



Predicting interaction design patterns for designing explicit interactions in ambient intelligence systems: a case study

Viridiana Silva-Rodríguez¹ · Sandra E. Nava-Muñoz¹ · Luis A. Castro² · Francisco E. Martínez-Pérez¹ · Héctor G. Pérez-González¹ · Francisco Torres-Reyes¹

Received: 21 May 2020 / Accepted: 15 December 2020 / Published online: 13 February 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature 2021

Abstract

Ambient intelligence (AmI) focuses on supporting people by designing sensitive and responsive environments to context through implicit and explicit interactions. Explicit interactions in AmI systems have requirements specific to making interactions robust, smooth, intuitive, and reliable. Based on requirements, the designers can detect and eliminate faults from the beginning of the design process and understand the users' needs and demands. This work presents a UIPatternM model for predicting interaction design patterns from processing text-based requirements through machine learning algorithms. We evaluate the predictions of our proposal. We also present a case study with professional designers who evaluated the UIPatternM recommender predictions according to a set of design-level requirements that emulate everyday needs. Our participants performed a set of tasks based on scenarios, and we evaluated the participants' using effectiveness, efficiency, and satisfaction as performance metrics. Applying the UIPatternM model helped to endorse the conception and refinement of user interface design for explicit interaction in AmI systems.

Keywords Interaction design patterns · Ambient intelligence systems · Design-level requirements · Natural language processing · Explicit interaction

1 Introduction

Ambient intelligence (AmI) focuses on supporting people by designing sensitive and responsive environments to context [15]. They integrate networked devices and can change the states in response to users' needs. Therefore, AmI has become a multidisciplinary research field that consists of several domains, such as ubiquitous computing, profiling, context awareness, and interaction design. Moreover, AmI addresses a broad range of topics and has various contexts of use [20, 24].

A typical context for AmI studies is the home location, though applications may also extend to public spaces like hospital environments or monitoring elders [5, 24]. However, the scope of AmI systems is not only research prototypes but also commercial products in everyday use, which confront us to the computational aspect of the environment that supports some form of intelligent interaction. According to [35], intelligent behavior involves four system elements: context awareness, personalization, adaptivity, and anticipatory behavior. In addition, AmI systems must initiate and maintain bidirectional communication with the user [40]. Thus, AmI systems success depends on technological expertise and user input to include implicit and explicit interactions [9, 20]. The classic interaction is of explicit nature. This takes place between users and devices, where the user tells the computer in a certain level of abstraction (e.g., direct manipulation using a user interface (UI), gesture, line commands, or speech input) what expects the computer to do [35]. On the other hand, implicit interaction includes those actions carried out by users without an explicit intention but that the system proactively captures and processes as inputs. Accordingly, in AmI systems user interfaces are typically multimodal, with more

✉ Viridiana Silva-Rodríguez
vsilva@alumnos.uaslp.edu.mx

Luis A. Castro
luis.castro@acm.org

¹ Autonomous University of San Luis Potosí,
San Luis Potosí, Mexico

² Sonora Institute of Technology (ITSON),
Ciudad Obregón, Mexico

than one interaction modality operating in either parallel or transitions between explicit and implicit interactions [20, 40].

Such complex systems require design and development approaches where particular aspects of AmI systems are specified. Thus, AmI systems design is influenced by user-centered methods that position the user at the center of the design activity, ask feedback through evaluations, execute tests to improve the design, and even co-create prototypes with a group of designers or users [2]. The user often enters and receives information that needs to be easily understood without prior professional or computational expertise [22].

AmI systems require techniques that are capable of ensuring that user context and requirements are integrated adequately into the entire design process from the very beginning. According to Byrne [9], data accountability, economics, and security issues in AmI systems have manifested in poor UI design, thus propagating a lack of continued user engagement. Even if the provided functionality is perceived as valuable, the user might reduce the usage time or stop using the offered AmI systems because they fail to support individual aspects. In order to provide adequate AmI solutions, it is necessary to place particular emphasis on obtaining, specifying, and documenting software requirements. These requirements are based on normative, social, and technical aspects and must be transferred into functional requirements for system development. According to Coronato et al. [16], the specification of requirements allows designers to detect and eliminate faults from the beginning of the design process.

AmI systems need to keep the humans in the loop and control, consequently empowering humans vs. autonomous impertunate behavior of AmI systems [35]. This will promote a psychological state of confidence in the users by increasing the understanding and control of the AmI systems interactions. Also, this will ensure that the system and users' interactions are designed to meet users' needs and requirements. Interaction design patterns offer an opportunity to keep users' needs and expectations in the first design stages [7], and also represent a method for presenting the case and solutions in a structured way. In addition, interaction design patterns also help merge concepts from different areas of study such as human factors, engineering, and interaction design, which facilitate their acceptance [27]. Thus, this set of interaction design patterns includes what questions the design should answer, its limits, and the principles used for the implementation stage.

There are recurring situations in many AmI application development projects, for which interaction design patterns based on requirements seem to be a suitable solution [12, 33]. However, the literature about integrating both requirements and interaction design patterns is scarce.

Some questions remain unanswered: How do you select the required interaction design patterns? And, how do you evaluate the process of using interaction design patterns? The UIs design is a process that requires creativity and skills, but they are highly subjective and error-prone. Thus, integrating both requirements and interaction design patterns has not been sufficiently highlighted in practical contexts.

In this paper, we present a model for inferring interaction design patterns from processing text-based requirements (called UIPatternM) through machine learning (ML) algorithms. Integrating both requirements and interaction design patterns will allow designers to save time by analyzing text requirements and support design decisions. The proposal consists of the UIPatternM model validation by professional interaction designers to determine its feasibility as a case study.

In Section 2, related work is discussed to lay the foundations of the proposed model (UIPatternM). In Section 3, the process of searching for suitable interaction design patterns for specific requirements is explained in detail, which is our main contribution. Section 4 describes the results of the evaluation method for predictions issued by the classification algorithms. Section 5 describes the evaluation of the perception of the UIPatternM model application through a recommender tool prototype, while Section 6 presents the results of evaluations. Section 7 presents the discussion. Finally, Section 8 presents the conclusion and future work, and then we present our acknowledgements.

2 Related work

AmI systems have evolved over time. They have kept a specific scheme like the architecture for the development of AmI systems shown in [26]. This architecture suggests including artificial intelligence methods and techniques and integrating them with the systems' operational part. An important matter is the systems' functional part because it must consider an implicit and explicit interaction [30]. In this direction, it has been suggested that AmI systems must be of a multimodal nature based on context model acquisition [20]. This supposes a problem when designing such systems. Then, we must know how to integrate data from different inputs and generate relevant outputs for designers and developers to interpret and implement requirements. Another promising research is presented in [10], in which they present a conceptual framework structured along with three layers in a system and presented an information space of events, conditions, and actions, linked together in Event-Condition-Action rules and operating according to the interconnection metaphor.

They proposed a set of tools focused on a list of interactions that considers events and conditions. Both research [10, 20] are relevant proposals. However, they are focused on the end user, not on designers.

With AmI systems' design in mind, we discuss several design methods and tools, which provide support in the early design phases and integrate interaction design patterns. Thus, we uncover their strengths and limitations following their application to design explicit interaction in the AmI approach.

2.1 Model-based user interface development (MBUID) tools

Given the heterogeneity of current interactive systems, approaches such as model-based user interface development (MBUID) help reduce the gap between requirements and implementation. To achieve AmI systems design that is congruent with requirements, a set of models (e.g., Tasks, Abstract UI, Concrete UI, Final UI) and their relationships are needed. The Task and Domain models correspond to the hierarchies of tasks that need to be performed in a specific temporal logical order for achieving users' goals through the interaction with the UIs [11]. However, the designer must enter specific metadata to relate the information contained in the model-based system.

Several model-based tools have been developed to present possible solutions [19, 32, 36, 38]. Pattern-driven and model-based user interfaces (called PD-MBUI) framework [32] proposes to integrate the interaction design patterns in the design process. This approach's starting point is a clear description of the users' requirements, which includes answers to questions such as the following: Who are the users? What are their tasks? In what environment will the system be used? The answers to these questions are essential to building models that reflect users and their real needs in any scenario. However, this approach considers a specific structure for the interaction design patterns that are defined through an ontology. PD-MBUI infers interaction design pattern information at the request of the user through topic selection.

2.2 Interaction design patterns

Researchers of the AmI approach have focused their efforts on new techniques to improve systems and products that include the human factor. Vegas-Barbas et al. [39] propose to consider these cognitive features in the early stages of smart spaces' design by defining a set of interaction design patterns. Then, these interaction design patterns were validated by end users through an informal discussion. As a result, end users concluded that patterns were adequate to cover the needs of the design of smart spaces.

Another interesting work is that of Chung et al. [13], who developed an initial and emerging interaction pattern language for ubiquitous computing. This consists of 45 pre-patterns describing application genres, physical-virtual spaces, interaction and systems techniques for managing privacy, and methods for fluid interactions. They recruited 16 pairs of designers to validate the effectiveness of the 45 pre-patterns. Pre-patterns helped new and experienced designers unfamiliar with ubiquitous computing to generate and communicate ideas and avoid design problems early in the design process. However, there is no reference to the model or tool to implement both sets of interaction design patterns for AmI systems design.

In order to support design decision making, which can be regarded as subjective and prone to errors, the approach presented in this paper consists of the construction of a recommendation system based on a requirements classification model to support designers in the process of designing interactive systems, through the inference of the related interaction design patterns in a set of requirements. Instead of relying on manually crafted rules, text classification with machine learning learns to make classifications based on past observations. While we show prediction accuracy in the context of graphical UI design, it is possible to extend our framework through a greater variety of requirements and interaction design patterns, including AmI contexts and others. Therefore, it is possible to apply it in other application scenarios as long as there is a database that can be exploited.

The works identified and described in the previous paragraphs are significant but isolated efforts. It is necessary to analyze what is needed and design an appropriate interactive system to be able to specify the selection of interaction design patterns. The interaction design patterns derived from requirements are useful to bridge gaps in the early phase of system development, where recurring requirements call for similar solutions. In the following section, a proposal for a model of interaction design patterns recommendation is described.

3 Prediction of interaction design patterns using text-based requirements

Interaction design patterns offer the opportunity to maintain users' needs and expectations in the early stages of design while allowing unifying concepts from different areas of study such as human factors, engineering, and interaction design to guarantee an optimal level of acceptance [1]. The needs are specified in a natural language. Thus, requirements need to be written correctly to understand them and obtain all the AmI systems' design information. Each requirement must accurately describe a capability that

meets some stakeholders' needs and must clearly explain the functionality to be built.

Based on the above, in this paper, we present a model to predict interaction design patterns from functional requirements in textual format. The relevant information is extracted regarding the tasks that the AmI systems must perform. The model called UIPatternM (see Fig. 1) arises from our previous work [34], in which a series of metadata is considered in the design decision making by interactive system designers. The design-level requirements are expressed in human-readable natural language as plain text. However, natural language processing (NLP) is a knowledge discovery approach to automatically extract the elements of concerns from raw plain text documents.

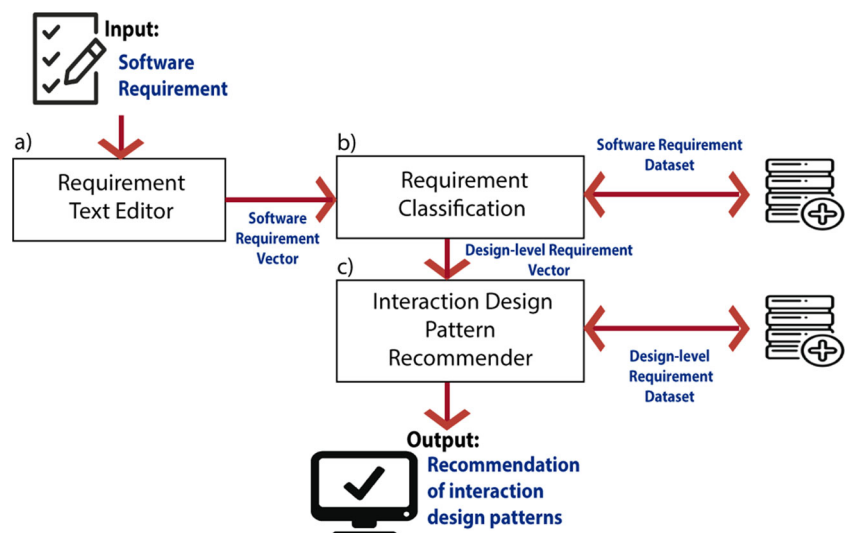
3.1 UIPatternM model

The UIPatternM model consists of three modules that require entries of data and generate output data. Following Fig. 1, the *Requirement Text Editor* module is the point of entrance in which the verification of textual requirement is carried out regarding the integrity of the metadata (e.g., the user, action, objects) and the grammatical structure to apply a text processing. Considering the advantages of NLP and formal representations, the proposal includes text verification based on a semantic parsing task proposed by [18]. For instance, given the requirement “The system will indicate the current context of the caregivers, which includes: identity, location, and activity; in support of decision making.”, we found that “system,” “current context of the caregivers,” “identity, location and activity,” and “to indicate” were specified as information elements. The structure of these information elements is similar to a grammatically adequate sentence.

Later, once a requirement text structure has been validated, in the *Requirement Classification* module, the requirement text is classified according to the type of requirement (functional or non-functional). The text classification consists of a trained, supervised learning algorithm with a requirements dataset associated with a classification class (a type of requirement). To train the *Classification Algorithm* it is necessary to transform the requirement text into a vector of numbers. Next, the vector representations were explored as Term Frequency-Inverse Document Frequency (TF-IDF) weighted vectors. This technique provides a score that represents the term frequency (TF) multiplied by the inverse document frequency (IDF) [21, 28]. The TF-IDF weighting is commonly used in text extraction and information retrieval to evaluate the importance of a linguistic term (usually unigram or bigram) in a dataset. The importance of the term increases concerning the frequency of this in the whole text. However, it is compensated with the frequency of the term in the domain of interest (e.g., frequent words like “the” or “for” will be reduced). Later, it generates a prediction of interaction design patterns associated with said input requirement.

Finally, in the *Interaction Design Pattern Recommender* module, the requirement text is classifying according to the type of interaction design pattern. This module uses a content-based filtering technique, whose objective is to tag design-level requirements using specific keywords to understand what actions will be performed in a UI for explicit interactions. Subsequently, those keywords are searched in the dataset and generate as an output, and interaction design patterns prediction that addresses similar design problems is predicted. In this case, the example prediction is the “Dashboard” interaction design pattern,

Fig. 1 UIPatternM: interaction design patterns recommendation model based on requirements



which is established as the following problem: “The user wants to digest data from multiple sources at a glance.” Furthermore, the model will be able to feedback with each prediction.

Based on a design-level requirements dataset, the recommender model of interaction design patterns is trained. The knowledge base is formed with a design level requirements of the AmI systems projects previously designed and labeled according to an interaction design pattern collection. In this work, we used machine learning algorithms for text classification and predicted an interaction design pattern for supporting design-related decision-making. We evaluated the algorithm with some performance measures. Likewise, an evaluation was designed to determine the feasibility of the concept presented through of perception of professional designers.

4 Evaluating the classification algorithms of UIPatternM model

For the inference of interaction design patterns from the requirements text, a set of data requirements were established, for which texts were collected from various sources such as the PROMISE corpus dataset [14], which is the most popular dataset of software requirements. These requirements describe the actors, objects, and resources of the system over which they act, as well as the user operations carried out (transitive or non-transitive). Subsequently, these requirements were classified based on the interaction design patterns of the Toxboes’ collection [37] (see Table 1).

The requirements dataset proposed in this work contains 307 requirements related to interaction design patterns: table filter (26), dashboard (20), search filters (22), sort by column (24), morphing controls (46), structured format (65), notifications (24), forgiving format (42), and input prompt (38).

4.1 Experiments

To perform the classification of a recommender model of interaction design pattern, we transformed the requirement text into a vector of numbers. The vector representations were explored as Term TF-IDF weighted vectors [21, 28]. The text extraction and information retrieval uses TF-IDF weighting to evaluate the importance of a linguistic term (usually unigram or bigram) in a dataset.

Once these representations of text vectors are generated, it was possible to train supervised classifiers with design-level requirements and predict their interaction design pattern label. There are several algorithms to perform training for this type of problem. The proposal uses the

75-25 modality (that is, 75% of the design-level requirements dataset for training and 25% for testing).

Once the requirements text classification model has executed, it was compared to four machine learning algorithms: logistic regression (LR), multinomial naive Bayes (MNB), linear support vector machine (LSVM), and random forest (RF) [3, 25]. We used a 10-fold cross-validation for the four classification models. Also, accuracy, precision, and recall are presented as classification performance measures.

4.2 Results

After training the four algorithms, the LSVM model showed a slight advantage with a mean accuracy of around 45% (see Table 2).

From this result with the LSVM model, the confusion matrix was generated (see Table 3), in which the discrepancies between the predicted and real labels can be observed. A vast majority of predictions end in the diagonal (predicted

Table 1 A sample set of interaction design patterns found in the Toxboes’ collection [37]

Interaction design pattern	Problem
Table filter	The user needs to categorically filter the data displayed in tables by the columns.
Dashboard	The user wants to digest data from multiple sources at a glance.
Sort by column	The user needs to be able to sort the data in a table according to the values of a column.
Morphing controls	The user wants only to be presented with controls available to the current mode.
Search filters	The user needs to conduct a search using contextual filters that narrow the search results.
Structured format	The user needs to enter data into the system quickly but the format of the data must adhere to a predefined structure.
Notifications	The user wants to be informed about essential updates and messages.
Forgiving format	The user needs to enter data into the system quickly, which then, in turn, interprets the users’ input.
Input prompt	The user needs to enter data into the system.

Table 2 Overall performance of classification algorithms

Algorithm	Accuracy	Precision	Recall	F1-score
LR	0.40	0.53	0.53	0.49
MNV	0.43	0.43	0.49	0.44
LSVM	0.45	0.58	0.54	0.52
RF	0.33	0.55	0.39	0.33

equal to real). However, a series of erroneous classifications are shown, which are requirements that affect more than one interaction design pattern.

5 Evaluating the UIPatternM model application with professional designers

This section describes a study with interactive systems professional designers, to obtain use perception of the UIPatternM model application through a recommender tool prototype. Then, we presented details on how we conducted the evaluation, the results, and its limitations.

5.1 Design of the study

As previously described, the UIPatternM model generates, as an output, interaction design pattern prediction from design-level requirement textual specification. The proposal is based on the training of a text classifier using classification algorithms. To obtain the point of view of professional designers in the context of AmI systems design, we designed a four-stage evaluation to determine the feasibility of the concept presented.

This study had a within-subjects design since all the treatments were applied to all the participants. The first part consisted of a training session about the nine interaction design patterns. All the participants were presented with

the interaction design of Toxboes' collection. Next, the participants had to select an appropriate interaction design pattern for ten textual requirements related to implicit interactions. The participants were free to choose the interaction design pattern, and after giving your answer, they indicated their perception through the three last questions using a 5-point Likert scale.

The participants had to validate ten design-level requirements previously pre-processed and classified with the UIPatternM model in the stage of interaction design pattern recommendation. After showing the recommender's prediction of interaction design patterns, the participants answered three questions to get their perception. In the last stage, the participants had to answer the Technology Acceptance Model (TAM) [17] and the System Usability Scale (SUS) [8] questionnaires. The whole study's objective was to analyze the ease of use and usefulness of UIPatternM model implementation in a recommender tool prototype.

The prototype of the recommendation tool that was presented to the participants consisted of a low-fidelity prototype. The main functions of the prototype were to enter the textual requirement, request the prediction, and display predictions of the recommendation tool. In order to show the prototype to the designers, we used a presentation of a use scenario.

5.1.1 Objective

The following objectives guided the evaluation:

- The perceived usefulness of the recommender model (UIPatternM) application to support design decisions in AmI systems explicit interaction design.
- The perceived effectiveness of the interaction design patterns recommendation based on the UIPatternM model to support design decisions in AmI systems explicit interaction design.

Table 3 Confusion matrix of LSVM algorithm in the requirements classification by interaction design pattern (TF Table Filter, DB Dashboard, SByC Sort By Column, MC Morphing Controls, SFilters Search Filters, SFormat Structured Format, Notif Notifications, FFormat Forgiving Format, IPrompt Input Prompt)

Actual	Predicted								
	TF	DB	SByC	MC	SFilters	SFormat	Notif	FFormat	IPrompt
TF	2	0	1	2	0	0	0	0	0
DB	3	1	0	1	0	0	0	0	0
SByC	1	0	2	1	1	0	0	2	0
MC	1	0	0	12	0	4	0	0	1
SFilters	2	0	2	2	1	1	0	0	1
SFormat	0	0	0	2	0	20	1	0	0
Notif	0	0	0	2	0	1	4	1	0
FFormat	0	0	0	2	0	2	0	6	0
IPrompt	0	0	0	3	0	2	0	3	5

5.1.2 Hypothesis formulation

In AmI, one of the challenges is to design explicit interactions such as interfaces that require minimal attention and cognitive effort of users to understand the information presented, and also explicit needs to be perceived as the aesthetic and naturally integrated into their environment. However, the use of interaction design patterns is centered on designing interactive systems for the web or mobile systems, and the selection of interaction design patterns is subjective and error-prone. Therefore, by using a recommender tool based on the UIPatternM model, it is possible to obtain interaction design patterns recommendations related to users' needs. As a consequence, and to frame the study, we established the following hypotheses:

- H1: The opinion of the designers related to interaction design pattern recommendation is considered good.
- H2: The selection of interaction design patterns by designers concurs with the prediction issued by the interaction design pattern recommendation model.
- H3: The recommender tool based on the UIPatternM model is useful to support interactive systems design (perceived usefulness).
- H4: The recommender tool based on the UIPatternM model is effective in identifying interaction design patterns (perceived effectiveness).

5.1.3 Participants

In order to guarantee the participation of interaction designers, a call was published. The interested designers answered a selection questionnaire to participate in the study. The participants were 14 professional designers from different companies in Mexico and Ecuador. All of them (14) were participating in interactive design projects for diverse areas (educational, healthcare, Internet of Things, administration in general, etc.). Their average age was 32 years old, with 5 years of experience on average. All of them had concluded their bachelor degrees related to human-computer interaction areas.

5.1.4 Instruments

This section presents the instruments developed to conduct this study.

1. Use scenario. This scenario was about an interactive systems designer working with a multidisciplinary team on the project (tourist services system). Based on the requirements collected, the designer had to generate interaction design proposals. Nevertheless, the designer faced the confusion of identifying which interaction design patterns to implement in his design,

and consequently had to consult the recommender tool to obtain a interaction design pattern prediction based on knowledge of previous projects.

2. Interaction design patterns presentation. Each participant had to review nine interaction design patterns of Toxboes' collection.
3. Questionnaire to select interaction design patterns. Each participant had to analyze ten design-level requirements and select the appropriate interaction design pattern. A questionnaire was prepared using a 5-point Likert scale and focused on the understanding of the design-level requirement.
4. Questionnaire to validate interaction design pattern predictions. Each participant had to review ten predictions of interaction design patterns related to ten different design-level requirements, to predict the output of the recommender tool based on the UIPatternM model. A questionnaire was prepared using a 5-point Likert scale and focused on the interaction design pattern prediction by design-level requirement.
5. SUS questionnaire. A questionnaire was prepared based on the SUS [8] using a 5-point Likert scale and focused on the recommender tool based on the UIPatternM model.
6. TAM questionnaire. A questionnaire was prepared based on the TAM [17] using a 7-point Likert scale and focused on the recommender tool based on the UIPatternM model.

5.2 Procedure

The experiment took place in four sessions with a total duration of 18 h. However, not all participants performed the test on the same day. The session stages are explained below.

- Introduction (duration ≈ 5 min). The participants were given an overall explanation of the study, along with their objectives.
- Use scenario (duration ≈ 5 min). The participants were put into context through a use scenario. Furthermore, the use of an interaction design pattern recommendation tool was shown.
- Interaction design patterns (duration ≈ 10 min). The participants received a short presentation session regarding identify the interaction design patterns of Toxboes' collection.
- Ten design-level requirements (duration ≈ 20 min). Each designer was asked to analyze ten design-level requirements and select an interaction design pattern for each one. Subsequently, the participants were asked to indicate how well they agreed with the understanding of the design-level requirement to select an interaction design pattern.

- Predictions of interaction design patterns (duration ≈ 15 min). The corresponding interaction design pattern predictions of ten design-level requirements were shown. Each designer was asked to answer a questionnaire to indicate how well they agreed with each recommendation generated by the recommendation tool.
- After the participants had finished evaluating interaction design patterns predictions, they answered the SUS and TAM questionnaires (duration ≈ 5 min).

As a result of the case study, a set of qualitative and quantitative data were obtained. These data were analyzed and the results obtained are presented below.

6 Use perception of the UIPatternM model by professional designers

The results obtained are presented in three parts: (1) the results of the interaction design pattern selection by professional designers, (2) the results of the evaluation by professional designers concerning UIPatternM recommender predictions, and (3) the results of participants' perceptions regarding the UIPatternM model application through a recommender tool prototype. The data obtained were analyzed the Minitab tool version 19.1.1 for Windows [23] to analyze the data obtained from the questionnaires.

6.1 Selection of interaction design patterns by professional designers

The interaction design patterns recommendation precision was determined by validating the interaction design patterns selected by participants and comparing them with the predictions of the UIPatternM recommender, to obtain the number of valid and invalid selections per participant.

In order to analyze the data, the participants were divided into two groups: junior designers (1 to 4 years

of experience) and senior designers (6 or more years of experience). Figure 2 shows that senior designers are the professionals that mostly selected interaction design patterns similar to those predicted by the UIPatternM recommender. The above is because senior designers selected similar interaction design patterns in five of ten design-level requirements (50%). On the other hand, the junior designers only selected similar interaction design patterns in two of ten design-level requirements (20%). This fact could mean that design decisions depend on the participants' experience.

6.2 Perception of recommended interaction design patterns

The participants evaluated recommendations of interaction design patterns for ten different design-level requirements those mentioned in the previous section. In order to obtain the participants' perceptions, a 5-point Likert scale questionnaire was used (1 = completely disagree; 5 = completely agree). The statistic used to compare whether there was a significant difference between means in the samples of each requirement is the *t*-Student. The *t* test calculates a *p* value based on the performance data to decide if the participants agreed with the recommendation of an interaction design pattern. If the *p* value is small enough (*p* value < 0.05) in most ten design-level requirements, it is concluded that the difference is significant. Table 4 shows the participants agreed with most of the recommendations predicted by the UIPatternM recommender (7 of 10).

6.3 Perception of usability

We used the SUS questionnaire and first six items of the TAM inventory to obtain the usability perception of the recommender tool. We obtained the scores from all the SUS questionnaires and transformed them according to the

Fig. 2 Percentage by participants' group (junior and senior designers) that selected similar interaction design patterns to the UIPatternM recommender

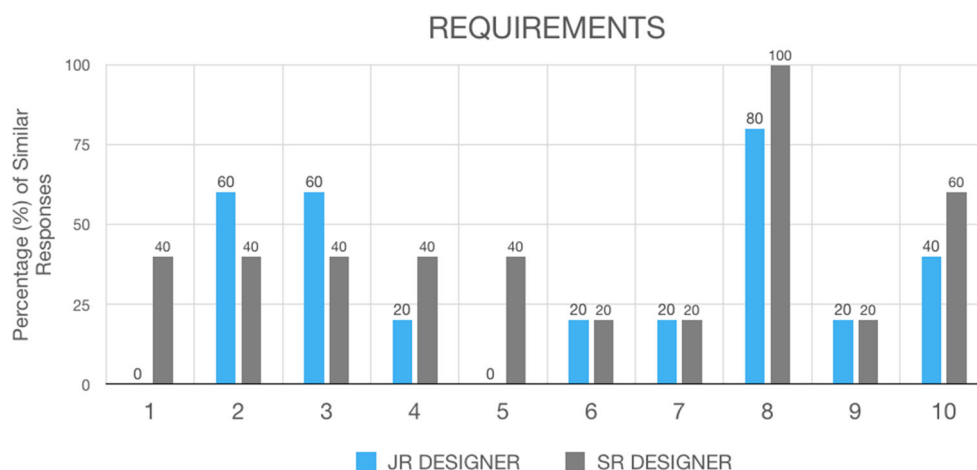


Table 4 Evaluation of UIPatternM recommender outputs by professional designers

Design-level requirement	Mean	StDev	Median	<i>p</i> value <i>t</i> -Student
Requirement 1	4.214	1.122	4.500	0.001
Requirement 2	4.571	0.514	5.000	0.000
Requirement 3	4.857	0.363	5.000	0.000
Requirement 4	4.214	0.893	4.000	0.000
Requirement 5	4.143	0.663	4.000	0.000
Requirement 6	3.429	1.505	4.000	0.153
Requirement 7	3.714	1.541	4.000	0.053
Requirement 8	4.429	1.453	5.000	0.001
Requirement 9	2.929	1.439	3.000	0.572
Requirement 10	4.143	1.167	4.500	0.001

curved grading scale [29]. The recommender of interaction design patterns obtained an averaged SUS score of 77 (mean = 78 = B+, StDev. = 15), corresponding to a B grade, and represents a good usability perception [4]. Likewise, the TAM questionnaire was applied in parallel to verify with six of its items the level of usability of the recommender tool. The TAM questionnaire uses a 7 point Likert scale (1 = completely disagree; 7 = completely agree), so it is considered an acceptable level of usability in a range of 25 to 42 (see Table 5). The means for the TAM questionnaire in its usability factor (first six items) indicate an acceptable favorable trend for the recommender tool (mean = 6, StDev. = 0.663).

In order to validate the ease of use, the data from the last six questions of the TAM questionnaire were obtained. The TAM questionnaire uses a 7-point Likert scale (1 = completely disagree; 7 = completely agree), so it is considered an acceptable level of usability in a range of 25 to 42 (see Table 5). The means for the TAM questionnaire in its ease of use factor indicate an acceptable trend for the recommender tool (mean = 5.530, StDev. = 0.730).

7 Discussion

In this section, the results from testing each hypothesis are analyzed and the implications are discussed.

7.1 The precision of classification algorithms

For the evaluation of the recommender module of the interaction design patterns, a comparison was carried out with four classification algorithms: logistic regression, multinomial naive Bayes, linear support vector machine, random forest, and performance metrics. The results obtained determine that the linear vector support machine method is the most appropriate with 58% of precision. However, the challenge consisted of establishing a knowledge base of design-level requirements. There is no evidence of datasets related to requirements for the Aml system design presented in the literature. For this reason, we created a dataset that currently is smaller than other proposals but it contributes to the body of works in this area.

Table 5 Results of the TAM questionnaire

Question	Mean
Usefulness	
Q1: Using the system would enable me to accomplish tasks more quickly	5.631
Q2: Using the system would improve my performance	5.722
Q3: Using the system would make it easier to do my work chores	6.021
Q4: Using the system would enhance my effectiveness at work	5.814
Q5: Using the system display would increase my productivity	5.247
Q6: I would find the system useful in my work	5.902
Usefulness, mean	5.907
Ease of use	
Q7: I find it easy to get the system to do what I want it to do	5.468
Q8: My interaction with the system is clear and understandable	5.710
Q9: Learning to operate the system would be easy for me	5.702
Q10: It would be easy for me to become skillful at using the system	5.211
Q11: I would find the system easy to use	5.495
Q12: I would find the system flexible to interact with	5.609
Ease of use, mean	5.530

7.2 The feasibility of a tool for recommendations of interaction design patterns

Regarding the H1 hypothesis, the opinion of the designers related to the recommendation of an interaction design pattern is considered acceptable. The hypothesis of $\mu > 3$ was accepted in seven of the ten recommendations for interaction design patterns based on the analysis of means performed. The possible reason for this is that the designers analyze the design-level requirement from the recommendation provided by the recommender tool, delimiting the number of interaction design patterns. However, the designers mentioned the importance of having the justification for the recommendation, to be confident that it is the correct interaction design pattern. To illustrate:

What I need is something that is backed up by data, in this case, the system tells me if this works, this does not work. Why? Where is the data? Where is the study behind these suggestions? How the system is making the selection, and how I can be sure that the system is making the correct selection. [Participant 4]

On the other hand, the H2 hypothesis was slightly accepted, although the results reflect that the participants selected interaction design patterns subjectively according to their interpretation of the design-level requirements. The participants with more years of experience (senior designer) carried out a more profound analysis (50% similarity with UIPatternM) than the participants with fewer years of experience (junior designer). However, in the design-level requirements where the grammar structure was simple, both senior and junior designers managed to identify the interaction design patterns similar to those predicted by the recommender tool.

In the case of the H3 hypothesis, which stated that the recommender tool based on the UIPatternM model is useful to support interactive systems design (perceived usefulness) was accepted, this means that the designers considered a UIPatternM recommender tool is easy to use, according to SUS and TAM results. The designers expressed that the requirements are provided in a textual way by a member of the project team. Therefore, it is not a complicated activity for them to enter design-level requirements into a recommender tool:

I have my data in my folders and I keep saving it; by having something like this is like having some type of library, then it would be easier to apply the process. It is easier, faster, and I think more efficient. I do not have to go find my files.... [Participant 12]

Finally, the H4 hypothesis, which stated that the recommender tool based on the UIPatternM model is effective to identify interaction design patterns (perceived

effectiveness), was accepted. The results indicate that the recommender tool is effective to support interactive design decisions, in comparison to doing a subjective process. The proposal was to apply the UIPattern model to design a recommender tool that would support design decisions in AmI systems' explicit interaction design. The evaluation results provide evidence indicating that they could obtain the recommendation of interaction design patterns congruent; and therefore, the recommender tool would be useful and well accepted in the context of interaction design.

In this regard, one of the participants commented: "If you are a rookie and have no experience, the system will teach you ... perhaps you will understand a little more how the behavior of the system is, by using the system... besides, it helps to improve your times, it will teach you if you are a rookie, to use tools correctly, based on best practices. Probably, based on the projects that have been done before, my approach as someone who has a little more experience, even if I still have my own five years of experience, I always look for related projects on the Internet; I see what they use and I try to analyze based on requirements... what works for me, what doesn't work for me." [Participant 7]

8 Conclusions and future work

The UI design is paramount within the AmI approach as its success influences future system usability with transitions between explicit and implicit interactions. AmI systems that adhere to UI interaction design patterns have the power to be an effective technological assistant, providing a balance between user-initiated interactions and unsolicited system intrusions [30]. For this reason, this research work describes a model for the recommendation of interaction design patterns based on requirements for UIs design for explicit interaction in AmI systems.

Explicit interactions are necessary to design a set of physical interactions that include the environment and communication, among other characteristics. The use of interaction design patterns confirms compliance with the principles of usability and user experience, thus promoting compliance with standards and the use of best practices. In specific contexts and users, it is necessary to comply with specific requirements when explicit interactions require direct manipulation in the UIs [20]. Interaction design patterns allow compliance with these requirements. This together with the concept of interaction provides us a useful way to describe solutions to everyday usability or accessibility problems in a specific context [6, 31]. The success of applying interaction design patterns in other design approaches has favored their inclusion in the AmI interaction scenario, where the understanding and control of

the user over the system become of utmost importance to ensure final acceptance.

However, it is a challenge to find the right interaction design pattern when the requirement is ambiguous, or the context of use is not specified. Due to these problems, this work proposes an UIPatternM recommender model to initially formulate grammatically correct requirements so that they can be processed and related to interaction design patterns. Thus, they have to be translated, as well as classified according to their functionality to associate them with the corresponding interaction design pattern, according to a knowledge base.

In order to evaluate our proposals, a case study was performed. According to our evaluation results, we obtained that the UIPatternM recommender supports the professional designers in design decisions making in comparison to how this is done by traditional search. The proposed tool helps professional designers obtain recommender of interaction design patterns based on the text of design-level requirements and a knowledge base of AmI systems design. Therefore, the reuse of design solutions is encouraged through the use of interaction design patterns. Consequently, the interactive system design in AmI with UIPatternM recommender could be more productive and less subjective and subject to error. The professional designers who participated in the case study know the methods and acceptable practices of interaction design. Therefore, the designers' perceptions provided this work with a real vision of current projects, as well as the feasibility of our proposal to adapt to the different design methodologies under which they currently work. As a future work, we envision the generation of a more robust dataset, with a bigger number of attributes to be able to classify the requirements with cross-platform interaction design patterns to AmI systems, and also to perform more experiments in order to obtain different results with which to validate this proposal. Thereby, we will be able to improve the prototype and test it in the design of real AmI systems.

Acknowledgements We thank the National Council for Science and Technology (CONACYT) in Mexico for its support with grant No. 246970.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Al-Samarraie H, Ahmad Y (2016) Use of design patterns according to hand dominance in a mobile user interface. *J Educ Comput Res* 54(6):769–792
- Bai A, Mork HC, Halbach T, Fuglerud KS, Leister W, Schulz T (2016) A review of universal design in ambient intelligence environments. *Smart Accessibility* 6–11
- Bali M, Gore D (2015) A survey on text classification with different types of classification methods. *Int J Innov Res Comput Commun Eng* 3:4888–4894
- Bangor A, Kortum PT, Miller JT (2008) An empirical evaluation of the system usability scale. *Intl J Human-Comput Interact* 24(6):574–594
- Bastaki BB, Bosakowski T, Benkhelifa E (2017) Intelligent assisted living framework for monitoring elders. In: 2017 IEEE/ACS 14th international conference on computer systems and applications (AICCSA), pp 495–500
- Borchers JO (2000) Interaction design patterns: Twelve theses. In: Workshop, the Hague, vol. 2, p. 3. Citeseer
- Bradley M, Kristensson PO, Langdon P, Clarkson PJ (2018) Interaction patterns: The key to unlocking digital exclusion assessment? In: International conference on applied human factors and ergonomics, pp 564–572. Springer
- Brooke J (1996) SUS-A quick and dirty usability scale. Usability evaluation in industry. CRC Press, Boca Raton. <https://www.crcpress.com/product/isbn/9780748404605>. ISBN: 9780748404605
- Byrne C, Collier R, O'Grady M, O'Hare GMPStreitz N, Markopoulos P (eds) (2016) User interface design for ambient assisted living systems. Springer International Publishing, Cham
- Cabitz F, Fogli D, Lanzilotti R, Piccinno A (2017) Rule-based tools for the configuration of ambient intelligence systems: A comparative user study. *Multimed Tools Appl* 76(4):5221–5241
- Calvary G, Coutaz J (2014) Introduction to model-based user interfaces. *Group Note* 7 W3C
- Castillo NG, Pérez JL, Gómez-Sanz JJ (2018) A computational approach to improve the gathering of ambient assisted living requirements. In: Multidisciplinary digital publishing institute proceedings, vol. 2, p. 1246
- Chung ES, Hong JI, Lin J, Prabaker MK, Landay JA, Liu AL (2004) Development and evaluation of emerging design patterns for ubiquitous computing. In: Proceedings of the 5th conference on Designing interactive systems: Processes, practices, methods, and techniques, pp 233–242
- Cleland-Huang J, Mazrouee S, Liguó H, Port D (2007) nfr. <https://doi.org/10.5281/zenodo.268542>
- Cook DJ, Augusto JC, Jakkula VR (2009) Ambient intelligence: Technologies, applications, and opportunities. *Pervas Mob Comput* 5(4):277–298
- Coronato A, Paragliola G (2017) A structured approach for the designing of safe aal applications. *Expert Syst Appl* 85:1–13
- Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quart* 319–340
- Diamantopoulos T, Roth M, Symeonidis A, Klein E (2017) Software requirements as an application domain for natural language processing. *Lang Resour Eval* 51(2):495–524
- Engel J, Martin C., Forbrig P (2015) A concerted model-driven and pattern-based framework for developing user interfaces of interactive ubiquitous applications. In: LMIS@ EICS, pp 35–41
- Gallardo J, Bravo C, Molina AI (2018) A framework for the descriptive specification of awareness support in multimodal user interfaces for collaborative activities. *J Multimodal User Interf* 12(2):145–159. <https://doi.org/10.1007/s12193-017-0255-x>
- Jones KS (2004) A statistical interpretation of term specificity and its application in retrieval. *J Document*
- Mandl T, Kirisci PT, Thoben KD (2018) A method for designing physical user interfaces for intelligent production environments. *Adv Human-Comput Interact* 2018:6487070. <https://doi.org/10.1155/2018/6487070>
- Minitab L (2019) Minitab, Inc., versã,o 19.1.1

24. Mohamed MAB, Elmahdy HN (2017) Enhancing the life quality of elderly using ambient intelligent technology (amit). *Egypt Comput Sci J* 41(3)
25. Pranckevičius T, Marcinkevičius V (2017) Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic J Modern Comput* 5(2):221
26. Ramos C (2009) An architecture for ambient intelligent environments. In: Corchado JM, Tapia DI, Bravo J (eds) 3rd symposium of ubiquitous computing and ambient intelligence 2008. Springer, Berlin, pp 30–38
27. Riehle D, Züllighoven H (1996) Understanding and using patterns in software development. *Tapos* 2(1):3–13
28. Salton G, Yang CS (1973) On the specification of term values in automatic indexing. *J Document* 29(4):351–372
29. Sauro J, Lewis JR (2016) Quantifying the user experience: Practical statistics for user research. Morgan Kaufmann, Burlington
30. Schmidt A (2000) Implicit human computer interaction through context. *Pers Technol* 4(2):191–199. <https://doi.org/10.1007/BF01324126>
31. Seffah A (2015) Patterns of HCI design and HCI design of patterns: Bridging HCI design and Model-Driven software engineering. Springer
32. Seffah A, Taleb M (2012) Tracing the evolution of hci patterns as an interaction design tool. *Innov Syst Softw Eng* 8(2):93–109
33. Serral E, Sernani P, Dragoni AF, Dalpiaz F (2017) Contextual requirements prioritization and its application to smart homes. In: European conference on ambient intelligence, pp 94–109. Springer
34. Silva-Rodríguez V, Nava-Muñoz SE, Martínez-Pérez FE, Pérez-González HG (2018) How to select the appropriate pattern of human-computer interaction?: A case study with junior programmers. In: 2018 6Th international conference in software engineering research and innovation (CONISOFT), pp 66–71. IEEE
35. Streitz N, Charitos D, Kaptein M, Böhlen M. (2019) Grand challenges for ambient intelligence and implications for design contexts and smart societies. *J Ambient Intell Smart Environ* 11(1):87–107
36. Thanh-Diane N, Vanderdonck J, Seffah A (2016) Uiplml: Pattern-based engineering of user interfaces of multi-platform systems. In: Research challenges in information science (RCIS), 2016 IEEE tenth international conference on, pp 1–12. IEEE
37. Toxboe A (2015) User interface design patterns. <http://ui-patterns.com/>. Online; Accessed 29 Jan 2020
38. Vanderdonck J, Simarro FM (2010) Generative pattern-based design of user interfaces. In: Proceedings of the 1st international workshop on pattern-driven engineering of interactive computing systems, pp 12–19. ACM
39. Vega-Barbas M, Pau I, Augusto JC, Seoane F (2017) Interaction patterns for smart spaces: A confident interaction design solution for pervasive sensitive iot services. *IEEE Access* 6:1126–1136
40. Waddell TF, Zhang B, Sundar SS (2015) Human-computer interaction. *Int Encycloped Interperson Commun* 1–9

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.