International Journal of Web Information Systems

Tukuchiy: a dynamic user interface generator to improve usability

Luisa Barrera-León Nadia Mejia-Molina Angela Carrillo-Ramos Leonardo Flórez-Valencia Jaime A. Pavlich-Mariscal

Received 18 September 2015
Revised 25 January 2016
Accepted 28 January 2016

Tukuchiy: a dynamic user interface generator to improve usability

Luisa Barrera-León, Nadia Mejia-Molina,
Angela Carrillo-Ramos and Leonardo Flórez-Valencia
Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana, Bogota, Colombia, and
Jaime A. Pavlich-Mariscal
Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana, Bogotá, Colombia and Departamento de Ingenieria de Sistemas y Computacion, Universidad Catolica del Norte, Antofagasta, Chile

Abstract

Purpose – This paper aims to present a detailed description of Tukuchiy, a framework to dynamically generate adapted user interfaces. Tukuchiy is based on Runa-Kamachiy, a conceptual integration model that combines human–computer interaction (HCI) standards to create user interfaces with user-centered concepts usually addressed by adaptation.

Design/methodology/approach – The first step was the definition of three profiles: user, context and interface. These profiles contain information, such as user disabilities, location characteristics (e.g. illumination) and preferences (e.g. interface color or type of system help). The next step is to define the rules that ensure usability for different users. All of this information is used to create the Tukuchiy framework, which generates dynamic user interfaces, based on the specified rules. The last step is the validation through a prototype called Idukay. This prototype uses Tukuchiy to provide e-learning services. The functionality and usability of the system was evaluated by five experts.

Findings – To validate the approach, a prototype of Tukuchiy, called Idukay, was created. Idukay was evaluated by experts in education, computing and HCI, who based their evaluation in the system usability scale (SUS), a standard usability test. According to them, the prototype complies with the usability criteria addressed by Tukuchiy.

Research limitations/implications – This work was tested in an academic environment and was validated by different experts. Further tests in a production environment are required to fully validate the approach.

Originality/value – Tukuchiy generates adapted user interfaces based on user and context profiles. Tukuchiy uses HCI standards to ensure usability of interfaces that dynamically change during execution time. The interfaces generated by Tukuchiy adapt to context, functionality, disabilities (e.g. color blindness) and preferences (usage and presentation) of the user. Tukuchiy enforces specific HCI standards for color utilization, button size and grouping, etc., during execution.

Keywords Web design metrics, Web data integration, Web-based education

Paper type Research paper

The authors thank the Systems Engineering and Computing Masters program from the Pontificia Universidad Javeriana for the funding provided for this project. The authors also thank people who assisted in the usability tests performed over Idukay: Diana Lancheros, Luis Carlos Díaz, Andrés de la Peña, Andrea Barraza and Patricia Gómez.

International Journal of Web Information Systems
Vol. 12 No. 2, 2016
pp. 150-176

1744-0084
DOI 10.1108/IJWIS-09-2015-0028

1. Introduction

During the 1980's, information system developers started to focus on the way to convey information to the user and improve the user interaction with systems (Card *et al.*, 1983). Human–computer interaction (HCI) emerged as a specialized research area. Its impact has grown over the past decades and has attracted professionals from other knowledge areas, such as psychology and ergonomy. Because of that, HCI has incorporated new concepts and paradigms in user interface (UI) development (Carroll, 2013). UI design comprises those "aspects of the system with which the user makes contact physically, perceptually or conceptually, while the aspects that are hidden from the user are called 'implementation'" (Calvary *et al.*, 2000). In other words, UI provides graphical or text elements that allow the user to perform tasks in a software application. To let UI accomplish their objective, several standards have been created to ensure that those interfaces are similar to user world and enable a more natural interaction with the system (Nielsen, 1995).

Usability is the common element of those standards. Usability in a UI is defined by several quality attributes proposed by Nielsen (1995) (Carvajal and Saab, 2010): efficiency, efficacy, satisfaction, facility to learn, memorization, error prevention and attractiveness.

Usability is also addressed by other disciplines. One of them is adaptation. An adaptive system is "A system that can change itself as a response to changes in its environment, in such way that its performance becomes better through the interaction with said environment" (Parker, 1984). Adaptive systems change according to the specific user and context characteristics, making it possible to achieve quality attributes, such as satisfaction and attractiveness. Changes performed by an adaptive system are reflected in three levels: presentation, in the UI design; content, enriching the information presented to the user; and navigation, reducing the amount of steps to perform a specific task.

Because of their similarities, HCI and adaptation have the potential to be integrated to achieve usability in adapted UIs. To achieve this goal, we created Runa-Kamachiy (Barrera *et al.*, 2013b), a conceptual integration model between HCI and adaptation, to improve UI usability.

This paper realizes the above model through a dynamic UI generator, called Tukuchiy, which takes into account classic HCI precepts and user characteristics. Tukuchiy is a general-purpose framework to dynamically generate UIs that adapt themselves during execution time, according to user profiles, context, and UI and presentation preferences. In addition, Tukuchiy enforces some HCI rules at execution time, to build a UI that supports usability criteria.

The remainder of this paper describes Tukuchiy and the way it integrates HCI and UI adaptation. Section 2 describes related work in dynamic generation of adapted interfaces. Section 3 briefly describes Runa-Kamachiy. Section 4 describes the design and implementation of Tukuchiy. Section 5 describes the conceptual and unit tests performed to Tukuchiy. Section 6 describes a proof-of-concept prototype, called Idukay, an e-learning system, used to validate Tukuchiy. Section 7 concludes and describes future work.

2. Related work

There are several works in the area of adaptation and HCI. Ergo-Conceptor (Moussa *et al.*, 2000) is a tool that takes as input a functional description of the application and generates UIs specifications that can be used to design the UI. These specifications are based on several HCI standards. For instance, the tool proposes a *layout* according to the application's purpose. This tool proposes pre-defined representations (such as stars or graph charts) to avoid cognitive errors (such as using red in components not related to alarms or dangerous situations). However, as Ergo-Conceptor was developed for industrial UIs, screens types are limited to monitoring and production machinery commands. Although the system uses HCI concepts and takes user and interaction into account, it does not take into account user preferences or context. The UI in Ergo-Conceptor does not change during execution time.

The authors in Criado *et al.* (2010) indicate that execution-time interface adaptation should be developed using a different paradigm than design-time interfaces. Consequently, UIs should be able to re-generate depending on context, user interaction and changing requirements. They propose a model-driven approach that has two stages:

(1) creating a meta-model to specify a component-based structure, visual behavior and UI architectures interaction; and
(2) performing a model-to-model transformation to adjust UIs according to behavioral rules and the actions performed by the user during execution time.

Although adaptation is performed at the meta-model level, using profiles and context data, UI adaptation is purely structural (e.g. component placement) and does not take into account certain rules to preserve usability (e.g. color combinations). Moreover, the system becomes more complex, as the interface adapts according to interactions between the user and the system, but it does not take into account user preferences (e.g. colors, sizes) or disabilities (e.g. color blindness).

DB-USE (Tran, 2010) is an application to dynamically create UIs and generate Java source code for database applications with generic interfaces. DB-USE creates the UI based on two models: a task model and a domain model. It also takes into account the code generation process. Each of these models is used as resources that are analyzed by different application processes. However, as DB-USE is model-driven, the interface is generated directly from models, so it does not take into account HCI rules for UI design in early development stages. In addition, DB-USE generates interfaces based on a static user model. As user characteristics and preferences may change over time, a static profile does not let the UI adapt during execution, but only during development.

GenURC, is a platform to automatically generate UIs (Zimmermann *et al.*, 2013), taking into account user preferences (presentation), device characteristics and other context data. GenURC focuses on Web interfaces of simple applications, such as domotic device controls through smartphones and tablets. However, GenURC focuses only on mobile devices, which may make more difficult to use in other contexts. Additionally, although it supports usability (e.g. button size limitations), it does not take into account HCI standards that refer to possible functional diversities of the user because the user preferences are only tied to the display preferences, such as colors.

Namgoong *et al.* (2006) propose a system to generate dynamic adapted interfaces for mobile devices. This system comprises four modules:

(1) a knowledge base with ontologies that describe the device, environment and messages (device access information), as a set of logical sentences;

(2) a message manager that communicates with devices in the environment and sends the reasoner description;

(3) a reasoner that translates concepts and descriptions to domain-specific concepts and compositions, which are more adequate to provide the required information to the UI engine; and

(4) a UI engine that uses the above information to create the UIs. This system takes into account user preferences. For instance, if a user gets close to a TV and his/her favorite TV program is being presented, the device creates an interface to interact with the TV and turns on the TV.

This system does not take into account the HCI standards for UI creation. It only creates buttons to activate the actions. Moreover, there is no user profile, but only preferences that are created from the usage history.

The Flexible Learning for Open Education (FLOE) project (Treviranus *et al.*, 2014) provides inclusive personalized resources. It provides different options for users according to their learning characteristics and also to content providers, through the open education resources community. FLOE lets students personalize their learning experience through a preference exploration tool, provides documentation and resources to create inclusive courses and has a metadata editor to add preferences to each object in the system.

FLOE provides a way to change some aspects of the UI (colors, font size, button and link styles, etc.) However, this process is not dynamic, as the user must manually change these elements. Preferences are associated to learning objects, but there are no preferences explicitly associated to UI, which means that the system has no adequate information to decide the best interface for each student. Moreover, FLOE only focuses on educational interfaces, and it is not a general-purpose framework.

Table I compares the related work discussed in this section. The (+) symbol means that the approach explicitly supports the corresponding criterion. The (±) symbol means that implicitly supports the criterion. The (−) symbol means that it does not supports the criterion.

Most approaches take into account the user profile. However, user profiles are built only to yield the initial UI design. They do not take into account that those profiles may change over time and affect the UI design.

Few approaches directly use HCI techniques to generate dynamic UIs. When an approach uses any HCI techniques, it does not use them during execution time, but only during design time.

Overall, it is necessary to integrate user profiles and context with HCI techniques to improve the interaction between users and systems. Those techniques should be used both during design and execution.

3. Runa-Kamachiy

Runa-Kamachiy[1] (Barrera *et al.*, 2013b) is a conceptual integration model that combines HCI standards to create UIs with user-centered concepts usually addressed by adaptation. Concepts integrated by Runa-Kamachiy directly address usability criteria (Section 1). More information can be found in one of the authors' masters thesis (Barrera-Leon, 2015b).

Table I. User interface generator comparison

Criterion	Moussa *et al.* (2000)	Criado *et al.* (2010)	Tran (2010)	Zimmermann *et al.* (2013)	Namgoong *et al.* (2006)
Takes into account physical user aspects	−	−	+	−	−
Takes into account user context	+/−	+	−	+	+
Takes into account user preferences	−	−	+	+	+
Takes into account user profile changes during execution time	−	+	−	−	+/−
Utilizes HCI standards to design the UI	+	+	+	+	−
Utilizes HCI methods/techniques (e.g., Fitts law)	+	−	−	+/−	−
Utilizes task models	−	+	+	−	−
Takes into account HCI methods/techniques during execution time (E) or design time (D)	D	E,D	D	D	D

Figure 1 is a conceptual diagram that describes *Runa-Kamachiy and its relation to other tools*. To realize Runa-Kamachiy, a framework, called Tukuchiy, was developed. Tukuchiy automatically generates dynamic UIs, based on the concepts of Runa-Kamachiy. Two prototypes based on Tukuchiy were created: an e-learning application, called Idukay (Section 6), and a radiology application called Midiku (Barrera *et al.*, 2014).

Runa-Kamachiy uses four main sources of information:

(1) the Ishihara test, which detects color blindness;

(2) basic user information: name, role within the system, gender, etc.;

(3) usage history, which yields the presentation preferences of each user; and

(4) enriching the context profile the system extracts information about the device used to access the system and access time.

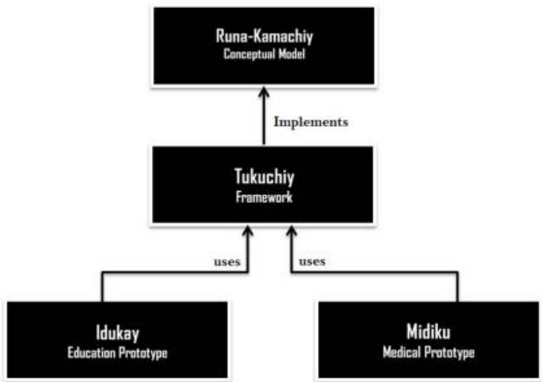

Figure 1. Runa-Kamachiy and its relation with Tukuchiy, Idukay and Midiku

Adaptation models are organized into three categories: user, context and interface. Each of these models is the following sections.

3.1 User profile

The user profile models all of the relevant user aspects. As described in Figure 2, the user profile represents two main aspects: basic characteristics and preferences. Basic characteristics represent three more specific user aspects:

(1) physical characteristics, such as users' functional diversities or health problems (e.g. color blindness);

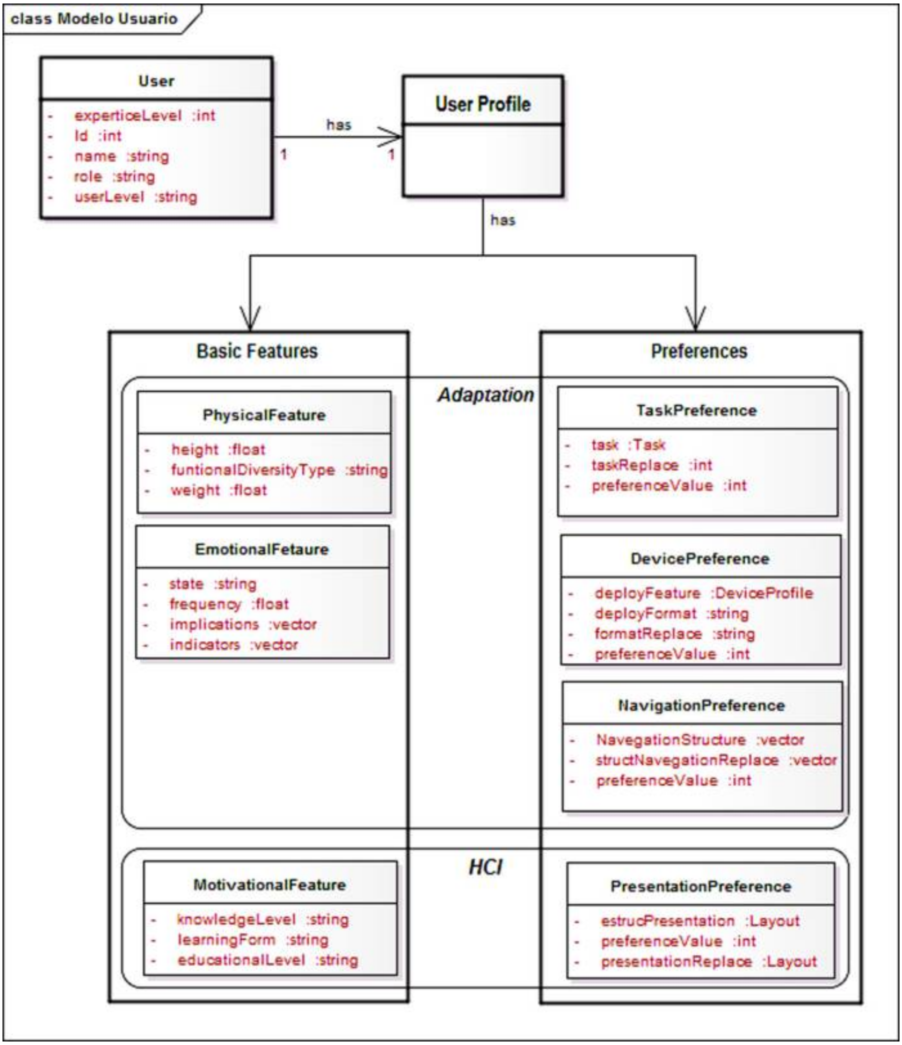

Figure 2.
User profile

(2) motivational characteristics, which are related to user learning level and strategies; and
(3) emotional characteristics, which denote the user emotional state and the way it affects his/her performance when using the system.

Tukuchiy address all of the above, except emotional characteristics. As this aspect requires very different techniques to recognize user emotions (Maat and Pantic, 2006), it is outside the scope of this research.

Preferences are divided into four types:

(1) task preferences – users may perform the same tasks differently and they may prefer different ways to perform tasks (e.g. executing a command vs pressing a button);
(2) device preferences, related to the presentation formats preferred by users, which are constrained by contextual characteristics of the access device;
(3) navigation preferences, related to the preferred way of interaction between user and system; and
(4) presentation preferences, the way the user prefers to perceive different UI elements (e.g. color and position of a button).

The focus of this research is UI presentation and the corresponding preferences. Therefore, Tukuchiy does not explicitly address device or navigation preferences. However, the framework was designed with extensibility in mind. The authors believe that the integration of new types of preferences would not require big changes to the framework. The following is Backus-Naur Form (BNF) (Backus, 1958) formalization of the user profile:

User Profile
<User_Profile>::= <Basic_Data> <Physical_Features>
<Motivational_Characteristic> <Emotional_ Characteristic>
Basic Data
<Basic_Data> ::= <Name> <LastName> <Age> <Address> <Gender>
<Birthday-Date> <PhoneNumber> <e-mail>
(<Name> <lastName> <Address> <e-mail>) ::= <String>
Physical Features
<Physical_Features> ::= <PhysicalIssues_Type> <weight>
<height>(<weight> <height>) ::= <number>
<PhysicalIssues_Type> ::= <issueName> <InterfaceImplications>
Motivational Features
<MotivationalFeature> ::= <knowledgeLevel> <learningStyle> <educationalLevel>
<knowledgeLevel> ::= Few | Medium | High
<learningForm> ::= visual | auditory | reading-writing | kinesthetic
<educationalLevel> ::= high school | university | …

Based on this profile and the related work analysis (see Section 2), one can conclude that the most common Adaptation concepts are similar to their counterparts in HCI and several of them match very closely. However, emotional characteristics and presentation preferences are concepts most commonly adressed by HCI.

3.2 Context profile

The context profile shows environmental characteristics that, according to Runa-Kamachiy, are relevant to improve interface usability. Figure 3 describes the context profile, which includes the following HCI concepts:

- ergonomic profile, which denotes physical characteristics of the user context and their constraints;
- environment profile, which represents the environmental conditions that surround the user (e.g. luminosity, weather, etc.); and
- task profile, which represents the tasks that will be performed by the user and the activities that are part of those tasks.

An adaptation concept addressed by this profile is the device profile that describes the devices used by the user to access the system, based on the "*Composite Capability/ Preference Profile*" (Klyne *et al.*, 2004). This part of the profile is improved with the user interaction information with the device, which provides additional context information.

The following text is BNF (Backus, 1958) formalization of the context profile:

Context
<Context> ::= <ErgonomicProfile> <EnviromentalProfile><DeviceProfile><Task>
Ergonomic Profile
<ErgonomicProfile> ::= <Restrictions> <UserLocation>
<Restrictions> ::= <Software> <Hardware> ::= <text>
<UserLocation> ::= <visualDistance> <visualAngle> <Position> ::= <number>
Environmental Profile
<EnviromentalProfile> ::= <Name> <DurationStay> <PhysicCharact> <CurrentTime>
<PhysicCharact> ::= <noise> <stress> <illumination> <workSpace>
<CurrentTime> ::= <number>:<number>
Device Profile [klyne_composite_2004]
<DeviceProfile> ::= <DeviceType> <GeneralCharact>
<DeviceType> ::= <inDevice> <outDevice> ::= <Format>
<GeneralCharact> ::= <HardwareCharact> <SoftwareCharact> <ApplicationCharact>
<HardwareCharact> ::= <screenSize> <DeviceType> <Memory> <generalSize> <AccessMechanism>
<SoftwareCharact> ::= <name> <version> <supportedApp> <seller>
<AppCharact> ::= <name> <versión> <AppType> <softwareReq.> <hardwareReq>
Task Profile
<Task> ::= <name> <complexityLevel> <Preconditions> <taskType> <taskPresentation> <Commands> {<Activity>}*
<Activity> ::= <name> <Description> ::= <text>

3.3 Interface profile

Figure 4 describes the interface profile. The interface profile explicitly integrates the elements from both HCI and adaptation. From the HCI point of view, this profile includes the primary UI elements (controls, windows, layout, among others). Those elements also

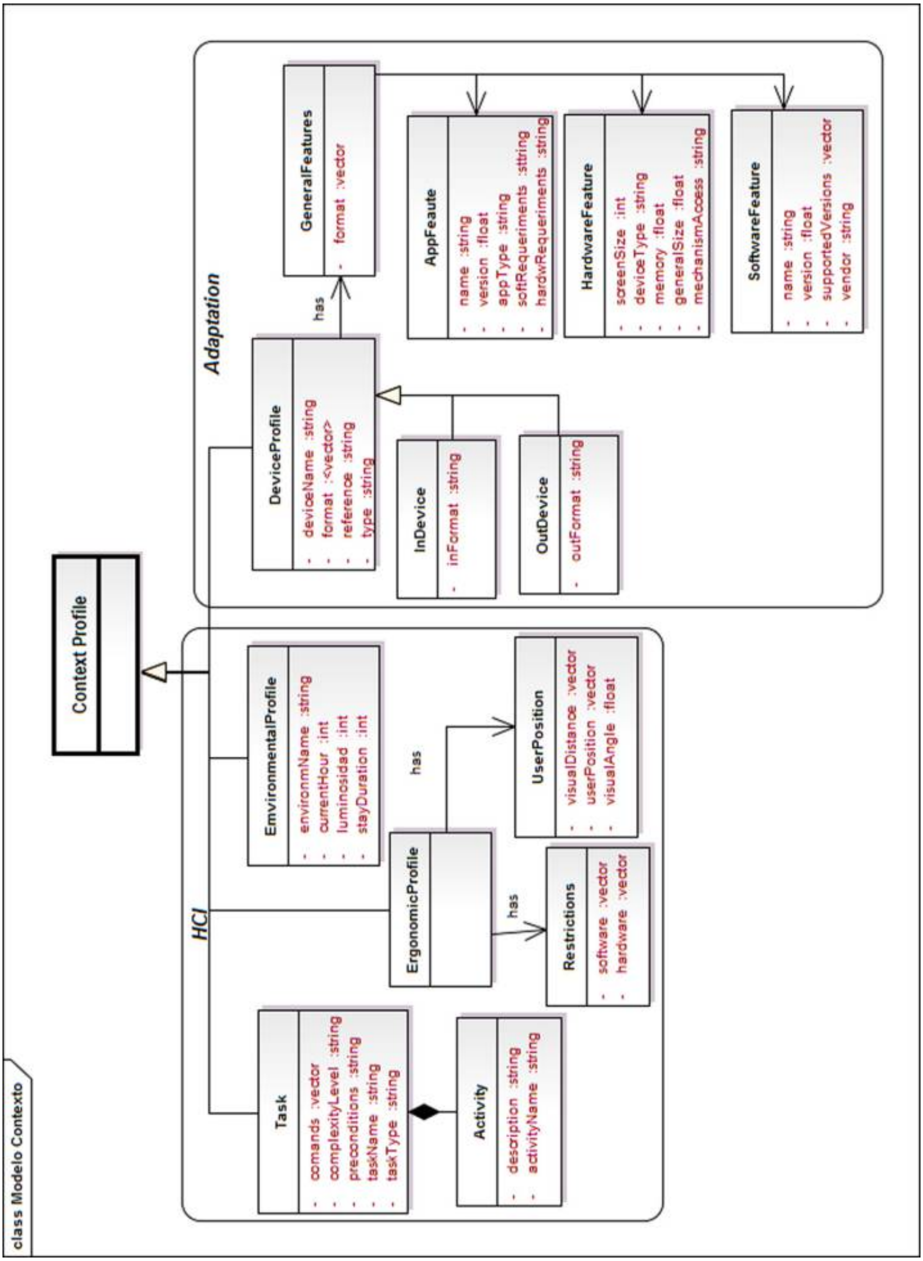

Figure 3.
Context profile

denote their intentionality, i.e. the goals of each interface element with respect to the user (e.g. the close button is associated with a dangerous intentionality).

From the adaptation perspective, this profile includes two concepts:

(1) R**ule** that represents all of the rules required to adapt the interface.

(2) Preferences of the user with respect to the UI design (e.g. buttons or text color).

The following text is BNF (Backus, 1958) formalization of the interface profile:

```
Interface
<InterfaceProfile> ::= <Primitive _GUI> <Layout> <Rule>
Layout
<Layout> ::= {<Primitives_GUI>}* <GenerationRules>
Primitives GUI
<Primitive _GUI> ::= <actualSize> <Dimension> <Position> <maxSize>
<Primitive> <Estate> <MinSize> <Intention>
<Primitive> ::= <Control> | <Window> | <Icon>
<Control> ::= <Button> | <InField>
<Icon> ::= < IconType> <Style>
<Window> ::= <Title> <WindowStyle>
<Intention> ::= <IntentionName> <AllowedColors> <ForbiddenColors>
Rule
<Rule> ::= <Form> <Color> <Preference>
<Form> ::= <baseSize> <adaptedSize> ::= <number>
<Color> ::= <colorIntention > <ColorProfile> <ColorCombination_Allowed>
<ColorCombination_Forbidden>
```

4. Tukuchiy

Tukuchiy is a framework to generate adapted UIs, integrating both adaptation and HCI standards during execution time. This section describes the development of Tukuchiy. Section 4.1 describes the HCI methods used to design the framework. Section 4.2 describes the prioritization of the HCI and adaptation rules. Sections 4.3 to 4.5 describe the design and implementation of the system.

4.1 Design methods

Tukuchiy uses several HCI methods to address adaptation requirements. The following sections describe those methods.

4.1.1 Miller's law (7 ± 2). Miller's (1956) law indicates that people can perform better decisions if the number of stimuli that they receive in the short term is adequate. Empirically, Miller demonstrated that the number is 7 ± 2.

UIs tend to group options, tools and graphical elements that can be used to interact with the application. Tukuchiy generates interfaces taking into account the 7 ± 2 rule to create groups of buttons with similar functionlity, groups of buttons with related functionality and toolbars.

4.1.2 Steering law. Steering law (Accot and Zhai, 2001) establishes that the time a person moves inside a tunnel is inversely proportional to the relative width of the tunnel. To generate UIs with menus, this law is used to guarantee that those menus could be quickly navigated. In practice, this means that a menu should not have more than two sub-levels.

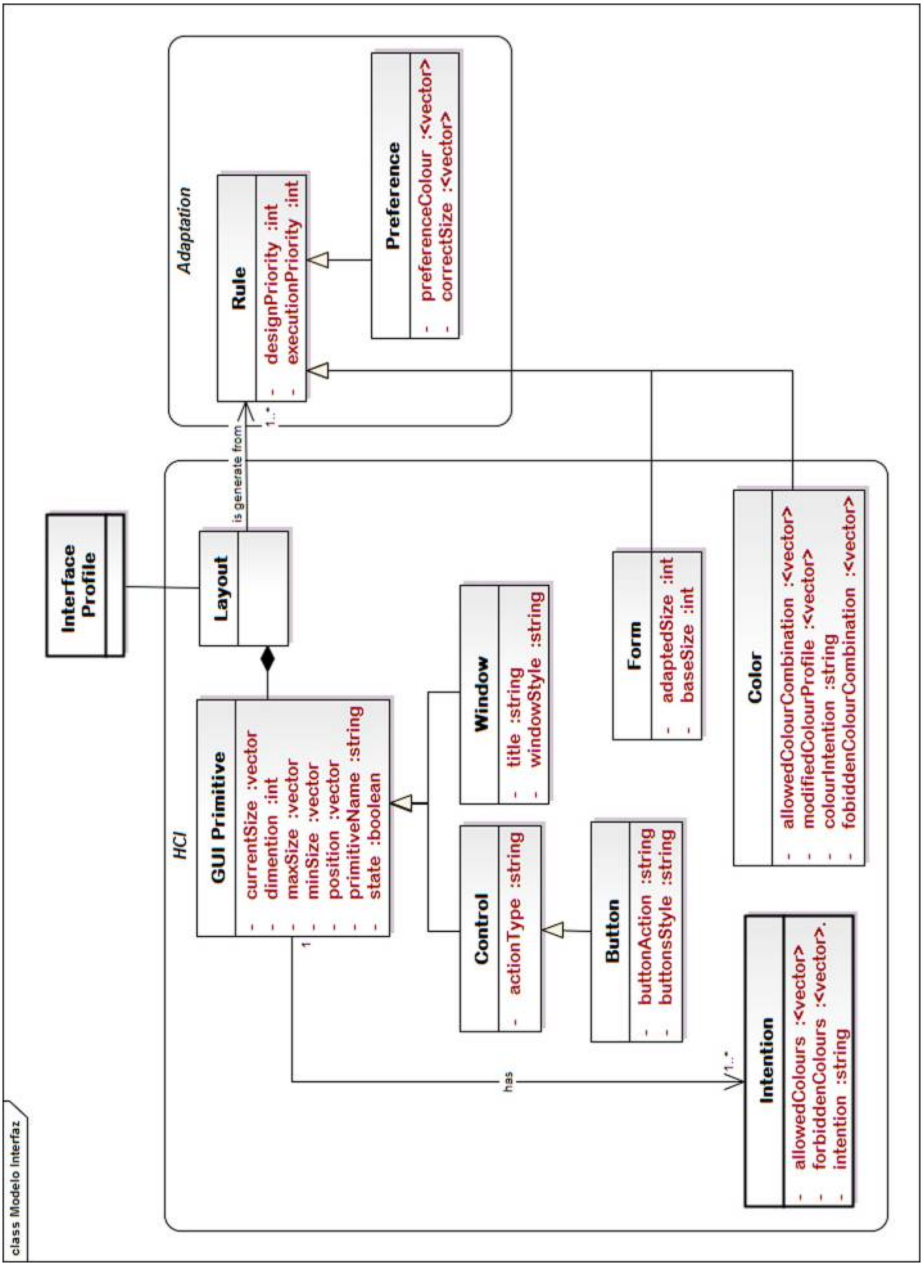

Figure 4.
Interface profile

4.1.3 Hick's law. Hick's law (Seow, 2005) predicts the time a person would take to make a decision, depending on the number of options provided. In practice, this means that UIs should not have too many options per menu or functional group. For instance, if the UI must be used under emergency situations (which require short decision-making times), it is desirable to have few options in the screen to avoid delaying the decisions.

Tukuchiy implements Hick's law by offering a reduced number of options to provide an adequate interaction. This number was based on Miller's law.

4.1.4 Non-functional requirements.

(1) Consistency:
- Graphical elements must be classified in functional groups consistent with business rules.
- Screens with similar functionality must have the same layout graphical elements with common functionality must have the same appearance (layout).
- Graphical elements must comply with the name and coloring conventions of the final user.
- Labels must begin with uppercase letters. Text blocks must be written as sentences with mono-space characters.
- Acronyms must be consistent among screens.

(2) Visual harmony:
- Screens must be balanced.
- Menus must be at most four levels deep.
- Dialogs must have at most 140 characters.
- Dialogs must only be used to guide task executions.
- Tooltips must be used to guide usage of graphical elements.
- Functional groups must have five, seven or nine objects.
- Command buttons must have an intentionality associated and a color consistent with that intentionality.

(3) Tasks:
- Tasks performed by each hand of the user must be specified.
- System commands must be designed according to the physical abilities of the user.

4.2 Execution methods

This section details the HCI rules that are taken into account for the dynamic UI generation during execution time.

4.2.1 Fitt's law. Fitt's law (Seow, 2005) predict the time required to access a goal (e.g. tap, press, select, etc.) with one movement. Tukuchiy uses Fitt's law to determine how long a person will take to select a goal or reach some place in the screen (i.e. move the pointer to some place) in urgent situations or tasks that must be performed quickly. In this specific case, it is not desirable to delay the task.

4.2.2 Practice's law. Practice's law (Roessingh and Hilburn, 2000), also called *power law of practice*, predicts time and speed in which a task would be carried out, based on the number of past intents. The dynamic interface generator applies this law to determine the degree of practice that a user has when using the application. Based on that information, the system offers more or less help (tooltip information) to perform the tasks.

4.2.3 Color rules. Tukuchiy has several color rules that are taken into account during execution. Tukuchiy automatically removes the color from the opposite color palette.

There are color combinations that must be taken into account when generating interfaces. Color combination rules guarantee that graphical elements are presented and visualized adequately. For instance:

- *Background color vs font color*: This rule specifies which background and font colors must not be combined to provide adequate visualization.
- *Background color vs graphical element colors*: This rule specifies the color combinations that must be avoided, as they produce discomfort in the user or may make the user to erroneously interpret the information.

Tukuchiy's color palettes update each time a color is assigned to a button's background or to the text.

Colors are used to denote ideas from the real world. This makes it easier for the user to understand the intentionality of the graphical elements. Examples of these rules are Bedolla Pereda (2002):

- Red is commonly used to indicate dangerous actions or warnings. Red should not be used to indicate tranquility, as the user may not figure out the correct intentionality.
- Green is commonly used to denote that an action was successful or to denote tranquility.
- Blue is commonly used for therapeutic purposes. This color is associated with safety, comfort and amiability. However, some cultures, such as Anglo-Saxon, associates blue with sadness.

The Rule class ensures that Tukuchiy's color palettes contain the adequate colors, according to the intentionality of the graphical component. For instance, if a component's intentionality is "danger", its color will be red. Colors incompatible with that intentionality (green or blue) will be removed from the palette.

4.3 Design

Tukuchiy's design is based on Runa-Kamachiy's adaptation model (Barrera *et al.*, 2013b) (Section 3). Figure 5 shows the architecture of Tukuchiy. Tukuchiy uses the Model-View-Controller architecture, where the model component generates the interfaces; the data component manages users' basic information, preferences and context; the controller component manages events; and the view component displays the basic UI elements. These components are detailed as follows.

4.3.1 Model component. This component performs the required filtering to transform the interface according to the user characteristics. It also manages the user information and updates them when necessary. This component also enforces HCI laws (Section 4.2) during execution time.

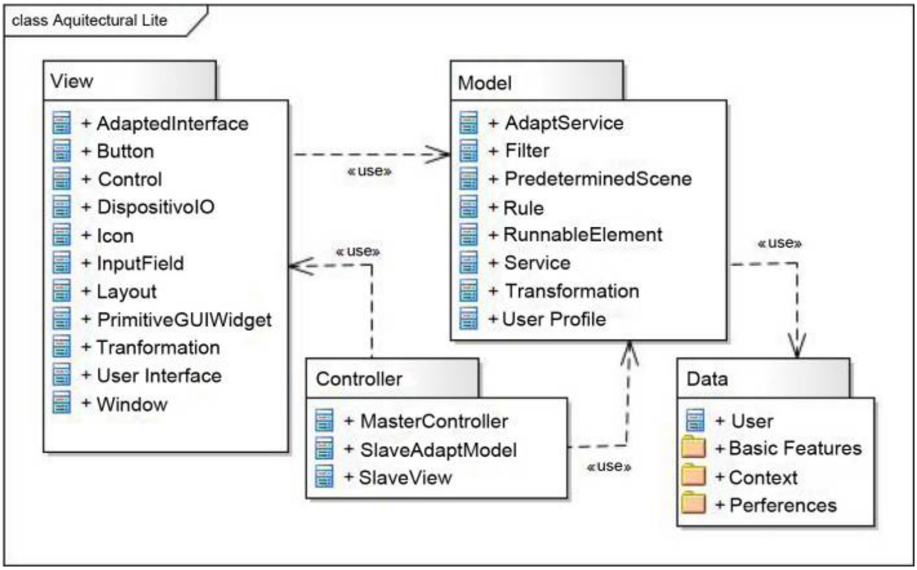

Figure 5.
Tukuchiy's general architecture

4.3.2 Data component. This component represents the user in the system and comprises the following elements:

- basic characteristics – basic user data, such as physical attributes, motivation and emotions;
- preferences, as detailed in Section 3.2; and
- context, as detailed in Section 3.2.

In this version of Tukuchiy, emotional aspects were modeled, but not implemented. This is because collecting emotional data requires using technologies that are outside the scope of this current research (e.g. Face Reader Maat and Pantic, 2006).

4.3.3 View component. This component represents the interface characteristics that could be modified, based on the user and context profiles. Note that the widget organization may change depending on the programming language, widget library or the way to represent the required UI elements.

4.3.4 Controller component. This component manages events of the framework. The controller was developed using Master-Slave pattern, "when a command is received, a slave is launched to execute the command while the master resumes listening for more commands" (S.J.S.U.C. of Science).

4.4 Prioritization rules

The rule prioritization process determines which rules take precedence over other ones, while executing an application. Each rule was analyzed to determine how important it was and the way each rule may affect the execution of other rules:

(1) First level:
   - *HCI*: Effective color combinations.
   - *HCI*: Physical user attributes.
   - *HCI*: Intent of graphical elements.
(2) Second level:
   - *HCI*: Miller (7pm2).
   - *Adaptation*: Help and guidance.
   - *HCI*: Hicks.
(3) Third level:
   - *Adaptation*: Color preferences.

The rules with highest priority are those that guarantee that the information is presented correctly, according to HCI precepts, and that the user can use the essential functionality of the application. The next priority is for rules that improve usability. The least priority is for rules that provide personalization according to user preferences.

The rules grouped in the first level address the effective presentation of information, through adequate color combinations and graphical element adaptation (color and size) to the physical conditions of the user (Wright *et al.*, 1997).

The first rule uses the interface profile to find HCI effective color combinations. Given a base color, the system filters the palette to show only colors that properly match the base color (Bedolla Pereda, 2002). Color combinations are organized pessimistically, managing color lists that do not match the base colors.

The second rule adapts the interface based on physical user attributes to addresses color blindness and myopia. To achieve this, the system provides two processes: color simulation and polarization (Duck), and button enlargement.

The third rule assigns colors according to intent. This is based on the premise that colors are associated to psychological states with specific intentions (e.g. red is associated to danger situations) (Bedolla Pereda, 2002). The system assigns an intent to each color and filters the palette to show the user only colors associated with the particular intent of the user for each use case.

The second level contains rules that improve interface presentation, but which are not essential for the interface functionality. This level comprises Miller's law, help and guidance rules and Hicks' law, as they facilitate the application usage by reducing and organizing options, and providing adequate help according to the user expertise.

The third level comprises color preferences. A rule organizes the color palette according to user preferences, placing the most preferred color first and the least preferred colors last.

4.5 Prototype

Tukuchiy was implemented in C++ and QT 5 (Digia, 2013). Figure 6 is an overview of Tukuchiy's UI. The interface comprises several elements: Element 1 is the toolbar associated to the specific application domain. Element 2 is a tab that includes all of the options associated to Miller's law. Element 3 is an example of a button that has been enlarged based on Fitt's law. Element 4 is a tooltip based on Practice law. Element 5 is the workspace associated to the specific application domain. Element 6 is a dialog to

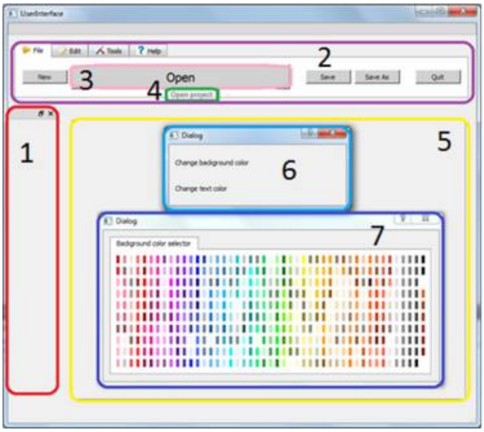

Figure 6.
Tukuchiy's UI

personalize buttons. Element 7 is a dialog with a color palette associated to color laws (Section 4.2.3).

Color palettes are transformed, according to the type of color blindness of the user. Figure 7 shows the available color palettes in Tukuchiy. The first palette is for people without any color blindness. The second palette is for people with tritanopy (inability to differentiate between blue and green and between some types of yellow and violet) (Flück, 2006). The third palette is for protanopic people (cannot perceive red) (Flück, 2006). The fourth palette is for deuteranopic people (cannot discriminate red and green) (Flück, 2006). The latter three palettes provide combinations of colors that are easy to discriminate according to each type of color blindness.

5. Results

To evaluate Tukuchiy, the authors performed conceptual validations and unit tests with the main algorithms used by Tukuchiy. This section describes the results of this evaluation. For more details about the evaluation, the reader can refer to Barrera-Leon (2015a).

5.1 Conceptual validation

The first part of the evaluation is to investigate about other research projects that applied the same laws used by Tukuchiy (Section 4) to properly achieve the usability criteria. The next list describes the main usability hypotheses and the research work in which each hypotheses is validated:

H1. Fitt's law: supports efficiency (Hertzum and Hornbæk, 2007; Guiard *et al.*, 1999).

H2. Fitt's law and Practice law. Support efficacy (doing the right thing) (Lin, 2005).

H3. Hick's law and Practice law: facilitate learning (Johnson *et al.*, 2003).

H4. Fitt's law: supports memorization (Nielsen, 2009).

H5. Practice law: supports error prevention (Johnson *et al.*, 2003).

It is important to note that the authors did not find any research work that demonstrated that the laws directly support the attractiveness criterion or the satisfaction criterion of

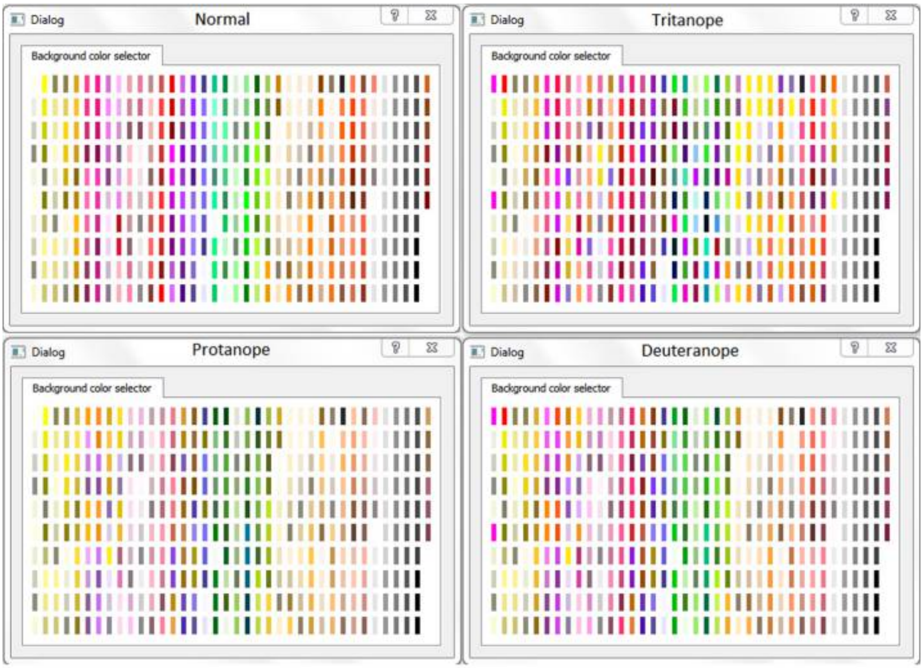

Figure 7.
Different palettes for normal people and people with different types of color blindness

Source: Barrera *et al.* (2013a)

UIs. However, the authors believe that the attractiveness criterion is satisfied when adaptation models take into account user tastes (Lancheros Cuesta *et al.*, 2012), and the satisfaction criterion is satisfied when taking into account preferences (Lancheros Cuesta *et al.*, 2012).

5.2 Functional tests

Functional tests were created to evaluate the functionality of the following algorithms.

5.2.1 Prepare palette algorithm. This algorithm prepares the color palette to personalize graphical elements, ensuring that color combinations are adequate and the intentions of each graphical component correspond to their colors.

5.2.2 Select color algorithm. This algorithm selects a color for a graphical component and, at the same time, forbids the utilization of that color in other parts of the interface that would produce a mismatch. For instance, if green is selected for a button background, this color is forbidden for the button text, as that would make it difficult to read that text.

5.2.3 User profile colors algorithm. This algorithm modifies the application colors to make it easier to visualize by a user with color blindness. If the user has problems of vision scale or focus (e.g. hypermetropia), buttons are enlarged to make them easier to see by the user.

5.2.4 Context change algorithm. This algorithm modifies the application colors, so they can be seen by users, according to the time of day. During night time, colors are made darker. During morning time, colors maintain their default brightness. At noon, colors are brighter, as there is too much natural illumination. This algorithm also analyzes the profile of the task being performed. If it is an important task, the size of the application in the screen will be larger, so the user can better focus on executing that task.

6. Idukay: functional prototype

This section describes Idukay, a functional prototype developed to test Tukuchiy. Idukay is an educational application that provides learning resources ([irtual learning objects (VLO), OVA (Miranda, 2008)], based on the student profile. For more information, the reader can refer to Barrera-Leon *et al.* (2013).

Figure 8 denotes the UI of Idukay. Idukay takes into account the learning style (Felder and Silverman, 1988) and OVA's display preferences, based on additional characteristics provided to the user profile that was implemented in an xml file:

```
<preferences>
<deviced-preferences>
<deploymentCharact/>
<deploymentFormat/>
<replaceFormat/>
<preferenceValue/>
```

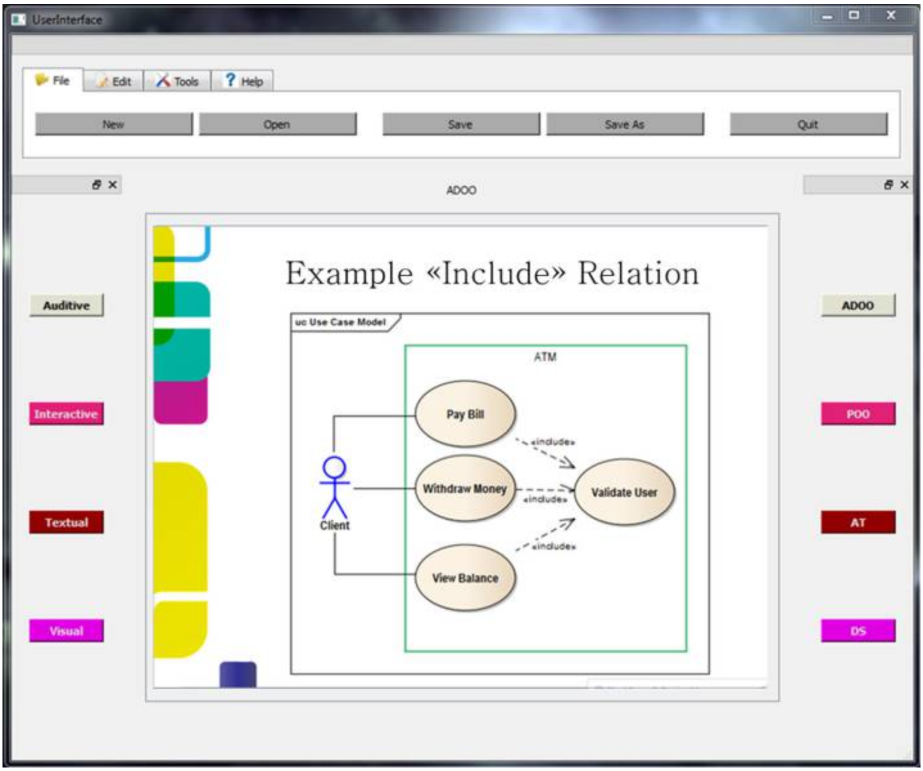

Figure 8.
Idukay's UI

</deviced-preferences>
<presentation-preferences>
<presentationStructure/>
<replacePresentation/>
<preferenceValue/>
<learningStyle>Visual</learningStyle>
<displayPreference>Auditive,Interactive,Textual,Visual</displayPreference>
</presentation-preferences>
</preferences>

Validation tests performed over Idukay showed that the application worked according to expectations. Idukay presents a UI adapted according to the student profile and following the Tukuckiy's HCI rules. The tests also determined that the system is usable.

The above validation was performed with the help of an expert in education and four experts in software development (one of them with knowledge in HCI and another with knowledge in usability and educational applications). The three experts were present during the execution of the tests. They participated as users of the system and provided feedback about the interaction, indicating if the execution of each test yielded the expected results. They also answered a usability questionnaire (Section 6.2).

6.1 Functionality

The system was tested by five experts (four software developers and one education expert). Each expert used the system using four user profiles at different times during the day and verified whether the system behaves as expected.

Different profiles were used to test different scenarios. There were two different scenarios for each profile. The results of the tests indicate that the system behaves in the expected way in all of the scenarios and profiles. Table II shows the expected results for each scenario.

6.2 Usability

Usability of Tukuchiy was measured using the system usability scale (SUS) (Affairs), which has been successfully applied to measure hardware, research projects, Web page usability and for industrial evaluations. In this test, odd-numbered questions evaluate user comfort when using the system. Even-numbered questions measure system discomfort, complexity or user insecurity when interacting with the application (Affairs). The following list shows ten SUS basic assertions that are measured in a scale from one to five, 1 is the lowest level of agreement and 5 is the highest:

(1) I think that I would like to use this system frequently.

(2) I found the system unnecessarily complex.

(3) I thought the system was easy to use.

(4) I think that I would need the support of a technical person to be able to use this system.

(5) I found the various functions in this system were well integrated.

(6) I thought there was too much inconsistency in this system.

(7) I would imagine that most people would learn to use this system very quickly.

(8) I found the system very cumbersome to use.

Topic choice		VLO choice	
Scenario 1	Scenario 2	Scenario 1	Scenario 2
The user can select a topic	The user can select a topic	The user can select an educational resource	The user can select an educational resource
The educational resources are shown in the resource bar	The educational resources are shown in the resource bar	The educational resource is shown in the resource viewer	The educational resource is shown in the resource viewer
The name of the topic is shown above the resource viewer	The name of the topic is shown above the resource viewer	The source where the educational resource was obtained is shown below the resource viewer	The source where the educational resource was obtained is shown below the resource viewer
The visual resource of the topic is shown in the resource viewer	The interactive (kinesthetic) resource of the topic is shown in the resource viewer	The resources are sorted according to the user's learning preferences, in the following order: textual, visual, auditive, interactive (kinesthetic)	The resources are sorted according to the user's learning preferences, in the following order: auditive, interactive (kinesthetic), textual, visual
Given that the user has visual scale issues, the size of the buttons should grow when he brings the cursor close to them	Given that the user has Tritanopia, the colors of the palettes will be adapted to such physical condition	Given that the user has Tritanopia, the colors of the palettes will be adapted to such physical condition	Given that the user has no color blindness conditions, the original colors of the palettes will remain the same
Given that the user has Deuteranopia, the colors of the palettes will be adapted to such physical condition	Given that the user's preferred colors are green and blue, the buttons will be shown prioritized according to those colors and the color palette will also be prioritized according to that	Given that the user's preferred colors are orange and gray the buttons of his learning preferences will be shown according to those colors	Given that the user's preferred colors are purple and yellow the buttons of his learning preferences will be shown according to those colors
Given that the user's preferred colors are green and blue, the buttons will be shown prioritized according to those colors and the color palette will also be prioritized according to that	Given that the time of the day is between 11 am and 1 pm the brightness of the colors will be modified according to the brightness of the environment. (Lighten)	Given that the time of the day is between 2 pm and 4 pm, the brightness of the colors will be modified according to the brightness of the environment. (No change)	Given that the time of the day is between 5 pm and 10 pm, the brightness of the colors will be modified according to the brightness of the environment. (Darken)
Given that the time of the day is between 8 am and 10 am, the brightness of the colors will be modified according to the brightness of the environment. (No change)			

Table II. Expected results for each validation scenario

(9) I felt very confident using the system.

(10) I needed to learn a lot of things before I could get going with this system.

In the SUS questionnaire, a score of 68 corresponds to the average score of the application of 500 tests in previous studies. Each application was classified in a category from A (best usability) to F (worst usability) (Sauro, 2012).

Although SUS evaluates one dimension of usability, which is the perceived ease-of-use by the user, some research works demonstrate that SAS provides results in two satisfaction sub-scales: ease of learning and usability (Lewis and Sauro, 2009). To test Idukay, SAS questions were re-classified according to usability criteria that are specific for the context of Tukuchiy. Table III shows the results of the application of the test, organized by the new classification criteria.

The above test was performed by five experts: an education expert; a computing expert; a computing and adaptation expert; and an expert in computing and HCI. The following sections detail the results of the validation, according to each expert who participated in the evaluation process.

6.2.1 Education expert. The score given by an education expert (92.5) classifies Idukay in the A category, the best possible classification in SUS. According to this expert, the application satisfies most usability criteria. However, Idukay did not obtain perfect score in learning or memorization scores, which suggests that a user from the education area may need technical help or instructions to learn to use the system.

6.2.2 Computing expert. The score given by the computing expert is 87.5. The results suggest that the application satisfy most usability criteria, according the expert. The criterion with the highest score is memorization. The criterion with the lowest score is satisfaction.

6.2.3 Computing and education expert. This expert gave a score of 72.5 to Idukay. According to this expert, the application satisfies several criteria with a reasonable high score, particularly, efficacy and efficiency obtained the highest scores. However, the aspects with the lowest evaluation are related to learning and memorization. The reason for this low evaluation is that the expert has worked with real users in education-oriented applications, and, from her experience, this expert believes that Idukay should provide more assistance for learning and memorization.

More specifically, this expert commented that although the application gives indications and help to the users, it is not sufficient for some people. It is necessary to review which type of people uses the application to better determine what kind of help they need, the amount of information provided by the UI, etc. To address this issue, the authors will include more adaptation rules to change the UI under the above conditions (E.T.S. Institute, 2011).

6.2.4 Computing and adaptation expert. The expert in computing and adaptation gave a score of 92.5 to Idukay. According to this expert, the application satisfies most usability criteria by a high score. Satisfaction has a score lower than the ideal. Efficacy, efficiency, learning and attractiveness are very close to the ideal score. These scores may be explained by the fact that these criteria are interdependent. If one of them lowers, the other criteria are also affected. For instance, if the application has a slight problem in efficacy, efficiency or learning, it will become less attractive to the user.

Question No.	5, 6	2, 5, 8	6, 9	1, 3, 7, 9	3, 4, 7, 10	2, 4, 10	1, 8
Evaluator/Criterion	Efficacy	Efficiency	Error prevention	Satisfaction	Learning	Memorization	Attractiveness
Ideal	8	12	8	16	16	12	8
E1 (Education)	8	12	8	16	13	9	8
E2 (Computing)	7	11	6	1	15	12	7
E3 (Systems engineer, usability, education)	8	12	6	10	8	7	7
E4 (Computing, adaptation)	7	11	8	14	15	12	7
E5 (Computing, HCI)	3	5	3	9	11	7	4

Table III.
Test results, classified by usability criteria

This expert commented that brightness changes according to context should consider the region in which the person is located, which determines ambient luminosity, in addition to time of the day. She also commented to determine learning preferences from time the user interacts with each resource, in addition to the times the user selects that resource. Regarding UI scale, it is important to take into account the zooming tools usage. For instance, if the user frequently uses a zooming tool, it can be inferred that the user has vision problems, without directly asking the user about this problem. Finally, the expert suggested that the system should present the user different predefined color palettes, as the user may feel more comfortable choosing an entire palette, rather than manually select the color of each graphical component.

6.2.5 Computing and human–computer interaction expert. This expert gave a score of 52.5 to the application. This expert gave the lowest scores to the application usability. The lowest ones were learning and error prevention. This suggests that from the HCI point of view there are still issues to be addressed in these aspects to provide better usability. This expert commented about Idukay's functionality and identified the elements that could improve usability. For instance, it is necessary to provide explicit feedback, so the user could understand what is happening with his/her interface. In other words, highlight the changes, their causes and what happens with the interface at every moment of the execution. In addition, it is necessary that the user may undo changes he/she does to the interface, as the user may not be satisfied with those changes. In some cases, people may want to change colors outside the RGB space managed by the application. It would be useful to provide color spaces as an extra option. All of the above could improve the satisfaction and attractiveness criteria. Finally, it is necessary to take into account that different screens display colors differently, so some of the rules to transform palettes according to color blindness may yield unexpected results, depending on the monitor used to display the interface.

6.2.6 Analysis of results. When integrating the results obtained from each expert, one can determine which usability criteria are properly satisfied and which ones are not.

Figure 9 is a chart that integrates the results from every evaluator. The results indicate the following:

- The lowest-scored criteria are learning and memorization. This can be explained by the fact that most experts have experience in the application of adaptive systems and are able to find the systems drawbacks in terms of user learning.

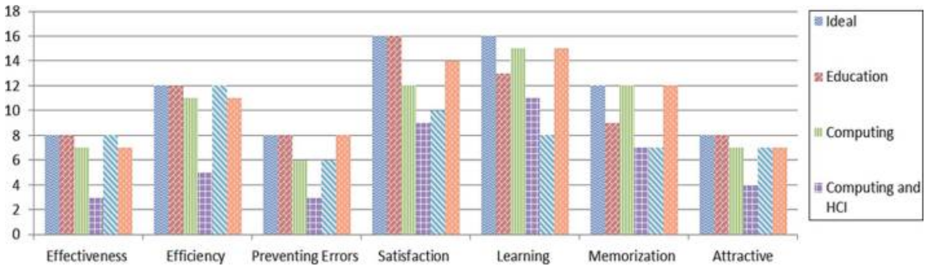

Figure 9.
Usability criteria evaluation

- The highest-scored criteria are efficiency, efficacy, error prevention and attractiveness. This can be explained by the fact that Idukay ensures that adaptation follows the HCI rules. In other words, although users have a degree of freedom to modify the interface, they were constrained by the HCI rules. For instance, if a user wanted to change the color of a button, the available colors were only those that satisfied the intentionality of the button and its relation with the rest of the interface.
- Three out of four experts in computing gave a high score to learning. This can be explained by the fact that these engineers are familiar to software interaction and development. However, the other system engineering expert, who is also expert in adaptation, assigned a low score to that criterion because of her experience in the deployment of educational tools; she was able to easily detect the drawbacks in the application. The education expert, although she gave a good score to learning, was the lowest-scored criterion, as she is not familiar with application utilization and development, so she needed technical help to interact with Idukay.
- Efficacy and efficiency got high scores from most experts. Error prevention was the least-scored by the HCI expert, as he easily detected the user interaction patterns that could yield errors.
- Satisfaction was low-scored by system engineering expert and highly scored by the education expert. This can be explained by the fact that engineers did not clearly see a contribution to their work, while the application was specifically designed to satisfy necessities of the education expert.

7. Conclusion and future work

This paper presented Tukuchiy, a framework that generates adapted UIs based on user and context profiles. Tukuchiy uses HCI standards to ensure usability of interfaces that dynamically change during execution time. The interfaces generated by Tukuchiy adapt to context profile, functionality, color blindness and preferences (usage and presentation) of the user. Tukuchiy enforces specific HCI standards for color utilization, button size and grouping, etc., during execution.

Tukuchiy was validated through a functional prototype for educational applications, called Idukay. Idukay provides virtual learning objects (VLO) according to the student profile. That prototype was tested by experts to validate the degree of support of the framework to different usability criteria. In addition, Tukuchiy was validated through another prototype for clinic radiology. The results were published in Barrera *et al.*, 2014).

The results suggest that Tukuchiy may improve usability of applications during execution time and that are several opportunities for improvement. To improve the application's knowledge about users, ongoing work is determining color sensory properties that could be utilized by applications to take into account psychological and emotional aspects of users. To improve Tukuchiy's validation, ongoing work is developing another prototype to support diagnostics based on medical imaging, particularly those obtained from clinical radiology.

Note

1. The name "Runa-Kamachiy" is based on the Quechua language. "*Runa*" means "person" and "*Kamachiy*" means "to adapt".

References

Accot, J. and Zhai, S. (2001), "Scale effects in steering law tasks", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'01*, *ACM, New York, NY*, pp. 1-8.

A.S. Affairs (2013), "System usability scale (SUS)", The system usability scale (SUS) is a reliable tool for measuring the usability. It consists of a 10 item questionnaire with five response options for respondents; from strongly agree to strongly disagree.

Backus, J.W. (1958), "The syntax and semantics of the proposed international algebraic language of the zurich ACM-GRAMM CONFERENCE", *Proceedings of the International Conference on Information Processing*, *UNESCO, Paris*, pp. 125-132.

Barrera, L.F., Mejia-Molina, N., Carrillo-Ramos, A. and Flórez-Valencia, L. (2013a), *Tukuchiy – Color Palettes*.

Barrera, L.F., Pavlich-Mariscal, J.A., Carrillo-Ramos, A. and Flórez-Valencia, L. (2013b), "Runa-Kamachiy: conceptual integration model between HCI and adaptation oriented to user interface usability", *DYNA -Revista de la Facultad de Minas de la Universidad Nacional de Colombia. Sede Medellin*, p. 10.

Barrera, L., Carrillo-Ramos, A., Florez, L., Pavlich-Mariscal, J. and Mejía, N. (2014), "Integrating adaptation and HCI concepts to support usability in user interfaces: a rule-based approach", *Proceedings of the 10th International Conference on Web Information Systems and Technologies*.

Barrera-Leon, L., Mejia-Molina, N., Carrillo-Ramos, A. and Flórez-Valencia, L. (2013), "Tukuchiy-Idukay: generación dinámica de interfaces en contextos educativos", *CAVA 2013*, San Juan.

Barrera-Leon, L.F. (2015a), "Runa-Kamachiy: documentos", available at: http://pegasus.javeriana.edu.co/PI121-05-HCI-AdaptUser/Documentos.html

Barrera-Leon, L.F. (2015b), "Runa-Kamachiy: Modelo de Integración Conceptual entre HCI y Adaptación orientado a la Usabilidad de Interfaces de Usuario", September.

Bedolla Pereda, D. (2002), "Diseño sensorial. Las nuevas pautas para la innovación, especialización y personalización del producto", PhD thesis, Universitat Politècnica de Catalunya, Departament de Projectes d'Enginyeria.

Card, S.K., Newell, A. and Moran, T.P. (1983), *The Psychology of Human-Computer Interaction*, L. Erlbaum Associates, Hillsdale, NJ.

Carroll, J.M. (2013), "Human computer interaction (HCI)", *The Encyclopedia of Human-Computer Interaction*, 2nd ed.

Carvajal, M. and Saab, J. (2010), "Fundamentos conceptuales de las directrices de usabilidad de gobierno en línea", Technical Report, Ministerio de Tecnologías de la Información y Comunicaciones, Bogotá.

Calvary, G., Coutaz, J. and Thevenin, D. (2000), "Embedding plasticity in the development process of interactive systems", *6th ERCIM Workshop "User Interfaces for All". Also in HUC (Handheld and Ubiquitous Computing) First Workshop on Resource Sensitive Mobile HCI, Conference on Handheld and Ubiquitous Computing*.

Criado, J., Vicente-Chicote, C., Padilla, N. and Iribarne, L. (2010), "A model-driven approach to graphical user interface runtime adaptation", Vol. 641, pp. 49-59.

Duck, D. “Daltonism: making games color blind friendly”, *SebbyLive-My Life, My Projects*.

Digia (2013), available at: http://qt-project.org/wiki/Qt_5.0

E.T.S. Institute (2011), “Human factors: inclusive eservices for all: optimizing the accessibility and the use of upcoming user-interaction technologies”.

Felder, R.M. and Silverman, L.K. (1988), “Learning and teaching styles in engineering education”, *Engineering Education*, Vol. 78 No. 7, pp. 674-681.

Flück, D. (2006), *Color Blind Essentials*, Vol. 1, Zürich.

Guiard, Y., Beaudouin-Lafon, M. and Mottet, D. (1999), “Navigation as multiscale pointing: extending fitts’ model to very high precision tasks”, *Proceedings of the SIGCHI conference on Human Factors in Computing Systems, CHI’99, ACM, New York, NY*, pp. 450-457.

Hertzum, M. and Hornbæk, K. (2007), “Input techniques that dynamically change their cursor activation area: a comparison of bubble and cell cursors”, *International Journal of Human-Computer Studies*, Vol. 65 No. 10.

Johnson, E.J., Bellman, S. and Lohse, G. (2003), *Cognitive Lock-in and The Power Law of Practice*, Social Science Research Network, Rochester, NY.

Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M.H. and Tran, L. (2004), “Composite capability/preference profiles (CC/PP): structure and vocabularies 1.0”.

Lancheros Cuesta, D., Carrillo-Ramos, A. and Pavlich-Mariscal, J.A. (2012), “Características de adaptación y discapacidad a tener en cuenta en entornos virtuales de aprendizaje”, *DYNA*, Vol. 79, pp. 6-14.

Lewis, J.R. and Sauro, J. (2009), “The factor structure of the system usability scale”, *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009, HCD 09, Heidelberg, Berlin*, pp. 94-103.

Lin, M., Price, K., Goldman, R., Sears, A. and Jacko, J. (2005), *Tapping on the Move – Fitts’ Law under Mobile Conditions*, San Diego, CA, pp. 132-135.

Maat, L. and Pantic, M. (2006), “Gaze-x: adaptive affective multimodal interface for single-user office scenarios”, *Proceeding of ACM International Conference Multimodal Interfaces, ICMI ’06, ACM, New York, NY*, pp. 171-178.

Miller, G.A. (1956), “The magical number seven, plus or minus two: some limits on our capacity for processing information”, *Psychological Review*, Vol. 63 No. 2, pp. 81-97.

Miranda, J.J.A. (2008), “Uso educativo de medios: objetos virtuales de aprendizaje – ovas”.

Moussa, F., Kolski, C. and Riahi, M. (2000), “A model based approach to semi-automated user interface generation for process control interactive applications”, *Interacting with Computers*, Vol. 12, pp. 245-279.

Namgoong, H., Sohn, J.-C., Cho, Y.-J. and Chung, Y.K. (2006), “An adaptive user interface in smart environment exploiting semantic descriptions”, *IEEE International Symposium on Consumer Electronics*, IEEE, St. Petersburg, pp. 1-6.

Nielsen, J. (1995), *Usability 101: Introduction to Usability*.

Nielsen, J. (2009), “Short-term memory and web usability”, *Nielsen Norman Group*, December.

Parker, S. (1984), *McGraw-Hill Dictionary of Engineering*, 6th ed., McGraw-Hill Book Company, New York, NY.

Roessingh, J.J.M. and Hilburn, B.G. (2000), *The Power Law of Practice in Adaptive Training Applications*, 1st ed., Nationaal Lucht- en Ruimtevaartlaboratorium, Amsterdam.

Sauro, J. (2012), “Measuring usability with the system usability scale (SUS)”.

Seow, S.C. (2005), "Information theoretic models of HCI: a comparison of the hick-hyman law and fitts' law", *International Journal of Human-Computer Interaction*, Vol. 20, pp. 315-352.

S.J.S.U.C. of Science, "Master-slave pattern".

Tran, V. (2010), "UI generation from task, domain and user models: the DB-USE approach", *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '10*, ACM, New York, NY, pp. 353-356.

Treviranus, J., Mitchell, J., Clark, C. and Roberts, V. (2014), "An introduction to the floe project", *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge*, Springer, pp. 454-465.

Wright, P., Mosser-Wooley, D. and Wooley, B. (1997), "Techniques and tools for using color in computer interface design", *Crossroads, The ACM Student Magazine*, Vol. 3, pp. 3-6.

Zimmermann, G., Jordan, J.B., Thakur, P. and Gohil, Y. (2013), "GenURC: generation platform for personal and context-driven user interfaces", *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, W4A '13, ACM, New York, NY*, pp. 1-6.

Corresponding author
Luisa Barrera-León can be contacted at: luisa.barrera@javeriana.edu.co