

# Migrating Legacy Web applications

Viviana Cajas · Matias Urbietta · Gustavo Rossi · Yves Rybarczyk

Received: date / Accepted: date

**Abstract** Applications are a ubiquitous tool when running on mobile devices become an opportunity for a new digital business model. However, enterprises implemented the core business in legacy applications that are not designed to run properly on mobile devices. Consequently, this migration involves time and cost. The main contribution of this study to these limitations is an approach to assist this migration based on Markov chains that is a predictive user behaviour method that allows that available functionality is prioritized, the experience of User Interface is re-thought, and the new design is evaluated with a preliminary running version of the legacy app. An experiment on ten participants shows that a dynamic user interface that implements the Markov model provides the users with better usability, evidenced with an improvement of productivity labour and the Nielsen heuristics comparison between versions. This work suggests this model makes possible a smart and proactive interface design.

**Keywords** IFML · legacy adaptation · Markov chains · mobile devices

---

Viviana Cajas  
Facultad de Ciencias Administrativas y Económicas,  
Universidad Tecnológica Indoamérica, Quito, Ecuador  
E-mail: vivianacajas@uti.edu.ec

Matias Urbietta  
Lifia, Fac. de Informática, UNLP, Argentina  
CONICET, Bs As, Argentina  
E-mail: murbieta@lifia.info.unlp.edu.ar

Gustavo Rossi  
Lifia, Fac. de Informática, UNLP, Argentina  
CONICET, Bs As, Argentina  
E-mail: gustavo@lifia.info.unlp.edu.ar

Yves Rybarczyk  
Faculty of Informatics and Data Sciences,  
Dalarna University, Falun, Sweden  
E-mail: yry@du.se

## 1 Introduction

In the last decade, the world has witnessed a fast adoption of new technology from mobile devices to smart TVs and Refrigerators between other electronic equipment that are interconnected and running applications; that is the internet of things (IoT) [1]. In most of the cases, the Web platform has been used for developing sophisticated user interfaces like games or advance movie players deployed in these devices. The arriving of a new device family introduces a huge challenge to engineers who have to accompany the rapid pace of the application evolution. The new application version should be adapted to run on top of a well-known platform such as the browser or even developed for a new unforeseen platform like smartwatch. In other words, they need to adjust a solution to the available channels which means providing different user experience because of the device restrictions. For example, moving from a desktop Web app to a mobile version likely reduce the available features because of the mobile viewport, and moving from a mobile version to a smartwatch version restricts it even more and also changes the navigation metaphor. To make matters worse, the continuous delivery strategy seems to state the applications in beta state: new interface features are quickly introduced, tested and then consolidated or discarded. The diversity of users requires a customized application, adapted to each individual profile and providing the best experience. Indeed, operators that provide field service spend their working time on the road or at customer or client place [2]. Starting from this premise, the companies that have not adapted their business models to mobile devices run the risk of disappearing in the medium term. A business model [3] is supported by a system that is designed to meet perceived market needs and thereby create value for all stakeholders involved: customers, strategic partners, suppliers, and the employees. The new business models can be implemented through mobile

applications [4] in order to make the business reachable to a wider range of business users and customers who mostly access Web applications from their mobile devices at any-time and anywhere. There are business applications [5] that were created before the development of mobile devices that are not responsive to different viewports. For example, these are applications so called legacies because there is no a clear evolution roadmap for them which were not designed to be mobile-friendly and it is not feasible to update it because of cost or technical limitations. This phenomenon repeats with the adoption of a new device like Smartwatch or the evolution of a new one like refrigerators that are providing IoT features. Legacy application usability problems have been studied in different works [6] pointing out the usability issues that arise when the application run in a mobile device. Some of them are: the legibility of the typography, the icons, the size, orientation of the screen, etc. Also, usability on mobile applications suggests that aesthetics graphics (balance between the colors, shapes, language, music or animation) is an important concept when evaluating them all [7].

Consequently, due to legacies are limited to work on Desktop web browsers providing poor usability while being accessed by other devices like smartphones. The main problems presented related to mobile viewports such as non-responsive content and improper navigation detected by the Google Mobile-Friendly Test [8] tool are: i) the graphics window has not been defined; ii) the text is too small to read or too big exceeding the viewport limits; iii) the clickable elements are too close to each other. Another usability problems with legacies: (i) The absence of a responsive design causes messages and other controls to be partially rendered; (ii) It is necessary to scroll and zoom several times to find a widget; (iii) Widgets are located in order alphabetic for this reason the search time is directly proportional to the initial letter of the name widget.

Cajas et al. [9][10] present a systematic mapping and a systematic literature review about the portability of legacies through a migration with different strategies. The authors found many alternatives to manage the migration of a Web application [9] to a RIA (Rich Internet Application): (i) The Document Object Model (DOM) is modified, so that the design adjusts to the size of the screens, providing an improvement in the web appearance; (ii) Model-driven development facilitates the generation of code semi-automatically from models; (iii) Mashups and (iv) Middleware require high know how in order to make merge APIs and separate data sources into a single integrated interface; (v) Augmentation technique, where the user is part of the process and can select certain customization of the applications for their convenience.

In [11] the authors conducted an experiment assessing the productivity and performance of users accessing both a legacy and a mobile-friendly version of an application.

The results of a controlled experiment based on two sites showed usability issues requiring more scrolling and zoom in/out events compared to the mobile-friendly version. Another study, [12] has collected usability experiments from 10 popular applications, where 3,575 users rated the usability using the System Usability Scale (SUS) questionnaire [13]. The average SUS rating was 77.7 out of 100, which is comparable to a C grade in the university grading scale [14]. It suggests that although mobile applications are largely used, they are not perfectly usable. Unfortunately, due to the lack of awareness of these issues, the users are forced to interact with applications even if they have poor usability. To tackle the problem, the industry tries to improve usability by increasing the width of the devices.

However, many well-known applications, such as Amazon.com, eBay.com and CNN.com, have used a radically different strategy. Instead of undergoing a thorough migrating process yielding a complete new version of them, they were subject to a smooth evolutionary approach. They started changing limited parts of their sites, introducing rich interface functionalities in a step-by-step way, evaluating them with customers and enriching the application seamlessly. In some cases, the enhancements were discarded after some time and the old style preferred.

An evolutionary approach was formalized by the authors in previous work by using the concept of Web Model Refactoring (WMR) [15]. A WMR is a change applied on the navigation or interface model of a Web application, aimed at improving its external quality while preserving the applications behavior. This paper focuses on WMRs over the applications graphical interface design that may transform a legacy Web application into a richer one, running on a new device family. We call them Rich Internet Application (RIA) refactorings. RIA refactorings are described as compositions of RIA interfaces [16], which in turn are formally specified with Interaction Flow Modelling Language (IFML).

This work is an extension of a previous study [17] [11], proposing an approach for designing the adaptation of a legacy application to be run in a new device family. The design of a new version relies on the use of Markov chains to analyse what application features should be prioritized and provided in a new version before migration, or portabilization, from legacy. It is a tool created by the Russian mathematician Andrei Markov in 1907 that mixes algebraic and statistical principles to analyze stochastic processes (that evolve over time in a set of states). It is an algorithm used in many mainly business and science areas, but in recent years it has gained importance in technology, primarily in hardware, networks and later in software. For this reason, beyond the patterns, it is difficult to discover what the next state will be in the process under study. In this project, the states of the Markov chain are formed with the screens and the controls that from here on will be generalized with the name

of widgets. This model helps to understand the mechanics behind navigation. Markov chains calculation is a stochastic process where the probabilities describe how the process will evolve in the future. These chains correspond to a standard mathematical technique recognized for the modeling process from the definitions of Cook and Wolf models [18], and for the development of Whittaker and Poore test cases [19]. Once the features have been filtered by means of the Systematic Layout Planning and the Markov chains, we use IFML to design the new user experience of the application. In this case, we first apply a set of rules for transforming the legacy UI to a new one that fits properly in the device. Next we design the application navigation using standard IFML elements considering the Markov suggestion about the widget order.

The main contributions of our approach are the following:

- The presentation of a novel formal approach to help to prioritize requirements that must be released in the new application.
- The formalization of the refactoring of user interfaces using IFML.
- The introduction of Web augmentation tool for instrumenting fast prototyping that can be used to assess the contribution of the portabilization.

The remaining of the article is organized into six sections. Section 2 defines the Background. Related work is described in section 3. Section 4 explains the proposed approach. In section 5 we present a running example providing a prototype and evaluation. Finally, the conclusions and future work are contemplated in section 6.

## 2 Background

### 2.1 Markov Chains

Markov Chains are a type of dynamic Bayesian networks [20] that predict the state of a system at a given time from the preceding state. The most important elements for the establishment of the chain are: the state space, the transition matrix, and the initial distribution [21]. The nodes (circles) represent the possible states, while the arrows show the transitions (the option to return to the same state is also considered). A transition probability defines the probability to move from one state to another. Successful algorithms use Markov chains, such as PageRank [22], created by Google, which allows granting a numerical value to each web page and, from it, establishes the order in which they appear after a search. It proves that this approach is suitable for establishing web browsing models.

**Table 1** Closeness or Proximity Matrix Scale

Key	Closeness	Value
A	Absolutely necessary	5
E	Especially important	4
I	Important	3
O	Ordinary	2
U	Without importance	1
X	Undesirable	0

### 2.2 Systematic Layout Planning

Systematic Layout Planning (SLP), proposed by Muther [23], allows the organization of the manufacturing plant improving productivity in the operations management. The improvement is achieved through the reduction of movements, according to the flow of information, materials handling, and the closeness of physical locations (i.e. workshops or offices). In order to represent the relationships existing, a relational table of activities or areas is used, called closeness matrix. It is usual to express these needs through a code of letters, following a scale based on the order of the five vowels, as shown in Table 1. This work makes an analogy between the software application and a manufacturing plant, which corresponds to space within the screen. Then, the distribution of elements in which users navigate must be optimal with respect to this matrix. Also, the workflow and the distribution of the processes must be taken into account. The efficient use of space within the screen is ensured according to the principle of optimal flow of the project.

### 2.3 IFML and User Interface transformation

In most mature Web design approaches [24], such as the UML-based Web Engineering (UWE), the Web Modeling Language (WebML), the Ubiquitous Web Applications (UWA), Hera, the Object-Oriented Web Solutions (OOWS) or the Object Oriented Hypermedia Design Method (OOHDM), a Web application is designed with an iterative process comprising -at least- the conceptual and navigational modeling. According to the state of the art on model-driven Web engineering techniques [25], these methods produce an implementation-independent model that can be later mapped to diverse runtime platforms. For the sake of clarity, the attention will concentrate on the IFML approach.

In 2013, the Object Management Group (OMG) [26] adopted the Interaction Flow Modeling Language (IFML) [27] as the standard approach to describe the interaction features of Web -and other kind of interactive - applications, making it part of the set of Software modeling standards like the Unified Modeling Language (UML) or the Business Process Modeling Notation (BPMN). The reader would find

a complete description of the language elements, examples and a metamodel description in the IFML site [27].

IFML implements Model-View-Controller (MVC) metaphor [28] and inherits the simplicity and expressibility of its parent WebML. Additionally, the language supports the same kind of extension mechanisms that UML already possesses (e.g., stereotypes or tagged values) and it is itself described by a clearly defined meta-model.

### 3 Related work

The main related studies are introduced in this section. The works demonstrate that Markov chains can be a useful tool in software engineering. Nevertheless, the potential of such a technique is still underutilized in the migration processes of the web to mobile applications.

Thimbleby et al. [21], propose a method based on the Markov models to integrate into the design stage, proving that these models are suitable when dealing with states and transitions systems. The goal of this project was a proposal for devices with buttons (mobile, vending machines, recorders), which can be represented as finite state machines. Another advantage is the fact that advanced knowledge of these models and mathematics is not required. It is a scalable model, since it works the same in large or small state machines, producing significant statistics and taking into account the errors that users can make, elections or landslides. The approach can be applied to abstract designs, prototypes, and animations, or fully operational systems. In this work, the Markov chains are used as an individual probabilistic state to improve the usability of controls but it does not consider migration or portabilization Issues.

Mao et al. [29] state that the use of the web can be accurately described by the Markov chain, which has proved to be an ideal model for statistical software tests, as well as being suitable for modeling user navigation in the Web. The authors propose an extended model (EMM) to develop plans and test methods from the components to the analysis of the system, the results of the test can gradually improve the EMM to perform regression tests and can also be used to correct programs, being a semiautomatic framework. Even though this model gets user navigation, it is not used to prioritize controls and design a better user interface.

Yanchun et al. [30] obtain the behavior model of a Mashup application, through a discrete Markov chain to build the scheme of the activities prior to develop the security risk model. The reason is that a web information system consists mainly of functional components described in terms of behaviors, interfaces and the interconnections between components. Here, the technique provides the prioritization of activities, but this information is not used to improve the interface.

Chohan et al. [31] use a hierarchical Markov chain to perform the software product line tests. This type of model manages to identify all the possible consequences in the usability of a family of systems. Three models are implemented: (i) captures the potential behavior of the products; (ii) keeps recording the functions with the similarities or variations between the products and (iii) makes a specification of the mapping of elements in the test transitions. Similarly to Mao et al. [29], this model is used strictly to application security and is out of migration topic.

Nwobi-Okoye et al. [32] use a Markov chain to model the stochastic process of software performance. The theoretical model of games and the concept of flow theory are used to model and obtain an optimal user interface with the balance in the competing user interfaces, recommending then to use prototyping techniques. The authors carry out three case studies for the evaluation of applications, a web application, a desktop application and the comparison of mobile interfaces, leaving Windows Phone 7 at a disadvantage with respect to Android and iPhone. Likewise, this work proposes Markov chains to make comparison about the interfaces between different platforms and how these features can be used to generate a migration process. However, these approaches, can be used to migrate an application to a mobile version but not to fully support the portabilization problem presented in this work. The contribution of our approach that addresses these existing limitations is the use of Markov chains to obtain in the first instance the behavior model of the web application that allows generating a more efficient adaptation based on usability and accessibility for end users, taking into account the probability determined by the most used areas, common activities, as well as, the most navigated links. The hypotheses the Markovian model contemplates are the following: i) assuming a finite number of states to describe the dynamic behavior of the widgets; ii) assuming a known distribution of initial probabilities, which reflects either what state belongs an application widget, or the percentages of widgets in each state in the application; iii) assuming that the transition from one current state to another in the future depends only on the current state (Markovian property); and iv) that the probability of this transition being independent of the time stage considered (stationary property), that is, it does not change in the study time of the system.

### 4 Research Methodology

This study is based on the engineering method (evolutionary paradigm) [33]: observe existing solutions, propose better solutions, build or develop, measure and analyze, repeat until no further improvements are possible. For this reason, the authors began the investigation process with a Systematic Mapping [9] and the Systematic Literature Review [10] that study the problem, approaches, and challenges present when

migrating legacy Web applications to mobile platforms in the last decade. After examining the previously published studies, the article establishes that the Markov has not been proposed as a tool for configuring User interfaces in the migration process of legacy applications to mobile. Having said that, we present an approach based on the Markovian model. In the following section, we introduce the approach which later is instantiated in a study case. Finally we conducted a simple evaluation reporting preliminary evidence about the benefit of our approach.

## 5 Our approach

The approach aims at adapting legacy sites by modifying its structure, content, and look&feel to become a mobile friendly application. The approach steps are described in the next sections and Fig. 1 shows the overall process roadmap [34].

### 5.1 Identify the legacy web site

In this step, a legacy website is chosen for its migration based on visible rendering problems when displayed on a smartphone device. To identify such a candidate, users are instructed to minimize or modify the size of the screen to verify if the content is rearranged in some way. If the content of the layout is not properly rendered, the application is considered as a typical case study. Another way is to access the site from different mobile devices, and check whether or not the site requires to perform scroll or zoom to browse or activate functionalities.

### 5.2 Functionality prioritization

In this step, the available functionality is evaluated based on its usage by expert users. The outcome is an ordered list of most used features which suggest the order they must be moved to the new application version.

#### 5.2.1 Define expert users

The selected expert users must have certain experience within the site, in order to get the simulation for expert user probabilities of navigation.

#### 5.2.2 Behavior Model Development

The behavior model is important because the models serve to specify the desired behavior of the system. Additionally, models manage to obtain a better image of the company and

an external and internal loyal customer is a brand propagator and thermometer. Next, in order to obtain the behavior model that will be represented on a Markov chain model, there is a specific scheme for each user role due to the fact that the navigation cannot be general, and each one involves widgets that may or may not coincide in certain cases. The first step to get the Markov model is to fill the state spaces, as in (1) defined with the widgets on the site. Then, the initial matrix (also called the initial probability vector) is filled in as in (2). Then, it is necessary to verify that  $P(0)$  adds 1. To start, the user always accesses the main page of the application. For this reason, the probability is 1 in the *widget1*.

$$E = [Widget1, Widget2, Widget3, \dots, Widgetn] \quad (1)$$

$$P(0) = [1, 0, 0, \dots, 0] \quad (2)$$

#### 5.2.3 Transition Matrix

The values of the transition matrix are probabilities, non-negative values from 0 to 1, and the sum of the cells belonging to a row must result in 1. The order of transition matrix must be equals to the number of widgets (#Widget), so the algorithm can multiply the transition matrix by the initial matrix. For each widget shown in a page selected in Step 2, a row and a column are placed in the matrix. Then, the matrix is filled out with the probabilities of navigation. In this case to fill in the matrix is to weigh (based on a Likert scale) the values according to the related activities, in a similar way as the closeness matrix of an SLP.

#### 5.2.4 Iterations Model

Once both the initial matrix and the transition matrix are obtained, the algorithm multiplies N times until obtaining a stationary matrix that represents the behavior model. It is called stationary distribution when the initial distribution does not change after multiplication by the transition matrix. As time goes by, it does not change with the passage of time and therefore it is called stationary or invariable distribution. It is said that a Markov Chain in discrete-time admits a stationary distribution to the extent that long-term probabilities exist and is independent of the initial distribution  $P(0)$ .

#### 5.2.5 Improvements Determination

After calculating the stationary matrix, the probabilities for optimal site navigation should be recorded in the database or file system. In addition, the strategies to establish the specific improvement of the site must be determined. Then, to make an initial improvement, first the algorithm must select

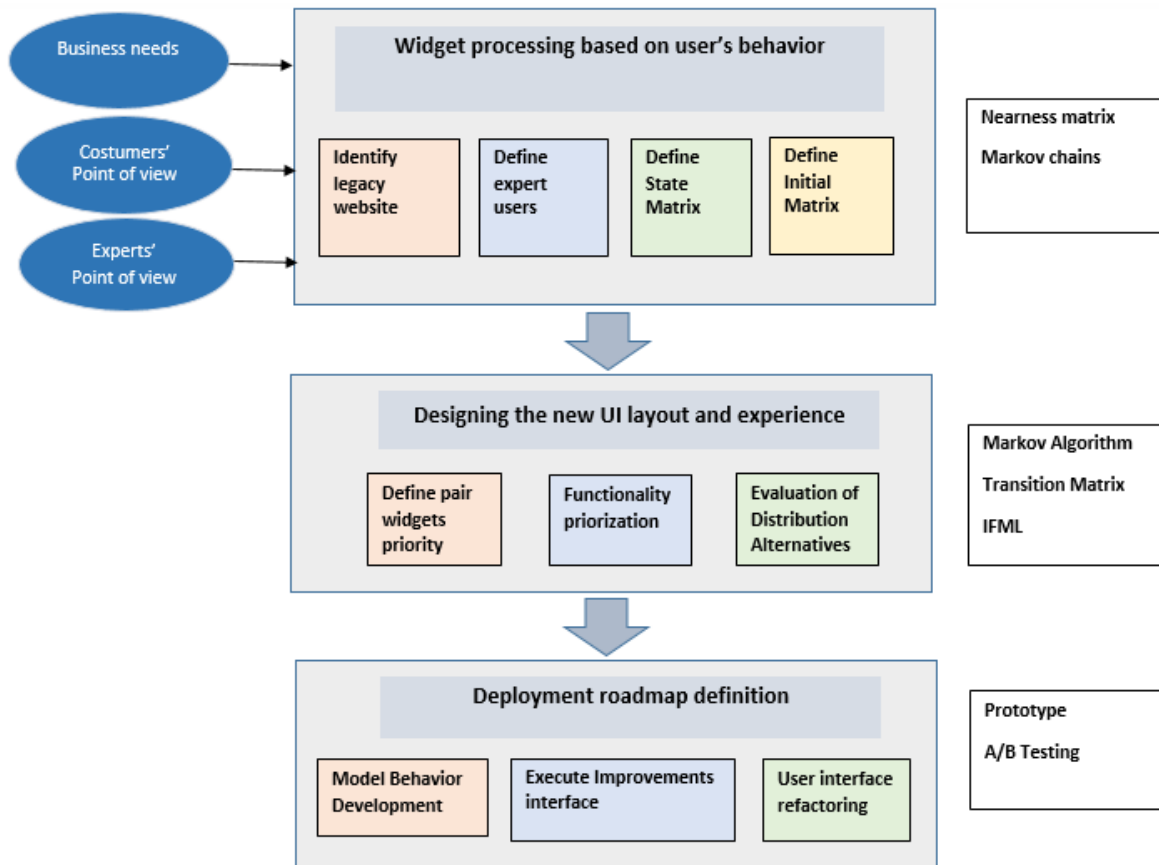


Fig. 1 Site Adaptation Roadmap

the widgets related to A (Absolutely necessary) and place them as close as possible on the user interface. On the contrary, the widgets related to X (Undesirable) or U (Without importance) must be placed as far as possible or kept invisible and only displayed on request, because there is no direct relationship between them.

5.2.6 Evaluation of Distribution Alternatives

The Table 2 shows the scheme of the evaluation of the usability that should be implemented to proceed with the improvement of the user interface. The effect of the manipulation of the independent variable (distribution of the widgets) is reflected in the dependent variables. Consequently, the screen distribution affects several aspects: the user performance (i.e spent time for the task, the effort required to accomplish), the User interface events (i.e. mouse clicks, scrolling and zooming, and keyboard keystrokes) and the interaction metaphors (i.e. using standard navigation between pages or using sophisticate metaphors like mouse hover or modal dialog).Therefore directly affects at cost and efficiency of the operation or task.

Table 2 Evaluation Matrix

Variable	Measure
Independent	Widgets Distribution
Dependent	Task completion time ( seconds)
	Input rate
	Count

5.3 User interface refactoring

During migration, the engineer needs not only to identify the most important features of the system but also the interaction experience requiring the refactoring of the interaction metaphor. So, the widgets of the pages are filtered accordingly to the Priority Matrix so that only most likely used widgets are kept.

The concept of Web model refactoring to seamlessly introduce Rich features into a Web application can be used; these specific WMRs RIA are called refactorings. RIA refactorings may be guided by RIA patterns, and they usually arise when it is discovered some issue in the application interaction style that can be eliminated by introducing some richer behavior convenient for the target device.

Some RIA refactorings are pure interface refactorings, but most of them combine transformations of the navigation and interface models. In fact, introducing RIA features often results in changing both the interaction styles and the hyper-text structure. This is the case, for example, when the target of a link is transcluded into the same page of the link.

The main goal of this step is that the user interfaces of a legacy application can be enriched supporting RIA features by:

- a Identifying the desirable changes by exploring a catalogue of refactorings and detecting issues in the corresponding application.
- b Applying the corresponding refactoring mechanics to the involved IFML Diagrams.
- c Generating the new application from the transformed IFML.

In this section, the attention will concentrate on steps a) and b) the above list, by showing that RIA refactorings can be obtained by oblivious composition of interface objects and interaction styles, which are modeled as belonging to different concerns (a related set of requirements). Figure 2 presents the IFML diagram of the legacy app navigation. For the sake of simplicity, only 4 components are modeled on the Posts main page. The [D] mark denotes the default page, which renders the list of Post (PostList), the social media links (SocialMedia component), the recommended posts grouped by category (Recommended Post By Category component), and the search form (search post component). The user can select one entry from the Post List component and then browse to the PostDetail page where the detail of the post is shown as well as the related news for such post. On the other hand, when filling out the search form, the user can access a link of those Posts that match the search criteria.

## 6 Running Example

In this section we will instantiate our approach providing detailed information about each of the steps. Then we will present a prototype resultant from the design. And finally we will describe an evaluation of the improvement.

### 6.1 Designing the migration

*Step 1.* Ecuador Legal Online [35] is a portal where Ecuadorian citizen can find support to legal questions. It is used as running example and, was described in section 1. The smartphone version of the EcuadorLegalOnline displays unnecessary information, while the content is reduced proportionally to the screen size, that is mean the version of site for a different device is a miniature of the original site. The site

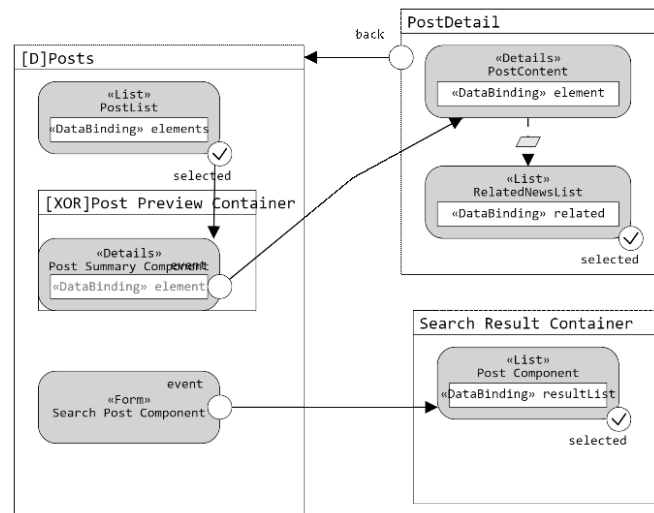


Fig. 2 IFML model of the legacy Web app

has several usability issues aforementioned and others will be detailed in the next section. When a user navigates on the main page all the information is displayed, that is to say, everything that the site offers, with an inefficient categorization opposite the Nielsen heuristics [36] recommendations, Fig 3.

*Step 2.* The selected users perform the following tasks:

1. They list the widgets present in the Ecuador Legal Online based on his/her user role, (Table 3).

Table 3 EcuadorlegalOnLine Tested Widgets

No.	Widget Name	No.	Widget Name
1	Legal Services	9	Procedure Guide
2	ANT	10	News
3	SRI	11	Calculator
4	Judicature	12	Technique Support
5	CNT	13	Help
6	CNE	14	Follow on Twitter
7	Work Code	15	Search
8	Cars		

2. According to the user’s navigation (Fig. 3), the widgets of Table 3 must be related in agreement to Muther methodology, as shown in Table 4. In a nearness matrix is important to include the reasons behind the selected closeness value, in order to have other priority at the moment of implementation. However in this special case for software, the column reason for all values corresponds to an administrative workflow. For this motive is not included in the table.

**Table 4** EcuadorLegalOnLine Closeness or Proximity Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>LegalServices</b>	1	5	4	2	1	3	4	3	1	1	2	3	1	2	3
<b>ANT</b>	2		4	3	2	4	3	4	3	1	1	3	4	1	3
<b>SRI</b>	3			3	1	1	3	2	2	1	1	2	3	1	2
<b>Judicature</b>	4				3	3	3	3	3	3	3	3	3	3	3
<b>CNE</b>	5					1	3	2	2	1	1	1	2	2	2
<b>CNT</b>	6						3	1	1	1	1	1	2	1	1
<b>Work code</b>	7							2	2	2	2	2	2	2	2
<b>Cars</b>	8								1	2	1	2	1	2	2
<b>Procedure Guide</b>	9							1		1	2	2	2	1	2
<b>News</b>	10									3	1	2	2	2	2
<b>Calculator</b>	11											3	2	2	2
<b>Technique Support</b>	12												2	1	2
<b>Help</b>	13													3	3
<b>Follow on Twitter</b>	14														2
<b>Search</b>	15														

*Step 3.* The space of states is established to determine the behavior model, such as:

$$E = [LegalServices, ANT, SRI, Judicature, CNT, CNE, WorkCode, Cars, ProcedureGuide, Calculator, TechniqueSupport, Help, FollowOnTwitter, Search, News] \quad (3)$$

Next, the algorithm must establish the initial matrix distribution:

$$P(0) = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \quad (4)$$

*Step 4.* The transition matrix is filled in with the percentages of closeness matrix. To do so, the algorithm must add the weights of the relation scale of each pair of widgets, which corresponds to the population of each row to obtain the percentages. The complete transition matrix is available in Appendix A.

*Step 5.* In order to compute the iteration model, the algorithm must multiply the initial matrix by the transition matrix to get the stationary matrix. The multiplication of the initial matrix by the transition matrix ( $P(0) \times P$ ) in the first iteration  $P(1)$  gives following result (Eq 5):

$$P(1) = [0, 00; 0, 14; 0, 11; 0, 06; 0, 03; 0, 09; 0, 11; 0, 09; 0, 03; 0, 03; 0, 06; 0, 09; 0, 03; 0, 06; 0, 09] \quad (5)$$

The next iterations must be calculated the same way according to theorem: If  $P(0)$  is the initial distribution vector and  $P$  is the transition matrix of a Markov chain, then the distribution vector from step  $n$  is the product  $P(n-1) \times P$ .

After that, the widgets will be relocated on the interface, according to their percentages resulting from iterations. The first time, it is used the nearness matrix because of there is no data in the transition matrix. After, the database or record file will record file stores the click events to recalculated the probabilities of use. For example, the Legal Services widget had the major percentage with 14.00%, for this reason, this widget will appear in first place on the interface and so on. The Java code for computing the transition matrix is in available at <sup>1</sup>.

*Step 6.* The determination of improvements is accomplished through the analysis of screen size: (i) Reorganizing the menu according to the results obtained; (ii) Including shortcuts to the features most likely to be used; (iii) Decreasing the zoom and scroll within the pages. For an initial improvement: the algorithm groups the widgets according to nearness matrix priority as represented in Table 5. This operation consists of pairing the widgets with value A, like Legal Services and ANT. Then we replicate the process with those widgets having value E, SRI and ANT and so on. Figure 5 shows the graphical interpretation [23] of Table 5 where widgets are sorted according Legal Services Widget pair priority. Similarly, the Figure 6 shows the graphical interpretation of column Table ANT widgets pair priority.

<sup>1</sup> <https://tinyurl.com/u52bzkw>





Fig. 3 Main page components of Ecuador Legal Online site

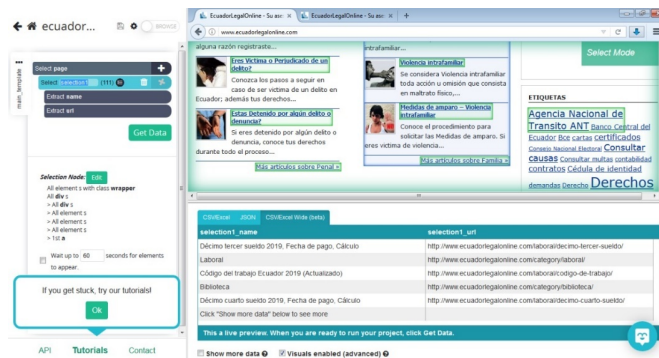


Fig. 4 Links navigated per user

Fig. 7 shows the link transition. The minimalist design is obtained first when a link is converted to a widget container, in order to offer a shortcut if it is one of the most used. The rest of the links become widgets children within a lower level with the purpose of to eliminate unnecessary controls that cause confusion.

Step 7. In this example, the Priority Matrix is used to select the most important items that should be kept in the mobile version. Based on application pages and their components, we use IFML to design the new application version

Table 5 Widgets Pair Priority

Key	Legal Services	ANT
A	1-2;	2-1;
E	1-3;1-7;	2-3;2-6;2-8;2-13;
I	1-6;1-8;1-12;1-15;	2-4;2-7;2-9;2-12;2-15;
O	1-4; 1-11;1-14;	2-5;
U	1-5; 1-9;1-10;1-13;	2-10; 2-11;2-14;

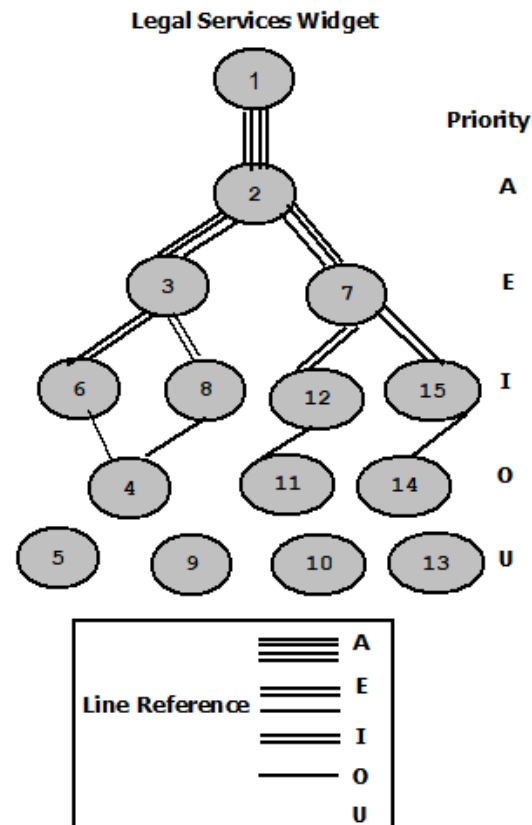


Fig. 5 Legal Services Widget Relationship Diagram

describing the new layout and component configuration. In Fig. 8, the PostList and Search Form Component are kept and the SocialMedia and Recommended Post by Category components were removed. The interaction was improved including a preview of a post in the same page, which is activated by selecting an element from the PostList Component. In this example, the navigation to the detail of a post in the PostDetail page, and the rendering of the search result in a different page (Search Result Container) were not modified.

Once raised the Markov distribution, it is necessary to determinate if the technology acceptance model (TAM) of the user is improved or still require other modifications. The evaluation consists of measuring the efficiency of new widgets distribution in comparison to the old one.

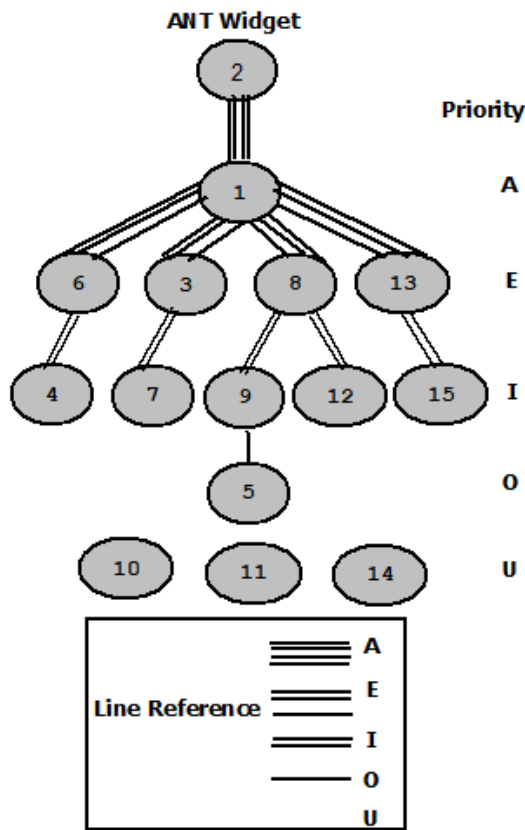


Fig. 6 ANT Widget Relationship Diagram

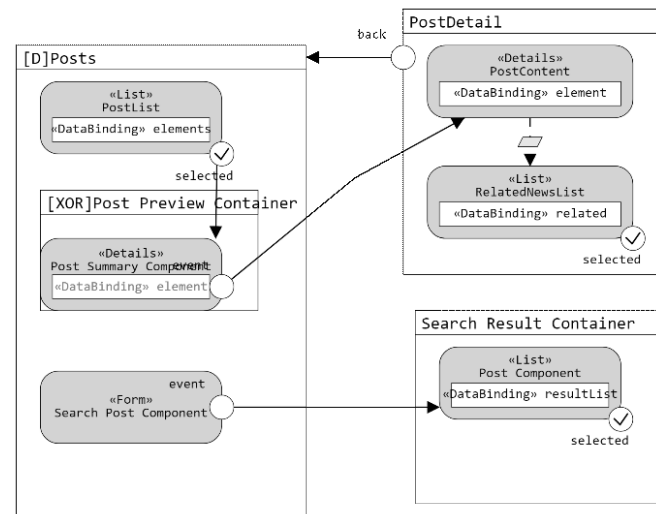


Fig. 8 Refactored User Interface Model

and Social Networks and like widgets children: Alimony, Divorces, Labor, Taxes, Penal, and Family; (iii) the interface allows a better navigation and improves the time spent on the task; and (iv) the interface has been prune removing unnecessary information implying a less cognitive load (minimalist design).

## 7 The usability evaluation

### 7.1 Experimental Design

In order to evaluate the new user interface, we conducted a preliminary evaluation that involved ten users who were divided into two independent groups, control and experimental, of 5 subjects each according to Nielsen recommendation [37]. Each subject was required to complete four tasks: (i) Search the divorces widget, (ii) Locate information about their legal services, (iii) Locate the taxes widget, and (iv) Find news with the search icon. Five participants performed the tasks on the EcuadorLegalOnline application (control group) and the five others on the EcuadorLegalOnline Prototype (experimental group). The hypothesis was that the user performance (e.g., completion time to carry out a task) should be better in the latter than the former group. The evaluation protocol was executed in the same way by all the subjects. They received a quick explanation of the EcuadorLegalOnline functionalities by a moderator. None of them have previously used the EcuadorLegalOnline application, and, moreover, they were not familiar with the objectives of the study. In addition, two witnesses observed the behavior of subjects while performing the required tasks in their mobile device using a Chrome browser. The assessment metric was based on the task completion time and the number of interaction events (zoom in/out, scroll, and click), Table 6. To

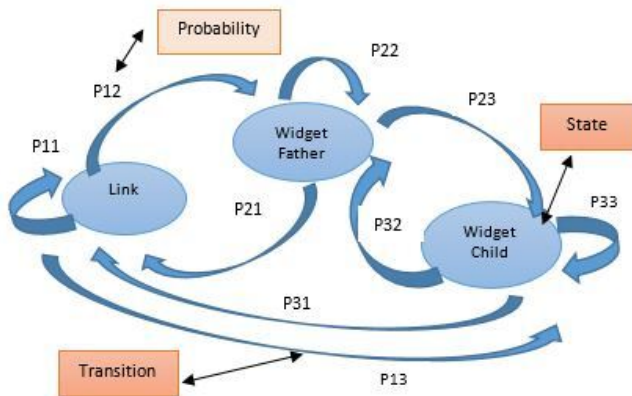


Fig. 7 Navigation Markov Chain Example Transition

### 6.2 The resultant prototype

Fig. 9 shows the new organization of the widgets based on the criterion of the users obtained through the closeness matrix and with the prototype. There are mayor changes from the UI configuration perspective: (i) The probabilities obtained with the first iteration ( $P(0) \times P$ ) allow locating the order of widgets  $P(1)$ ; (ii) the common relation among widgets allows reorganized the widgets like widgets container: Legal Services, Institutions, Forms, Articles of Interest, Help



Fig. 9 EcuadorLegalOnLine Prototype (Left: Container Widgets; Center: Children Widgets; Right: Page Navigation)

evaluate the new EcuadorLegalOnLine distribution a mobile application was created in ProtoPie de Adobe XD [38] that allowed to quickly replicate the new distribution obtained. The url of this application was shared with each participant to install and interact with the requested tasks.

7.2 Threats to validity

The scope of this approach leaves out the heuristic Nielsen number 9, covering *help users recognize, diagnose, and recover from errors* because the messages are part of the business logic and do not of the user interface. The heuristic 10 *helps nor documentation* neither is covered within the approach because this is a topic strictly implicit within the logic application. In addition, the measurement of multifactor productivity (MFP) is not simple due to some natural resources, capital and cost depend on the business.

Wholin Claes et al. [39] summarize the criteria proposed by Cook and Campbell [40] to avoid pitfalls to validity in the conclusions obtained from the case study. One threat corresponds to the *low statistical power*, which is the ability of the test to reveal a true pattern in the data. However, the proposed experiment is based on Nielsen’s recommendations [37], in which 5 users is sufficient to develop a usability test within a controlled environment. Another threat is *reliability of measures*, to overcome this issue a computational-based

timer was used. A form was filled out to keep track of measures and indirect metrics like the labor productivity [41]. Another threat is the *reliability of the treatment implementation*. In this case, the experiment was carried out with a protocol in which the selected users were naive participants in order to reduce the learning effect (maturation threat). A simultaneous training was received by all subjects regarding the functionalities of the site. All the individuals performed the same tasks requested according to their group (control vs experimental).

The experiment was conducted using devices with similar characteristics (i.e. bandwidth, hardware and measuring instruments) avoiding *instrumentation* threat. Moreover, the experiment was conducted in an isolated room. In this way, we avoided the *random irrelevancies in the experimental environment* threat where disturbing elements from outside that may alter the results, such as noise or a sudden interruption of the experiment.

7.3 Results

Table 6 shows the assessment result in an evaluation matrix of results. For each user we present the result for performing each task. Overall, the completion time was significantly improved in the experimental condition. The average completion time of task 1 was reduced from 5.46s to 2.80s. For

task 2 it was reduced from 34.97 to 22.42s. The task 3 it was reduced from 9.93s to 5.48s. Finally, the task 4 was reduced from 12.15s to 4.70s. In the current EcuadorLegalOnLine the Task 2 was the one that took the longest time in average to be completed by the users. On the other hand, the maximum optimization was reached on the task 4. Similarly, the number of user events was lower for the experimental than the control group in all the four tasks. It is to note that the new EcuadorLegalOnLine application mobile distribution avoided the screen glide because the widgets can be found at first sight. Also, this distribution provides the most optimal functional solution based on a hierarchy distribution. Such results support the hypothesis that implementing a dynamic user interface based on the Markov chains can optimize the navigation in migrated legacy Web applications.

### 7.3.1 Nielsen Heuristics Evaluation

Nielsen heuristics, which were mentioned in the introduction section, will be used to assess the application usability. For this reason, with the purpose of complementing the evaluation, Table 7 presents a comparison of site versions.

### 7.3.2 Productivity Evaluation

Labour productivity [41] is defined as output per worker or per hour worked. Factors that can affect labour productivity include workers skills, technological change, management practices and changes in other inputs (such as capital). For this reason, it should be noted that the UI improvements have a positive impact on labor productivity (LP) metrics [41] (Eq 6). In Equations 7 and 8 the productivity is computed for the legacy version and the new one respectively. The numerator corresponds to the number of tasks performed, while the denominator is obtained with the sum of the time invested by each of the users for the accomplishment of the tasks. The labour productivity growth (Eq 9) of interface improved is the 75%.

$$LP = \frac{\text{output}}{\text{hoursWorked}} \quad (6)$$

$$\begin{aligned} LP_{Old} &= \frac{\text{tasks}}{\text{secondsWorked}} \\ &= \frac{20}{62.51} = 0.32 \frac{\text{task}}{\text{second}} \end{aligned} \quad (7)$$

$$\begin{aligned} LP_{New} &= \frac{\text{tasks}}{\text{secondsWorked}} \\ &= \frac{20}{35.40} = 0.56 \frac{\text{task}}{\text{second}} \end{aligned} \quad (8)$$

$$\begin{aligned} LPGrowth &= \frac{LP_{New} - LP_{Old}}{LP_{Old}} * 100 \\ &= \frac{0.56 - 0.32}{0.32} * 100 = 75\% \end{aligned} \quad (9)$$

## 8 Conclusions and further work

A web information system consists of interconnected components where some of them are rendered in user interfaces [30]. For this reason, the application navigation can be modeled as a Markov chain, through finite states within a transactional Web system. This allows improving the software obtaining an optimized user interface promoting more intelligent navigation. Thus, this design-centered approach allows prioritizing the content of a legacy application in mobile devices. Once the layout is adjusted, the approach relies on IFML to model the interaction flow for those situations in which the original interaction metaphor does not properly work on the target device. The proposed approach, combining Markov and IFML, seems to bring a benefit in terms of usability. Firstly, the new distribution allows the users to locate faster the widgets they require. Secondly, the widgets are redistributed based on their functional affinity, which reduces the time to complete the tasks and the size of the scrollbar. Finally, the Markov chains allow improving the user interface, making navigation more intelligent cost-efficient, in-line with the principals of eco-efficiency proposed by Schaltegger [42] and Hart and Milstein [43]. The experimental validation shows that task completion time has improved in the new application version. Also, the hypotheses of the Markovian model has been successfully applied to improve the usability of the App, according to the Nielsen heuristic. Likewise, labour productivity has increased by 75% with the new design. The main outcomes of the study are summarized in Table 8.

Additionally, we will evaluate different algorithms to predict user behavior. In this case, Markov will be improved considering context variables. Secondly, the tool will allow validating the result on legacy applications using Web Augmentation. So the analyst can enhance the web page without modifying the source code. It would be also necessary to perform a throughout test evaluating on more metrics related to the user experience, such as emotions, attitudes, thoughts, behaviours, and perceptions of users (user experience), as suggested by [44]. To conclude, this study supports the idea that including Markov chains to predict user behavior could be one of the behavioral theories that can be applied to improve other approaches such as model-based systems in terms of automatic assignment of weights and priorities. Since it is a scientifically proven algorithm and it feeds itself according to the evolution of the system. Unlike the other approaches aforementioned, the Markov chains have been shown to provide a means of analyzing system components with an emphasis on empathy with the user. In addition, the system gets better optimization that involves its integral performance. For this reason, Markov chains have been part of the investigation of operations along the time

**Table 6** Old EcuadorLegalOnLine Measures Vs. New EcuadorLegalOnLine Measures

Old EcuadorLegalOnLine Measures					New EcuadorLegalOnLine Measures												
User	Task completion time seconds				UI Events				User	Task completion time /seconds				UI Events			
	T1	T2	T3	T4	T1	T2	T3	T4		T1	T2	T3	T4	T1	T2	T3	T4
<b>U1</b>	5.00	1.07	5.00	21.00	2	3	3	2	<b>U6</b>	1.00	26.00	3.00	5.00	1	1	2	1
<b>U2</b>	13.00	55.00	13.00	2.00	1	2	3	2	<b>U7</b>	6.00	22.00	9.00	4.00	1	1	2	1
<b>U3</b>	3.00	26.00	22.00	15.00	2	3	2	2	<b>U8</b>	3.00	24.00	4.00	6.00	1	1	2	1
<b>U4</b>	3.03	37.79	4.65	9.74	2	3	2	2	<b>U9</b>	2.01	2.08	1.38	3.50	1	1	2	1
<b>U5</b>	3.00	55.00	5.00	13.00	1	1	3	2	<b>U10</b>	2.00	38.00	10.00	5.00	1	1	2	1
<b>Avg.</b>	5.46	34.97	9.93	12.15	1.6	2.4	2.6	2		2.80	22.42	5.48	4.70	1	1	2	1

**Table 7** Nielsen Heuristics Comparison

Nielsen heuristic	EcuadorLegalOnLine Old Site	EcuadorLegalOnLine Proposed Site
1. Visibility of system status	This heuristics is observed partially because the real site does not have an adequate information structure.	This heuristic is appreciated during all the user interaction with the system because it keeps him informed about the section in which he is currently browsing.
2. Match between system and the real world	This heuristic is not visible throughout the website because the link resources do not have an homogeneous treatment. For example, there are certain images that can be browsed but other in a similar contest do not. This is a confusing functionality for the user.	This heuristic is visible in the redesign as it shows the user with clear and familiar words about the actions and possible mistakes that may be made during the interaction. There is no place for confusion because every icon or control with the same features has similar functions.
3. User control and freedom	This heuristic is partially appreciable during the application interaction because the user can use the back button to browse back to a page. In this way the user can recover from an error.	In the new version there are available a menubar and taskbar assisting the user in his activities.
4. Consistency and standards	This heuristic is not present in the real site because there are a lot of different controls without standardization.	The proposed site consider design guidelines for the target device which makes it friendly to users.
5. Error prevention	In the same way, the lack of standardization of the site is more prone to errors while being browsed.	This heuristic is perceivable in the redesign, it helps the user to be easily guided between the options that are presented during their interaction avoiding possible errors.
6. Recognition rather than recall	This heuristic is only noticeable when the user does not slide the information down since the top bar or taskbar automatically disappear completely which leaves the user without the guidance.	This heuristic is noticeable at all times in the redesign of the application. This is because the adoption of the best practices in mobile design. For example, the menu component will always point the section in which he is without having to remember where and what action he is executing or exploring.
7. Flexibility and efficiency of use	There is not any feature of flexibility for the user in the real site.	This heuristics is accomplished based on Markov chains to allow users to tailor frequent actions.
8. Aesthetic and minimalist design	Real site presents a lot of information and disordered visual components, unlike aesthetic and minimalist design recommendation of Nielsen. In addition, the user has a high cognitive load that increases the difficulty of use the site.	The redesigned application presents a better organization and simplification of options. The prototype has an optimal categorization so that the user has a lower cognitive load, which allows him to interact with the application more easily.

because the success of all cases depends on assertiveness in the definition of problems.

**References**

1. K. Asthon, "That 'Internet of Things' Thing," *RFID Journal*, p. 4986, 2010. [Online]. Available: <http://www.rfidjournal.com/article/print/4986>
2. P. M. Bosch-sijtsema, R. Fruchter, M. Vartiainen, and V. Ruohomäki, "A Framework to Analyze Knowledge Work in Distributed Teams," 2011.
3. C. Zott and R. Amit, "Business model innovation: toward a process perspective," *Oxford Handbook of Creativity, Innovation and Entrepreneurship*, pp. 395–406, 2015.
4. M. Stanley, "The Mobile Internet Report," *Most*, pp. 9–59, 2009.
5. T. O'Reilly, "What is web 2.0?: design patterns and business models for the next generation of software. O'Reilly." no. 65,

- pp. 17–37, 2005. [Online]. Available: <http://oreilly.com/web2/archive/what-is-web-20.html>
6. D. Zimmerman, D. Ph, and T. Yohon, "Small-screen Interface Design : Where Are We ? Where Do We Go ? Literature : Small-screen Interface Design," pp. 1–5, 2009.
7. H. Hoehle, R. Aljafari, and V. Venkatesh, "Leveraging Microsoft's mobile usability guidelines: Conceptualizing and developing scales for mobile application usability," *International Journal of Human Computer Studies*, vol. 89, pp. 35–53, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.ijhcs.2016.02.001>
8. Google, "Mobile-Friendly Test - Google Search Console."
9. V. Cajas, M. Urbietta, Y. Rybarczyk, G. Rossi, and C. Guevara, "Portability Approaches for Business Web Applications to Mobile Devices: A Systematic Mapping," 2018.
10. V. E. Cajas, M. M. Urbietta, and G. Rossi, "Migrating Web to Mobile : A Systematic Literature Review," *IEEE LATIN AMERICA TRANSACTIONS*, vol. 18, no. 5, pp. 861–875, 2020.
11. J. Rivero, M. Urbietta, S. Firmenich, M. Witkin, R. Serrano, V. Cajas, and G. Rossi, "Improving legacy applications with client-

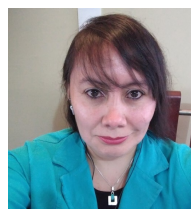
**Table 8** Results Comparison

Variable	Old System	New System
Task 1 completion time	5.46 seconds	2.80 seconds
Task 2 completion time	34.97 seconds	22.42 seconds
Task 3 completion time	9.93 seconds	5.48 seconds
Task 4 completion time	12.15 seconds	4.70 seconds
Task 1 UI Events count	1.6	1.0
Task 2 UI Events count	2.4	1.0
Task 3 UI Events count	2.6	2.0
Task 4 UI Events count	2.0	1.0
Productivity	0.32 task/second	0.56 task/second
Nielsen Heuristic 1	Partial	Total
Nielsen Heuristic 2	Nothing	Total
Nielsen Heuristic 3	Partial	Total
Nielsen Heuristic 4	Nothing	Total
Nielsen Heuristic 5	Partial	Total
Nielsen Heuristic 6	Partial	Total
Nielsen Heuristic 7	Nothing	Total
Nielsen Heuristic 8	Nothing	Total

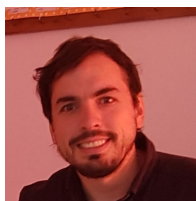
side augmentations,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10845 LNCS, pp. 1–15, 2018.

12. P. Taylor, P. Kortum, M. Sorber, P. Kortum, and M. Sorber, “International Journal of Human-Computer Interaction Measuring the Usability of Mobile Applications for Phones and Tablets Measuring the Usability of Mobile Applications for Phones and Tablets,” no. August, 2015.
13. J. Brooke, “SUS - A quick and dirty usability scale,” *Usability Evaluation in Industry*, pp. 4–7, 1996.
14. A. Bangor, T. Staff, P. Kortum, J. Miller, and T. Staff, “Determining What Individual SUS Scores Mean : Adding an Adjective Rating Scale,” vol. 4, no. 3, pp. 114–123, 2009.
15. G. Rossi, M. Urbietta, J. Ginzburg, D. Distant, and A. Garrido, “Refactoring to Rich Internet Applications . A Model-Driven Approach,” pp. 1–12, 2008.
16. G. Rossi, M. Urbietta, D. Schwabe, and D. Informatica, “Designing the Interface of Rich Internet Applications.”
17. V. Cajas, M. Urbietta, Y. Rybarczyk, G. Rossi, and C. Guevara, “An Approach for Migrating Legacy Applications to Mobile Interfaces,” in *In World Conference on Information Systems and Technologies*, vol. 3. Springer, Cham, 2019, pp. 916–927.
18. J. E. Cook and A. L. Wolf, “Discovering models of software processes from event-based data,” *ACM Transactions on Software Engineering and Methodology*, vol. 7, no. 3, pp. 215–249, 1998. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=287000.287001>
19. J. A. Whittaker and J. H. Poore, “Markov analysis of software specifications,” *ACM Transactions on Software Engineering and Methodology*, vol. 2, no. 1, pp. 93–106, 1993. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=151299.151326>
20. C. S. D. Murphy, K. P.(University of California, Berkeley, *Inference and learning in hybrid Bayesian networks.*, 1998.
21. H. Thimbleby, P. Cairns, and M. Jones, “Usability analysis with Markov models,” *ACM Transactions on Computer-Human Interaction*, vol. 8, no. 2, pp. 99–132, 2001. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=376929.376941>
22. P. Boldi, M. Santini, and S. Vigna, “PageRank: Functional Dependencies,” *ACM Transactions on Information Systems*, vol. 27, no. 4, pp. 19:1–19:23, 2009.
23. J. D. Muther, R., & Wheeler, J. D. (1994). *Simplified systematic layout planning. Management and Industrial Research Publications.ing. Management and Industrial Research Publications.*, 1994.
24. G. Rossi, Ó. Pastor, D. Schwabe, and L. Olsina, *Web Engineering: Modelling and Implementing Web Applications*, vol. 12. Springer-Verlag London, 2008.
25. G. Aragón, M. J. Escalona, M. Lang, and J. R. Hilera, “An analysis of model-driven web engineering methodologies,” pp. 413–436, 2013.
26. OMG, “Object Management Group.”
27. FMLTm, “Interaction Flow Modeling Language.” [Online]. Available: <https://www.omg.org/spec/IFML>
28. D. Lowe, B. Henderson-Sellers, and A. Gu, “Web extensions to UML: Using the MVC triad,” in *International Conference on Conceptual Modeling*, 2002, pp. 105–119.
29. C.-Y. Mao and Y.-S. Lu, “Testing and evaluation for Web usability based on extended Markov chain model,” *Wuhan University Journal of Natural Sciences*, vol. 9, no. 5, pp. 687–693, 2004. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-8844267550{&}partnerID=40{&}md5=e1ab4c3b1dd2b2159be5b70e473e661a>
30. C. Yanchun and W. Xingpeng, “A Security Risk Evaluation Model for Mashup Application,” 2009.
31. A. Chohan and A. Bibi, “Optimized Software Product Line Architecture and Feature Modeling in Improvement of Spl,” in *International Conference on Frontiers of Information Technology*, 2017.
32. C. Nwobi-Okoye and S. Okiy, “Application of game theory to software user interface evaluation,” *Cogent Engineering*, vol. 5, no. 1, pp. 1–18, 2018. [Online]. Available: <https://doi.org/10.1080/23311916.2018.1452471>
33. M. Lázaro and E. Marcos, “Research in Software Engineering : Paradigms and Methods,” *CAiSE Workshops (2)*, pp. 517–522, 2005.
34. I. Energy Agency, “Technology Roadmap A guide to development and implementation,” p. 27, 2014.
35. EcuadorLegal, “EcuadorLegalOnline - Su asesor legal en casa,” 2019. [Online]. Available: <http://www.ecuadorlegalonline.com/>
36. J. Nielsen, “Ten Usability Heuristics by Jakob Nielsen,” p. 10. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>
37. Nn/g, “Why You Only Need to Test with 5 Users,” 2000.
38. AdobeXD, “ProtoPie,” 2019. [Online]. Available: <https://tinyurl.com/wds36qa>
39. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*, 2012, vol. 9783642290.
40. K. H. Á, “Current practice in measuring usability : Challenges to usability studies and research,” vol. 64, pp. 79–102, 2006.
41. R. B. o. A. Education, “Productivity,” pp. 1–5, 2019. [Online]. Available: <https://www.rba.gov.au/education/resources/explainers/productivity.html>
42. R. Schaltegger, S., & Burritt, *Contemporary environmental accounting: issues, concepts and practice.* Routledge, 2017.
43. M. B. Hart, S. L., & Milstein, “Creating sustainable value,” *Academy of Management Perspectives*, vol. 17, no. 2, pp. 56–67, 2003.
44. R. Beaugregard and P. Corriveau, “User Experience Quality : A Conceptual Framework for Goal Setting and Measurement,” pp. 325–332, 2007.

**Viviana Elizabeth Cajas** graduated from Engineering in Systems and Informatics in the year 2004 in the ESPE (Ecuador). Since 2012 she is Magister degree in Business Administration and Marketing from Universidad Tecnológica Indoamérica (UTI) from Ecuador. She is a Ph.D. student



in Computer Science at the National University of La Plata (Argentina). She is the author of 3 Chapters of books and 5 articles of conferences. Her interests include legacy applications and their portability in terms of user, ecological, ergonomic, ergonomics, proactive and evolutionary software. In addition, the analysis of consumer behavior with a generational basis. She is currently a professor at the UTI.



**Mario Matias Urbietta** was born in Formosa, Argentina in 1981. He received this B.S. and Ph.D in computer science from the National University of La Plata (UNLP). Since 2008, he is a Research Assistant at Laboratorio de Investigación y Formación en Informática Avanzada (LIFIA), La Plata, Buenos Aires, Argentina. He is the author of 1 book chapter, 10 Journal Articles and 20 Conference papers. His research interests

include separation of concerns in Web Applications, Agile requirement engineering and Web Augmentation. He teaches Object-Oriented Programming concepts and Web engineering at bachelor, and doctoral levels at UNLP and Universidad Abierta Interamericana.



**Gustavo Rossi** received his PHD diploma in PUC-Rio, Brazil in 1996. His PhD thesis was the development of the OOHDM design approach, one of the mature methods for the development of Web Applications. He is currently full professor at Facultad de Informática, Universidad Nacional de La Plata. He has been a visiting professor at the Universities of Lyon and Montpellier in France and has received the Habilitation pour Diriger Recherches (HDR) at INSA-Lyon. He has been part of the PC committee of the most important conferences of his research field such as ACM WWW, ICWE, and ACM Hypertext. He is member of the editorial board of IEEE Internet Computing, IEEE IT Professional, WWW Journal, Journal of Web Engineering, Journal of Internet Services and Applications and the International Journal of Cooperative Information Systems.

He has been part of the PC committee of the most important conferences of his research field such as ACM WWW, ICWE, and ACM Hypertext. He is member of the editorial board of IEEE Internet Computing, IEEE IT Professional, WWW Journal, Journal of Web Engineering, Journal of Internet Services and Applications and the International Journal of Cooperative Information Systems.



**Yves Rybarczyk** has a PhD in robotics from the University of Evry, France. His teaching and research activities focus on Artificial Intelligence, HCI, Data Mining and Modelling. He was assistant professor in the Department of Electrical Engineering at the Nova University of Lisbon, Portugal, between 2007 and 2015. Then, he moved to South Amer-

ica where he held a position of associate professor and head of the Intelligent Interactive Systems Lab (SI2 Lab) at the Universidad de Las Américas, Ecuador, until 2019. Currently, he is full professor in the Faculty of Data and Information Sciences at Dalarna University, Sweden. He has participated in several projects on the modelling of complex systems for environmental and health purposes. He is the author of over 90 scientific publications.

## A Appendix A

The transition matrix (P) is shown in Table 9 calculated based on the proximity matrix.

**Table 9** Transition Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>LegalServices</b>	0,00	0,14	0,11	0,06	0,03	0,09	0,11	0,09	0,03	0,03	0,06	0,09	0,03	0,06	0,09
<b>ANT</b>	0,12	0,00	0,10	0,07	0,05	0,10	0,07	0,10	0,07	0,02	0,02	0,07	0,10	0,02	0,07
<b>SRI</b>	0,13	0,13	0,00	0,10	0,03	0,03	0,10	0,07	0,07	0,03	0,03	0,07	0,10	0,03	0,07
<b>Judicature</b>	0,05	0,07	0,07	0,00	0,07	0,07	0,07	0,07	0,07	0,07	0,07	0,07	0,07	0,07	0,07
<b>CNE</b>	0,04	0,08	0,04	0,13	0,00	0,04	0,13	0,08	0,08	0,04	0,04	0,04	0,08	0,08	0,08
<b>CNT</b>	0,13	0,17	0,04	0,13	0,04	0,00	0,13	0,04	0,04	0,04	0,04	0,04	0,08	0,04	0,04
<b>Work code</b>	0,11	0,09	0,09	0,09	0,09	0,09	0,00	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06
<b>Cars</b>	0,11	0,15	0,07	0,11	0,07	0,04	0,07	0,00	0,04	0,04	0,07	0,04	0,07	0,04	0,07
<b>Procedure Guide</b>	0,04	0,12	0,08	0,12	0,08	0,04	0,08	0,04	0,00	0,04	0,08	0,08	0,08	0,04	0,08
<b>News</b>	0,05	0,05	0,05	0,05	0,14	0,05	0,05	0,09	0,05	0,00	0,14	0,05	0,09	0,09	0,09
<b>Calculator</b>	0,05	0,05	0,05	0,05	0,14	0,05	0,05	0,09	0,05	0,00	0,14	0,05	0,09	0,09	0,09
<b>Technique Support</b>	0,11	0,11	0,07	0,11	0,04	0,04	0,07	0,04	0,07	0,04	0,11	0,00	0,07	0,04	0,07
<b>Help</b>	0,03	0,12	0,09	0,09	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,00	0,09	0,09
<b>Follow on Twitter</b>	0,08	0,04	0,04	0,13	0,08	0,04	0,08	0,04	0,04	0,08	0,08	0,04	0,13	0,00	0,08
<b>Search</b>	0,10	0,10	0,06	0,10	0,06	0,03	0,06	0,06	0,06	0,06	0,06	0,06	0,10	0,06	0,00