at recommending the most relevant HCI design patterns for a given design problem in order to help and assist developers find appropriate design patterns. The system combined two main recommendation techniques based on the use of semantic similarity along with ontology models. These ontology models were considered to offer semantics for design patterns and design problem representation. Moreover, we developed a Web application that communicates with the services provided by the proposed recommender system. This application was used to assess the IDEPAR system, along with a pre-study questionnaire and a post-test questionnaire. In order to validate our system, we conducted a user-centric evaluation experiment wherein participants were invited to fill both questionnaires. The evaluation outcomes illustrated that participants' perceived experiences of the system's quality were positive, and the recommended HCI design patterns are relevant and match well with the design problem. Nevertheless, further enhancement regarding the information provided on the system and on design patterns is needed in order to improve the proposed system regarding information sufficiency and behavioral intention. As part of future work, we will target our emphasis to enhance the proposed recommender system. We intend to take advantage of these insights obtained from the evaluation study and consider them for improving the presented system. We will also investigate the possibility to cover more complex design problems within the IDEPAR system that could be selected or presented as text descriptions entered by designers or developers. Furthermore, we plan to work on extending the approach considered in our system with a larger repository of HCI design patterns. Another interesting future work area would be to focus on a group assessment, wherein more experts in the HCI domain would be involved in the evaluation study to enhance the validation of the proposed recommender system. Finally, we intend to work on the ICGDEP system, which is the second system within the global AUIDP framework, to achieve the implementation of the design patterns recommended by the IDEPAR system and to evaluate the generated user interfaces with specific questionnaires.

Author Contributions: Conceptualization, Methodology, Formal Analysis, Investigation, Writing—Original Draft Preparation, Writing—Reviewing and Editing, Visualization, A.B.; Methodology, Investigation, Visualization, Supervision, M.K.; Investigation, Supervision, Visualization, F.B.; Visualization, F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Post-Test Questionnaire

The possible values for the score are, 1: Strongly disagree, 2: Disagree, 3: Neither/Nor Agree, 4: Agree, 5: Strongly agree.

Table A1. Post-test questionnaire.

Questions	1	2	3	4	5
Q1. The recommended design patterns are relevant and match the given design problem.					
Q2. The design patterns recommended to me are novel.					
Q3. The IDEPAR system helped me discover new design patterns.					
Q4. The recommended design patterns are diverse.					
Q5. The information about the IDEPAR system is sufficient for me.					
Q6. The information provided for the recommended design patterns is sufficient for me.					
Q7. The IDEPAR system helped me to find the relevant design patterns.					
Q8. I became familiar with the IDEPAR system very quickly.					
Q9. Overall, I am satisfied with the IDEPAR system.					
Q10. I am convinced of the design patterns recommended to me.					
Q11. The IDEPAR system can be trusted.					
Q12. I will use the IDEPAR system again.					
Q13. I will tell my colleagues about this recommender.					

References

- Ruiz, J.; Serral, E.; Snoeck, M. Evaluating user interface generation approaches: Model-based versus model-driven development. *Softw. Syst. Model.* **2018**, *18*, 2753–2776. [CrossRef]
- Gomaa, M.; Salah, A.; Rahman, S. Towards a better model based user interface development environment: A comprehensive survey. *Proc. MICS* **2015**, *5*. Available online: https://www.researchgate.net/profile/Syed-Rahman-5/publication/228644279_Towards_A_Better_Model_Based_User_Interface_Development_Environment_A_Comprehensive_Survey/links/551dbfc90cf213ef063e9ca9/Towards-A-Better-Model-Based-User-Interface-Development-Environment-A-Comprehensive-Survey.pdf (accessed on 6 September 2021).
- Letsu-Dake, E.; Ntuen, C.A. A conceptual model for designing adaptive human-computer interfaces using the living systems theory. *Syst. Res. Behav. Sci.* **2009**, *26*, 15–27. [CrossRef]
- Alexander, C. *A Pattern Language: Towns, Buildings, Construction*; Oxford University Press: New York, NY, USA, 1977.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Design patterns: Abstraction and reuse of object-oriented design. In *European Conference on Object-Oriented Programming*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 406–431.
- Coram, T.; Lee, J. Experiences—A pattern language for user interface design. In *Proceedings of the Joint Pattern Languages of Programs Conferences PLOP*, Monticello, IL, USA, 4–6 September 1996; Volume 96, pp. 1–16.
- Tidwell, J. *A Pattern Language for Human-Computer Interface Design*; Tech. Report WUCS-98-25; Washington University: Washington, DC, USA, 1998.
- Graham, I. *A Pattern Language for Web Usability*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2002.
- Borchers, J.O. A pattern approach to interaction design. In *Cognition, Communication and Interaction*; Springer: London, UK, 2008; pp. 114–1331.
- Van Duyne, D.K.; Landay, J.A.; Hong, J.I. *The Design of Sites: Patterns for Creating Winning Web Sites*; Prentice Hall Professional: Upper Saddle River, NJ, USA, 2007.
- Tidwell, J. *Designing Interfaces: Patterns for Effective Interaction Design*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2010.
- Patterns in Interaction Design. Available online: <http://www.welie.com/> (accessed on 6 September 2021).
- Konstan, J.A.; Riedl, J. Recommender systems: From algorithms to user experience. *User Model. User-Adapt. Interact.* **2012**, *22*, 101–123. [CrossRef]
- Sanyawong, N.; Nantajeewarawat, E. Design Pattern Recommendation: A Text Classification Approach. In *Proceedings of the 6th International Conference of Information and Communication Technology for Embedded System*, Hua Hin, Thailand, 22–24 March 2015.
- Gomes, P.; Pereira, F.C.; Paiva, P.; Seco, N.; Carreiro, P.; Ferreira, J.L.; Bento, C. Using CBR for automation of software design patterns. In *European Conference on Case-Based Reasoning*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 534–548.
- El Khoury, P.; Mokhtari, A.; Coquery, E.; Hacid, M.-S. An Ontological Interface for Software Developers to Select Security Patterns. In *Proceedings of the 19th International Workshop on Database and Expert Systems Applications (DEXA 2008)*, Turin, Italy, 1–5 September 2008; pp. 297–301.
- Hamdy, A.; Elsayed, M. Automatic Recommendation of Software Design Patterns: Text Retrieval Approach. *J. Softw.* **2018**, *13*, 260–268. [CrossRef]
- Hussain, S.; Keung, J.; Sohail, M.K.; Khan, A.A.; Ilahi, M. Automated framework for classification and selection of software design patterns. *Appl. Soft Comput.* **2019**, *75*, 1–20. [CrossRef]

19. Youssef, C.K.; Ahmed, F.M.; Hashem, H.M.; Talaat, V.E.; Shorim, N.; Ghanim, T. GQM-based Tree Model for Automatic Recommendation of Design Pattern Category. In Proceedings of the 2020 9th International Conference on Software and Information Engineering (ICSIE), Cairo, Egypt, 11–13 November 2020; ACM Press: New York, NY, USA, 2020; pp. 126–130.
20. Abdelhedi, K.; Bouassidar, N. An SOA Design Patterns Recommendation System Based on Ontology. In *International Conference on Intelligent Systems Design and Applications*; Springer: Cham, Switzerland, 2018; pp. 1020–1030.
21. Naghdipour, A.; Hasheminejad, S.M.H. Ontology-Based Design Pattern Selection. In Proceedings of the 2021 26th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, 3–4 March 2021; pp. 1–7.
22. Celikkan, U.; Bozoklar, D. A Consolidated Approach for Design Pattern Recommendation. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; pp. 1–6.
23. Mu, R.; Zeng, X. Collaborative Filtering Recommendation Algorithm Based on Knowledge Graph. *Math. Probl. Eng.* **2018**, *2018*, 9617410. [[CrossRef](#)]
24. Braham, A.; Khemaja, M.; Buendía, F.; Gargouri, F. UI Design Pattern Selection Process for the Development of Adaptive Apps. In Proceedings of the Thirteenth International Conference on Advances in Computer-Human Interactions ACHI, Valencia, Spain, 21–25 November 2020; pp. 21–27.
25. Braham, A.; Buendía, F.; Khemaja, M.; Gargouri, F. User interface design patterns and ontology models for adaptive mobile applications. *Pers. Ubiquitous Comput.* **2021**, *25*, 1–17. [[CrossRef](#)]
26. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; Fernández-López, M. The NeOn Methodology for Ontology Engineering. In *Ontology Engineering in a Networked World*; Springer: Berlin, Germany, 2012; pp. 9–34.
27. Pu, P.; Chen, L.; Hu, R. A user-centric evaluation framework for recommender systems. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 157–164.
28. IBM Corp. *IBM SPSS Statistics for Windows*; Version 28.0; IBM Corp: Armonk, NY, USA, 2021.
29. Jnr, B.A. A case-based reasoning recommender system for sustainable smart city development. *AI Soc.* **2021**, *36*, 159–183. [[CrossRef](#)]
30. Nunnally, J.C. *Psychometric Theory 3E*; Tata McGraw-Hill Education: New York, NY, USA, 1994.