Journal of Universal Computer Science, vol. 21, no. 4 (2015), 526-560
submitted: 14/3/14, accepted: 26/3/15, appeared: 1/4/15 © *J.UCS*

RITA: a useR Interface evaluaTion frAmework

Selem Charfi
(LAMIH-UMR CNRS 8201, UVHC, Valenciennes, France
& Computer Science Departement,
University of Pau and Pays de l'Adour, 64000, Pau, France
Selem.Charfi@univ-pau.fr)

Houcine Ezzedine
(LAMIH-UMR CNRS 8201, UVHC, Valenciennes, France
Houcine.Ezzedine@univ-valenciennes.fr)

Christophe Kolski
(LAMIH-UMR CNRS 8201, UVHC, Valenciennes, France
Christophe.Kolski@univ-valenciennes.fr)

Abstract: Interactive systems are constantly evolving. This evolution leads to new challenges. One of these challenges pertains to the quality of dialogue between interactive systems and humans. This dialogue essentially takes place through user interfaces. User interface evaluation is essential to improve communication between a system and its users. The research on interface evaluation is plentiful. Nevertheless, user interface evaluators still encounter difficulties. Therefore, in this article we suggest expanding the functionalities of existing evaluation tools by proposing a user interface evaluation framework. This framework is composed by software applications structured following a modular architecture. It is based on three different evaluation techniques and has a modular architecture that can be configured to evaluate different user interfaces. After being presented, this framework is tested on a network supervision system for a project in the transportation domain. The main advantages of the presented Framework are the following: (1) the guidelines are not hard coded into the evaluation engine; (2) it is based on three different evaluation techniques to avail further data for the evaluation; (3) it is structured following a modular architecture to enable flexible and configurable use; (4) interaction data are automatically captured and analyzed; and (5) the framework is intended for the evaluation of different kind of user interfaces.

Keywords: user interface evaluation, electronic informer, questionnaire, ergonomic quality inspection, ergonomic guidelines, utility, usability
Categories: H.5.2

1 Introduction & Background

"*I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone*."

Bjarne Stroustrup[1]

This quote illustrates the complexity of interactive systems that operate on different platforms (e.g., PCs, touch tablets, Smartphones, and mobile phones).

---

[1] Bjarne Stroustrup is a Danish computer science professor and the author of the C++ programming language.

Today, interactive systems are used in all fields and sectors. Computers turns from calculating machines to tools for assisting users in various tasks. Then, their use became more complex and less convenient. The associated user interfaces (UI) evolve from console interfaces in order to handle the inputs and the outputs to complex interfaces including further information and functionalities. This creates new challenges in promoting techniques for designing useful and usable user interfaces [Schneiderman, 00; Nielsen, 93].

Note that usability is defined by ISO standard 9241-11 as follows: "*The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*" [ISO, 98]. A system is said useful when it respects the utility factor. The utility (or usefulness) is defined by MacDonald and Atwood (2014) as "*the extent to which a system's functions allow users to complete a set of tasks and fulfill specific goals in a particular context of use*". The definition of utility closely mirrors the usability ones. This is due to the close relationship between utility and usability. In [Hornbæk, 06], the author considers the utility as a system's appropriateness for a specific context and the usability as its effectiveness, efficiency and satisfaction within that context.

This challenge is intensified mainly in error sensitive application areas such are the cases of transportation, healthcare, and military fields. The HCI community has been very active during the past three decades by proposing different recommendations, best practices, processes, and tools for better designing UI. It has been proposing also many contributions for UI evaluation [Karat, 94, Stone, 05]. Although the quality and the quantity of these contributions, there is not any consensus on a universal definition about UI evaluation. For instance, Senach (1990) defines the evaluation as a comparison between a referential and an observed task models in order to deduce conclusions about the evaluated UI. Other works consider the evaluation as a knowledge related to software ergonomics (i.e. ergonomic guidelines and recommendations) validation, in conjunction, with the interactive system to evaluate such is the case of [Charfi, 12; Farenc, 96; Vanderdonckt, 05]. In this case, the evaluation consists of ensuring the UI compliance in relation to the recommendations and guidelines issued from software ergonomics. Preece et al. (1994) define the evaluation as gathering usability data of a product by a specific group of users for a particular activity within a specified group of users or work context.

The related tools are numerous. They mainly aim to identify aspects that may generate usability problems[2] [Nielsen, 93; Ivory, 01] and minimize error risk [Wickens, 04]. They also attempt to improve the user acceptability of interactive systems [Zhang, 99]. Globally, UIs are designed to improve and even optimize the efficiency and productivity of interactive systems [Zhang, 99]. In this paper, we suggest introducing a new UI evaluation tool. This tool consists of a framework[3]. It is composed by software applications structured following a modular architecture. It is called "RITA[4]" for useR Interface evaluaTion frAmework.

---

[2] A usability problem is defined by [Lavery, 97] as "*an aspect of the system and / or a demand on the user which makes it unpleasant, inefficient, onerous or impossible for the user to achieve their goals in typical usage situations.*"

[3] The term framework is used to refer to a software suite: a collection of software programs that inter-communicate through XML files. The reason behind this choice is the fact that the evaluator can use a set of these programs for the evaluation.

The remainder of this article is structured as follows: In the first section, we present a state of the art of existing UI evaluation tools. The proposed framework is presented in the second section. The third section includes the process adopted for UI evaluation. The fourth section is comprised of a case study of the proposed framework as applied to a transportation network supervision system. This article ends with a conclusion and suggestions for future research.

2 Related research: existing UI evaluation tools

Ivory and Hearst (2001) structure the evaluation process into three phases: capture, analysis, and critique. Each phase can be automated, semi-automated or manual. As in the majority of evaluation tools, the capture and analysis phases' tools execute automatically. The critique phase is generally performed manually by the evaluator. This may be due to the fact that the evaluation requires not only quantitative measures but also qualitative judgments from one or more evaluators [Rubin, 08; Ivory, 01].

Rohrer (2014) identified the different evaluation methods through three dimensions: (1) Attitudinal vs. Behavioral; (2) Qualitative vs. Quantitative; and (3) Context of Use. The qualitative based methods focus on the users' behviour and/or attitudes. The quantitative based methods are generally derived from mathematical analysis since the instruments of data capture captures large amounts of data. The qualitative and quantitative data are respectively measured directly (through observations) and indirectly (through measures). According to [Rohrer, 14], qualitative methods are much better suitable for answering questions about why and how to fix a use problem, whereas quantitative methods are used to determine how many and how much types of questions. We estimate that coupling between these two types of data is doubtedly interesting for better evaluation results.

There are many UI evaluation tools [Ivory, 01; Nielsen, 94]. These tools differ in several ways. Their applications differ with interactive system lifecycle. Some tools and approaches are intended for use in relation with the phases of specification and design. Such evaluation tools are known as early evaluation tools. Examples of evaluation tools supporting the evaluation since the design phase include Aspect-Oriented Programming (with aspects that focus on evaluation) [Tarby, 09] and evaluation widgets (including self-evaluation functions) [Charfi, 14]. Most evaluation tools can only be used to assess final systems or advanced prototypes; examples include AWebHUT [Rukshan, 11] and EISEval [Tran, 08]. Furthermore, these tools vary according the automation level of the adopted evaluation process.

The evaluation tools also differ with the nature of the system being evaluated, e.g., WIMP (Windows, Icons, Menus, Pointing device), Web or Mobile UIs. Moreover, evaluation tools vary with the principle adopted for the evaluation. Some evaluation tools use a comparison between a referential model and an observed task model to draw conclusions [Senach, 90]. This is generally the case for electronic informers intended for UI evaluation [Tran, 08]. The adopted evaluation basically involves comparing a sequence of elementary user-performed actions (observed task model) to another sequence of system designer-planned actions to perform a task (referential model). Other tools consider an evaluation as a verification of the compliance of UIs with software ergonomics recommendations [Abascal, 04; Leporini, 08].

---

[4] http://www.rita-framework.com/

Evaluation tools can be classified according to the aspects of the evaluation as well. There are three categories:

- The first one pertains to dynamic evaluation. It is based on analyzing and interpreting the interaction between a user and the evaluated system. The main advantage of the related tools is that their results are generally considered reliable because they provide numerical data generated by real activities and they involve users in the evaluation process. Examples include the EISEval [Tran, 08] and Sherlock [Grammenos, 00] electronic informers.
- The second category includes tools that focus on static information presentations of graphical UIs. They do not consider the user during the evaluation process. The related tools are numerous. Examples include Synop [Kolski, 91] and AccessEnable [Brinck, 02]. The tools in this category are mainly based on software ergonomics research. They generally incorporate ergonomics guidelines (EG) in their design and in the UI evaluation. The main advantage of these tools is the low cost in terms of time and required material resources of the evaluation process in question. However, the selection of the guidelines is a delicate task. In fact, although guidelines can provide helpful pieces of information, they suffer from their disconnected nature for a given UI design. It can be difficult in many cases to determine which guidelines should be followed [Henninger, 97].
- The third category focuses on the users' appreciation toward the evaluated interactive system. Generally, a questionnaire is used to collect user appreciation. Examples include SUMI (Software Usability Measurement Inventory) [KiraKowski, 96] and SUS (System Usability Scale) [Brooke, 96].

Generally, each of the existing tools utilizes one evaluation technique at a time. Note that every technique has its own specificity and deals with aspects different of other techniques ones [Nielsen, 93]. Then, the provided evaluation results cover a larger image of the aspects in question. The evaluation is only of user interaction, the static presentations of the graphical UI or a user appreciation toward the UI in question. Many authors recommend using more than one evaluation technique to get better evaluation results [Nielsen, 93; Rubin, 08].

In addition, the collected data are generally difficult and costly to process. For instance, some mouse tracking tools provide statistics and information on user clicks and mouse pointer movement during user-interface interactions. The evaluator is then confronted with a considerable amount of information but does not have any tools or methods to process the information [Schmettow, 08; Nielsen, 93].

More, the tools following EG for UI evaluation generally hard code these EG into the evaluation engine. Consequently, the evaluator cannot use EG other than those proposed by the evaluation tool. It is recommended to separate the guidelines from the evalution engine [Vanderdonckt, 05]. Another limit pertains to the fact that each existing tool is generally employed for a specified kind of UI, such as WIMP, Web or Mobile. The majority of existing tools are used only for Web UI evaluation such is the case of Destine [Mariage, 04]. This is undoubtedly due to the fact that it is much easier to access an interactive system source to gather information about interface control attributes than other UI types [Beirekdar, 02].

During the study of the existing evaluation tools, we consider the following criteria for the evaluation:

- The type of supported UI: WIMP, Web or Mobile UI,
- The type of the evaluation tool: software, web service, etc.
- The adopted evaluation principle: ergonomic guidelines validation, mouse tracking tools, etc.
- Provided evaluation report: a range of utility and usability problems identified in the evaluated user interface, information about the evaluated UI quality and even suggestions for correcting and improving it, simply some statistics or heat maps about the interaction, etc.
- Type of the captured events: high or low event level;
- Stakeholders involved into the evaluation process: final or experimental user(s), evaluator, designer, etc.
- Systems development life-cycle phase: in which the tool is applied for the evaluation;
- The automoation level of the supported evaluation process: manual, semi-automated or automated;
- The supported features by the evaluation tool: the identification of UI design problems, automatic or assisted UI repair, user action statistics, heat maps;
- The used technique for data collection: log files, event handler, proxy, screen shots, questionnaire, etc.

Finally, the major motivation of this study is to expand the range of functionalities and the scope of existing evaluation tools by proposing an evaluation framework. This framework is presented in the following section.

3 RITA: a useR Interface evaluation frAmework

3.1 Motivation & Overview

Most existing tools for the evaluation of UIs are essentially based on a single evaluation technique. Each technique has its own evaluation characteristics and incorporates different aspects of other techniques. As mentioned earlier, it is interesting to exploit several evaluation techniques to evaluate a UI [Nielsen, 93]. This is the main reason for establishing framework based on various evaluation techniques. Furthermore, each evaluation tool is generally used to evaluate a given type of UI, such as WIMP, Web or Mobile. Therefore, a tool that can evaluate using several techniques is of great interest. This paper aims to expand the range of functionalities for existing UI evaluation tools by proposing a framework that is:

- Generic: the framework is intended for the evaluation of different interactive systems. It supports the evaluation of WIMP, Web and Mobile UIs.
- Configurable: the framework is structured according to a modular architecture and can be configured to evaluate different UIs.
- Flexible: the proposed framework follows EG for UI evaluation. These EG are not hardcoded into the evaluation engine. RITA codes the EG as external

XML files. Subsequently, the evaluator can add new EG or modify existing ones.

- Multi-evaluation techniques: the tool relies on (1) the evaluation of the UI static presentations, (2) the ensured interaction between users and the system, and (3) the appreciation of the user toward the evaluated UI in question. In order to ensure that such an evaluation takes place, RITA exploits three evaluation techniques:
  - Electronic Informer: the tool collects information about real time user actions to analyze them and assist the evaluator in identifying interface use problems [Ezzedine, 08]. This technique is selected since it enables us to collect many numerical data about the interaction that we can easily interpret and use for the evaluation
  - Ergonomic quality inspection: the tool validates the compliance of the interface with a set of EG to detect design inconsistencies and ensure that the UI is compliant with these criteria [Charfi, 14]. This technique is selected since it enables configuring the selected EG for the evaluation in order to check the compliance of specific EG. It is implemented in our Framework considering the EG as not hard coded into the evaluation engine.
  - The questionnaire: about user's appreciation and/or inspect the aspects it was impossible to inspect using the two other evaluation techniques, such as error message clarity [McNamara, 11]. As mentioned earlier (Section 2), the qualitative based evaluation method enables the evaluator to mainly ask the user on why and how to fix a problem in the UI.

UIs are evaluated through RITA once the system has already been implemented or, in the case of an advanced prototype, during the final interactive system design phase (test phase).

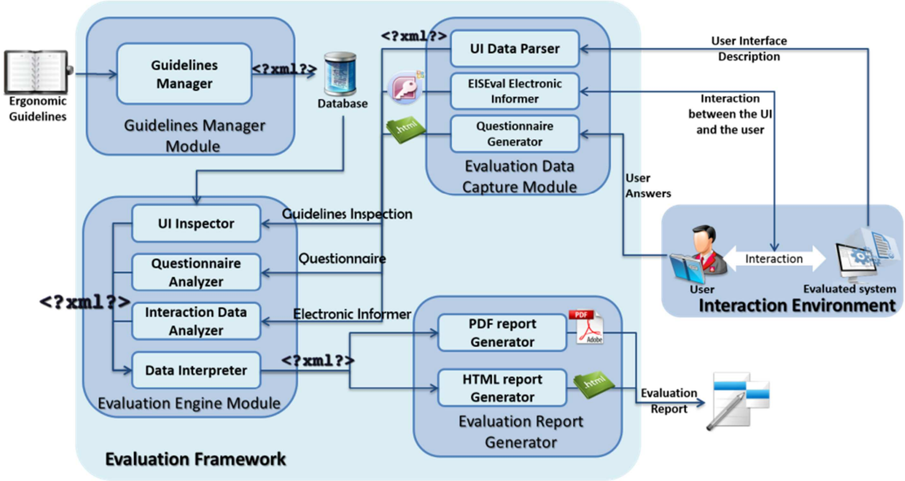

Ergonomic Guidelines
Guidelines Manager
<?xml?>
Database
Guidelines Manager Module
<?xml?>
UI Data Parser
EISEval Electronic Informer
Questionnaire Generator
Evaluation Data Capture Module
User Interface Description
Interaction between the UI and the user
User Answers
UI Inspector
Guidelines Inspection
Questionnaire Analyzer
Questionnaire
<?xml?>
Interaction Data Analyzer
Electronic Informer
Data Interpreter
<?xml?>
Evaluation Engine Module
Interaction
User
Evaluated system
Interaction Environment
PDF report Generator
HTML report Generator
Evaluation Report Generator
Evaluation Report
Evaluation Framework

Figure 1: RITA Framework Functional Architecture [Charfi, 13]

3.2 Functional Architecture of the RITA Framework

As mentioned above, RITA has a modular architecture. It includes four modules, as shown in Figures 1 and 2: (1) Ergonomic Guidelines Manager, (2) Evaluation Data Capture Module, (3) Evaluation Engine, and (4) Evaluation Report Generator.

These modules are described below. RITA inputs are provided according to the EGs being taken into consideration and the UI source. The RITA outputs include an evaluation report that is produced in the fourth module. The evaluation process has three phases: the first involves preparing data for the evaluation. The second pertains to the evaluation itself. The third is comprised of establishing the evaluation report. These phases are described in more detail in section 4.

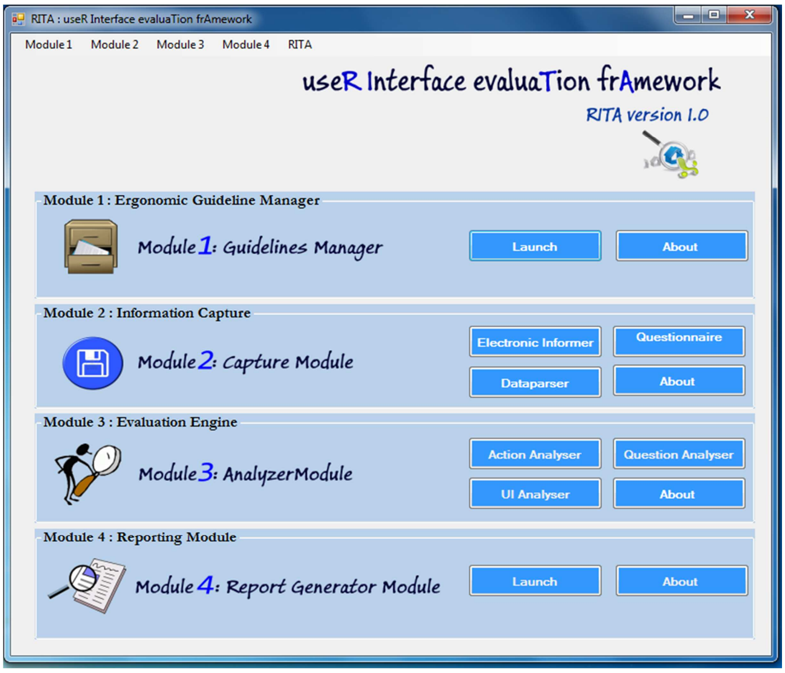

RITA : useR Interface evaluaTion frAmework
Module 1
Module 2
Module 3
Module 4
RITA
useR Interface evaluaTion frAmework
RITA version 1.0
Module 1 : Ergonomic Guideline Manager
Module1: Guidelines Manager
Launch
About
Module 2 : Information Capture
Module2: Capture Module
Electronic Informer
Questionnaire
Dataparser
About
Module 3 : Evaluation Engine
Module3: AnalyzerModule
Action Analyser
Question Analyser
UI Analyser
About
Module 4 : Reporting Module
Module4: Report Generator Module
Launch
About

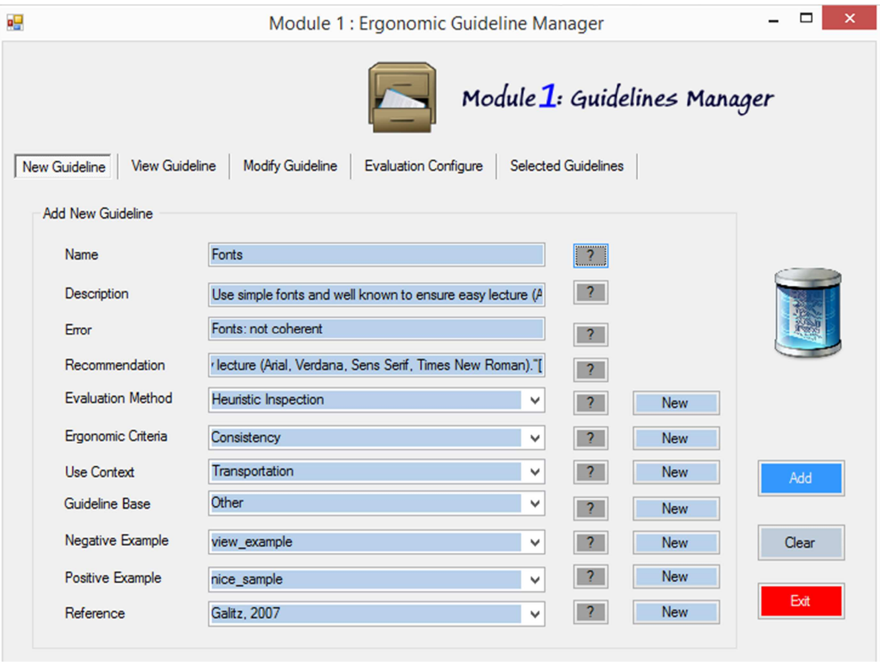
Figure 2: A screenshot of the main UI of the RITA Framework

3.2.1 Guidelines Manager Module

Since RITA does not hardcode the EG for UI evaluation, the evaluator must provide RITA with a set of EG. This module allows the user to manage EG, as shown in Figure 1. It assists evaluators in managing a guidelines database of XML files, as shown in Figure 3. An EG is defined by: (1) a numerical identifier, which is attributed automatically to new EGs, (2) a general description of the EG, (3) the associated errors and recommendations in the case of noncompliance with the EG (the error and the recommendation will be used for the report by integrating the detected errors and suggesting the improvement recommendations), (4) the related ergonomic criteria and subcriteria to which the EG belongs in this tool were taken from [Bastien, 99], (5) the context of use of the EG, where context is defined by the <user, environment,

platform>triplet [Calvary, 03], (6) a positive example and a negative example to illustrate the UI's compliance with the EG, (7) the type of interface to which the EG is applied (WIMP, Web or Mobile), (8) the author(s) of the EG and the source from which the EG is extracted.

To facilitate the UI evaluation in our framework, we opted for EG modeling that is simpler than the broader one proposed in [Mariage, 05].

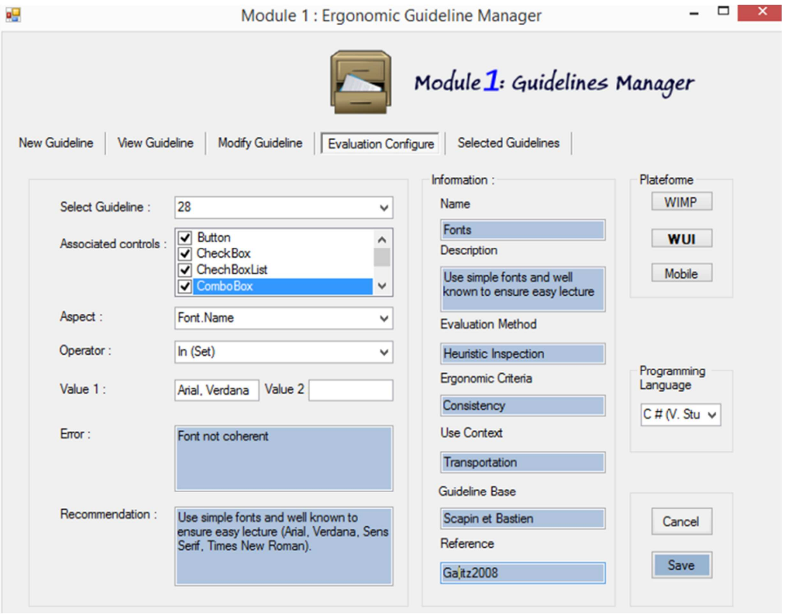

Module 1 : Ergonomic Guideline Manager
Module 1: Guidelines Manager
New Guideline
View Guideline
Modify Guideline
Evaluation Configure
Selected Guidelines
Add New Guideline
?
New
Add
Clear
Exit

Name	Fonts
Description	Use simple fonts and well known to ensure easy lecture (A
Error	Fonts: not coherent
Recommendation	lecture (Arial, Verdana, Sens Serif, Times New Roman)."[
Evaluation Method	Heuristic Inspection
Ergonomic Criteria	Consistency
Use Context	Transportation
Guideline Base	Other
Negative Example	view_example
Positive Example	nice_sample
Reference	Galitz, 2007

Figure 3: EG Manager Module: interface for adding new guidelines

In Figures 3 and 4, an example is provided, from: "*Use simple fonts and well known to ensure easy lecture (Arial, Verdana, Sens Serif, Times New Roman).*"[Galitz, 07]. In Figure 3, this EG is entitled, "*Fonts*". Its statement is used for description and recommendation. The associated error is "*Font: not coherent*". It is associated with the "*Heuristic Inspection evaluation method*" and related to the "*consistency*" ergonomic criterion. Figure 4 demonstrates how it is defined for inspection in UIs. An identifier (*ID, value: 28*) is attributed to the EGs. This EG is applied for the entire graphical controls list. It deals with "*Font.Name*". It uses the operator "*in (set)*" (i.e.: *"Arial, Verdana, Sens Serif, Times New Roman"*).

In Figure 4 we also see the specified aspect mentioned again later on to the evaluator (e.g., Name, Description, Evaluation method, Ergonomic Criteria).



Select Guideline : 28

Associated controls :
☑ Button
☑ CheckBox
☑ ChechBoxList
☑ ComboBox

Aspect : Font.Name

Operator : In (Set)

Value 1 : Arial, Verdana Value 2

Error : Font not coherent

Recommendation : Use simple fonts and well known to ensure easy lecture (Arial, Verdana, Sens Serif, Times New Roman).

Information :

Name: Fonts

Description: Use simple fonts and well known to ensure easy lecture

Evaluation Method: Heuristic Inspection

Ergonomic Criteria: Consistency

Use Context: Transportation

Guideline Base: Scapin et Bastien

Reference: Galitz2008

Programming Language: C # (V. Stu

Figure 4: Guidelines Manager Module: UI for specifying EG

Through this module, the evaluator can specify EGs as XML files. Thus, the EGs database is a set of XML files that the evaluator can update. When a new EG is added using this module, it is first associated with a list of graphical controls for which the EG is applied. Next, it is associated with numerical values and a comparison operator (Figure 4). The proposed operators are*: equal, different, superior, inferior, belongs to, different from, and between*. The evaluator-specified values are deduced from the EG being used to evaluate the ergonomic quality of the UI and according to the context of use. The evaluator specifies the guideline through the adopted modelling. In other words, first, the guideline is modelled through the guideline manager. Then, it is associated with the inspected aspect, to a comparison operator, and to recommended values. The interpretation of the guideline is established by the evaluator. The guideline manager can accommodate multiple sources of guidelines. The evaluator can select for the evaluation EG issued from different sources of EG**.** The selection of the EG is done through the item list and then through saving it to the list of adopted EG for the evaluation.

Another example is a recommendation from [Galitz, 07]: "*Keep the system default background image or color*". It is modeled as follows using the first module, Figure 5.

- Guideline ID: 29
- Associated Controls: All
- Aspect: “BackgroundColor”
- Operator: “Equal”
- Value 1: “Default”
- Value 2: None
- Error: “Background is not coherent”
- Recommendation: “Keep the system default background image or color”
- Name: “System Background Image Color”
- Description: “Keep the system default background image or color”
- Evaluation Method: “Heuristic Inspection”
- Ergonomic Criteria: “Consistency”
- Use Context: “Transportation”
- Guideline Base: “Gbase1”, and
- Reference: “Galitz2007”.

Figure 5: A first example of a guideline definition

Figure 6 depicts another EG extracted from [Galitz, 2007, p. 248]: *“User Control |Give users tools to allow them to succeed: Give the User Control:* The user should have control of the system wherever possible. This includes the ability to control the start and end of the system’s actions, the ability to cancel transactions in progress, and the ability to exit from the system and/or speak with a service representative”.

- Guideline ID: 63
- Associated Controls: Button
- Aspect: “Text”
- Operator: “In (Set)”
- Value 1: “Previous, Cancel, Call off, Invalidate, Undo, Repeal, Countermand, Override, Withdraw, Nullify”
- Value 2: None
- Error: “No control are provided to cancel the current task/action”
- Recommendation: “The user should have control of the system wherever possible. This includes the ability to control the start and end of the system’s actions, the ability to cancel transactions in progress, and the ability to exit from the system and/or speak with a service representative”
- Name: “Give the user control”
- Description: “The user should have control of the system wherever possible. This includes the ability to control the start and end of the system’s actions, the ability to cancel transactions in progress, and the ability to exit from the system and/or speak with a service representative
- Evaluation Method: “Heuristic Inspection”
- Ergonomic Criteria: “Explicit Control”
- Ergonomic Criteria: “User control”
- Use Context: “Transportation”
- Guideline Base: “Gbase1”, and
- Reference: “Galitz2007”.

Figure 6: A second example of a guideline definition

In figure 6, the used meta-model of EG is presented. The EG are modeled through a list of attributes and joined to associted error, recommendation, used evaluation method, etc.

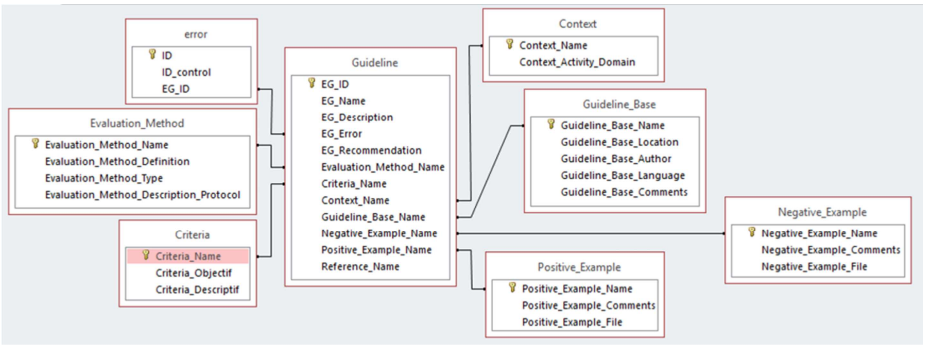

error
ID
ID_control
EG_ID
Evaluation_Method
Evaluation_Method_Name
Evaluation_Method_Definition
Evaluation_Method_Type
Evaluation_Method_Description_Protocol
Criteria
Criteria_Name
Criteria_Objectif
Criteria_Descriptif
Guideline
EG_ID
EG_Name
EG_Description
EG_Error
EG_Recommendation
Evaluation_Method_Name
Criteria_Name
Context_Name
Guideline_Base_Name
Negative_Example_Name
Positive_Example_Name
Reference_Name
Context
Context_Name
Context_Activity_Domain
Guideline_Base
Guideline_Base_Name
Guideline_Base_Location
Guideline_Base_Author
Guideline_Base_Language
Guideline_Base_Comments
Negative_Example
Negative_Example_Name
Negative_Example_Comments
Negative_Example_File
Positive_Example
Positive_Example_Name
Positive_Example_Comments
Positive_Example_File

Figure 7: The used meta-model of EG

3.2.2 Evaluation Data Capture Module

This module is used to capture different data for the evaluation process. As mentioned earlier, RITA is intended to consider (1) UI static presentations, (2) the quality of user-interface interaction, and (3) the user appreciation toward the evaluated UI. Subsequently, RITA requires three categories of data to be captured (i.e. UI presentation, user actions while interacting with the UI, and user satisfaction). These data are collected using three submodules, as shown in Figure 1.

3.2.2.1. Questionnaire Generator

The questionnaire generator is used first to generate a database of questions. The evaluator can add new questions as well as modify or delete existing ones. Each question is saved as an XML file. We opted to use closed-ended questions in questionnaire form to facilitate the processing and analyzing of user answers. We restrict to the user to choose only one answer per question. The evaluator selects a set of questions to compile the questionnaire for evaluating the UI.

Using an XSLT script [Williams, 09], the selected XML files (questions) are converted into a Web page containing the questionnaire, as shown in Figure 8. Whenever users answer a question and submit them, these answers are stored into a database. These answers are later processed by the third module and more specifically by its second submodule (i.e. Evaluation Engine Module | The questionnaire analyser). Note that the evaluator has the entire freedom to select the appropriated questions for the questionnaire. There is not any instruction about the selection of questionnaire.

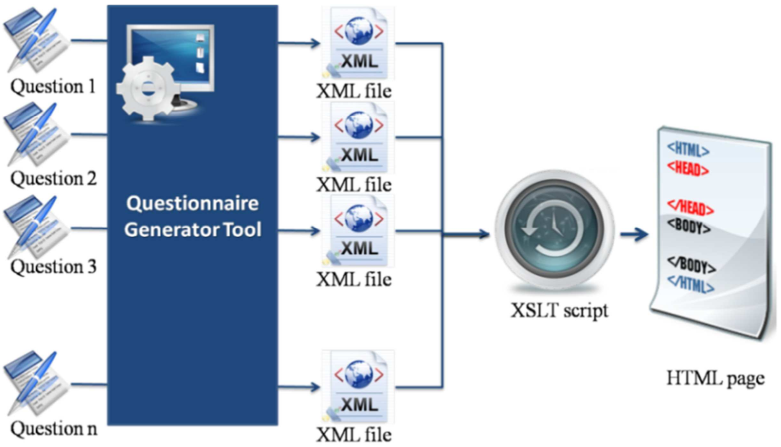
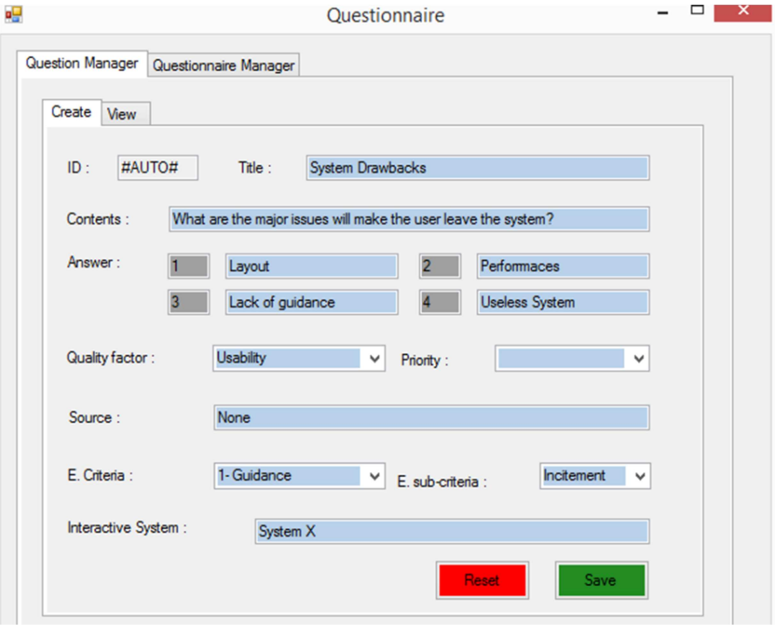

Question 1
Question 2
Question 3
Question n
Questionnaire Generator Tool
XML
XML file
XML
XML file
XML
XML file
XML
XML file
XSLT script
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
HTML page

Figure 8: The generation of a Web page questionnaire from XML files

Figure 9 illustrates this submodule in a screenshot demonstrating the addition of a new question. Note that a question is defined by:

- Name ("*System Drawbacks*")
- Content ("*What major issues will make this user leave our system?*")
- Proposed answer: Answer 1 ("*Layout*"), Answer 2 ("*Performances*"), Answer 3 ("*Lack of guidance*"), and Answer 4 ("*Useless System*")
- Quality factor to which it is associated ("*Usability*")
- Priority of the question ("*High*")
- Source of the Question ("*None*")
- Ergonomic Criteria (and Subcriteria) to which it is associated: ("*1- Guidance/Legibility*"), and
- Interactive System to which it is applied ("*System X*").



ID : #AUTO#
Title : System Drawbacks
Contents : What are the major issues will make the user leave the system?
Answer : 1 Layout 2 Performaces
3 Lack of guidance 4 Useless System
Quality factor : Usability
Priority :
Source : None
E. Criteria : 1- Guidance
E. sub-criteria : Incitement
Interactive System : System X

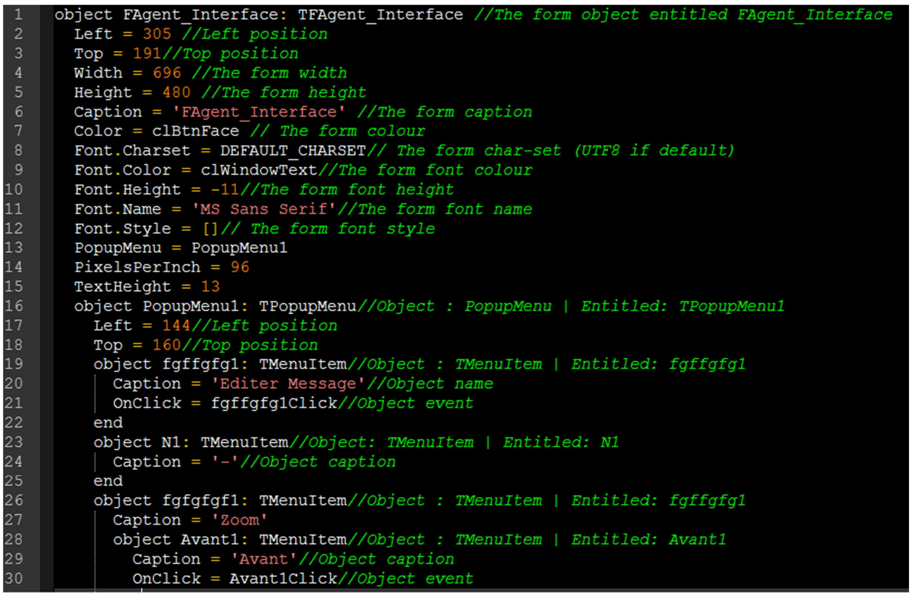
Figure 9: A screenshot of the questionnaire generator submodule

3.2.2.2. UI Data Parser

The data parser extracts the values of graphical control attributes from the UI source. The evaluator can specify using the provided submodule, the type of graphical controls interface (e.g. button, textbox, etc.) to be used to extract the attribute values (e.g. Then, the attributes for each graphical control type are selected. Next, the interface source code to inspect is specified. As an output, the extracted data is saved as an XML file, as shown in Figure 1. Figure 11 shows an example of interface control attribute extraction. The evaluator must specify the attributes to be extracted (e.g., Name, Backcolor, Location), the controls to be inspected (e.g., Button, Checkbox), the input path (i.e., the file containing the information) and the output path (i.e., the XML file where the extracted information are stored). On the right side, we see an example of the extracted information in a tree view. For the example cited above, we needed to extract the font name (Font.Name) for all controls (e.g., button, checklist, textbox) using the data parser, as shown in Figure 11. The data extraction is made automatically using the code and its general structure specified by the programming language editor. An example is provided to highlight the data extraction from a dfm[5] file, see Figure 10:

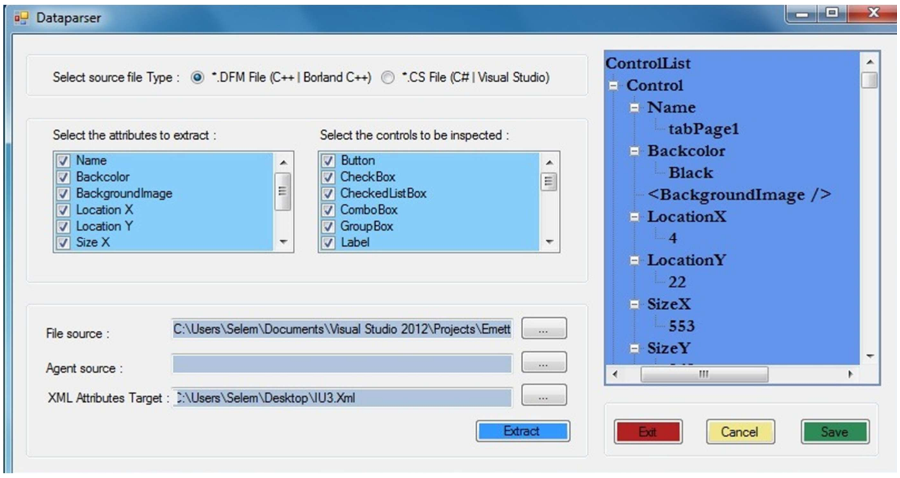
---

[5] The dfm files describe a form associated to c++ file. They are typically edited using C++ Builder IDE.



```
object FAgent_Interface: TFAgent_Interface //The form object entitled FAgent_Interface
  Left = 305 //Left position
  Top = 191//Top position
  Width = 696 //The form width
  Height = 480 //The form height
  Caption = 'FAgent_Interface' //The form caption
  Color = clBtnFace // The form colour
  Font.Charset = DEFAULT_CHARSET// The form char-set (UTF8 if default)
  Font.Color = clWindowText//The form font colour
  Font.Height = -11//The form font height
  Font.Name = 'MS Sans Serif'//The form font name
  Font.Style = []// The form font style
  PopupMenu = PopupMenu1
  PixelsPerInch = 96
  TextHeight = 13
  object PopupMenu1: TPopupMenu//Object : PopupMenu | Entitled: TPopupMenu1
    Left = 144//Left position
    Top = 160//Top position
    object fgffgfg1: TMenuItem//Object : TMenuItem | Entitled: fgffgfg1
      Caption = 'Editer Message'//Object name
      OnClick = fgffgfg1Click//Object event
    end
    object N1: TMenuItem//Object: TMenuItem | Entitled: N1
      Caption = '-'//Object caption
    end
    object fgfgfgf1: TMenuItem//Object : TMenuItem | Entitled: fgffgfg1
      Caption = 'Zoom'
      object Avant1: TMenuItem//Object : TMenuItem | Entitled: Avant1
        Caption = 'Avant'//Object caption
        OnClick = Avant1Click//Object event
```

Figure 10: An example of a dfm file containing a description of a c++ user interface.



Select source file Type : ● *.DFM File (C++ | Borland C++) ○ *.CS File (C# | Visual Studio)

Select the attributes to extract :
☑ Name
☑ Backcolor
☑ BackgroundImage
☑ Location X
☑ Location Y
☑ Size X

Select the controls to be inspected :
☑ Button
☑ CheckBox
☑ CheckedListBox
☑ ComboBox
☑ GroupBox
☑ Label

File source : C:\Users\Selem\Documents\Visual Studio 2012\Projects\Emett

Agent source :

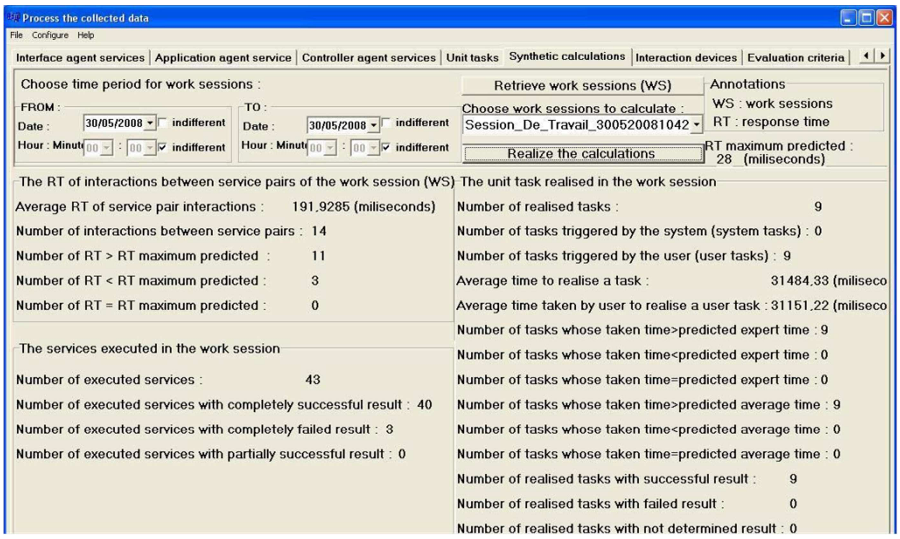
XML Attributes Target : :\Users\Selem\Desktop\IU3.Xml

ControlList
Control
Name
tabPage1
Backcolor
Black
<BackgroundImage />
LocationX
4
LocationY
22
SizeX
553
SizeY

Figure 11: A screenshot of the Data parser submodule

3.2.2.3. EISEval Electronic Informer

This electronic informer was designed during Tran's PhD thesis [Tran, 08] (Figure 12).



Choose time period for work sessions :

FROM :
Date : 30/05/2008 ☐ indifferent
Hour : Minute 00 : 00 ☑ indifferent

TO :
Date : 30/05/2008 ☐ indifferent
Hour : Minute 00 : 00 ☑ indifferent

Choose work sessions to calculate : Session_De_Travail_300520081042

Annotations
WS : work sessions
RT : response time

RT maximum predicted : 28 (miliseconds)

The RT of interactions between service pairs of the work session (WS)
Average RT of service pair interactions : 191,9285 (miliseconds)
Number of interactions between service pairs : 14
Number of RT > RT maximum predicted : 11
Number of RT < RT maximum predicted : 3
Number of RT = RT maximum predicted : 0

The services executed in the work session
Number of executed services : 43
Number of executed services with completely successful result : 40
Number of executed services with completely failed result : 3
Number of executed services with partially successful result : 0

The unit task realised in the work session
Number of realised tasks : 9
Number of tasks triggered by the system (system tasks) : 0
Number of tasks triggered by the user (user tasks) : 9
Average time to realise a task : 31484,33 (miliseco
Average time taken by user to realise a user task : 31151,22 (miliseco
Number of tasks whose taken time>predicted expert time : 9
Number of tasks whose taken time<predicted expert time : 0
Number of tasks whose taken time=predicted expert time : 0
Number of tasks whose taken time>predicted average time : 9
Number of tasks whose taken time<predicted average time : 0
Number of tasks whose taken time=predicted average time : 0
Number of realised tasks with successful result : 9
Number of realised tasks with failed result : 0
Number of realised tasks with not determined result : 0

Figure 12: A screenshot of the EISEval [Tran, 08]

Its objective is essentially to detect and capture low-level events (e.g., mouse cursor movement, user mouse clicks and keyboard inputs). In addition, it helps the evaluator associate these events with specific user tasks (i.e. associate elementary actions to tasks). This classification breaks different interface events into two main categories: (1) EVIU (EVent Interface of User) concerns tasks in which the interface is used (for example, to open a window or read an error message); (2) EVDI (EVent of Devices of Interaction) addresses the events generated by interaction devices (such as right mouse clicking or pressing a key on the keyboard). The interaction is modeled through EVIU. These EVIUs are generated through the detected EVDI. Note that an EVIU is composed of one or more EVDIs. However, as shown in Figure 12, EISEval provides the evaluator with some statistics such as the number of tasks executed by the user, the number of system tasks, and the success rate of executed tasks. Interested readers can consult [Tran, 08].

Note that we opt for simplistic vision of the user events to model the events only on two levels. Hilbert and Redmiles proposed a more detailed model of events (2000), based on six user action levels. The first includes the highest event level. The second level concerns the goal of the tasks to be achieved by the user. The third level involves all the activities related to the same graphical UI object (i.e. GUI control). Then, the fourth level includes UI events (e.g. using the keyboard keys). The fifth level involves the actions performed using input devices. The last level provides information about the physical actions of the user. This level concerns low-level events. The EVDI corresponds to the sixth level (physical actions) and the EVIU encompasses the five first levels.

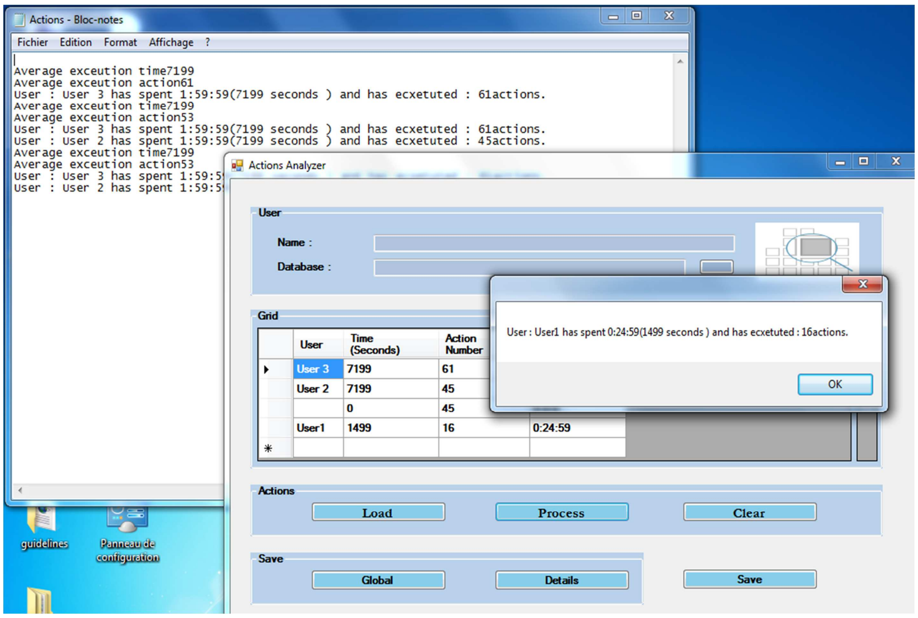
3.2.3 Evaluation Engine Module

This module is the "*heart*" of the Framework seen in Figure 1. It is intended to make use of its four submodules to analyze the data generated by the electronic informer, the questionnaire, and the data parser.

3.2.3.1. Interaction Data Analyzer

This submodule is based on numerical data about the interaction between the user and the interface. It provides some data, such as total task execution duration, the number of executed tasks per user and executed event sequencing, as shown in Figure 13. These calculations are established via the tasks decomposition into elementary events that are gathered by the electronic informer. The Interaction Data Analyzer enables an evaluator to establish comparisons between the sequence of executed user actions and the average sequence through Petri Nets (PN) following the principle described in [Charfi, 11]. The evaluation framework supports the frequency, the sequence, and the links between the tasks. Nevertheless, the task importance is not supported since we consider short actions sequence for the evaluation and then all the tasks are related to the same importance level.

The executed actions are modeled as PN in order to facilitate their processing. This modelling is established through a tool entitled *Renew* using the executed elementary actions and the tasks decomposition by elementary actions.



```
Average exceution time7199
Average exceution action61
User : User 3 has spent 1:59:59(7199 seconds ) and has ecxetuted : 61actions.
Average exceution time7199
Average exceution action53
User : User 3 has spent 1:59:59(7199 seconds ) and has ecxetuted : 61actions.
User : User 2 has spent 1:59:59(7199 seconds ) and has ecxetuted : 45actions.
Average exceution time7199
Average exceution action53
User : User 3 has spent 1:59:5
User : User 2 has spent 1:59:5
```

	User	Time (Seconds)	Action Number	
▸	User 3	7199	61	
	User 2	7199	45	
		0	45	
	User1	1499	16	0:24:59
*				

User : User1 has spent 0:24:59(1499 seconds) and has ecxetuted : 16actions.

Figure 13: Overview of the "Automated user events analyzer" interface submodule.

3.2.3.2. The questionnaire analyzer

This submodule analyzes users' answers to questionnaires and provides numerical data about these answers to give the evaluator a better overview of the answers, as shown in Figure 14.

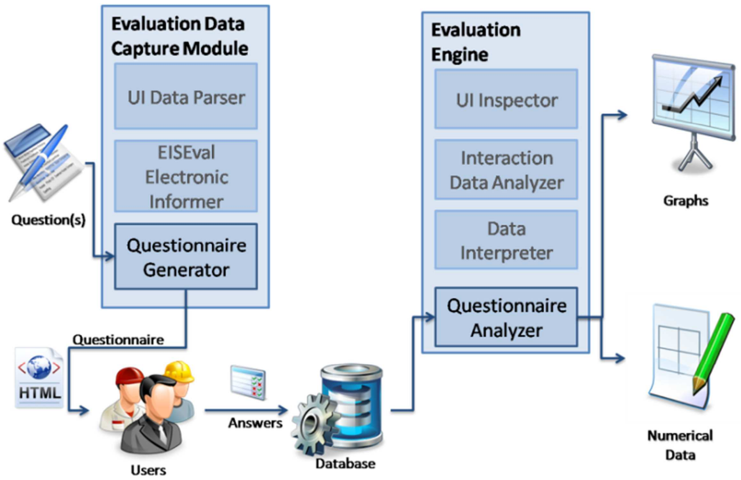
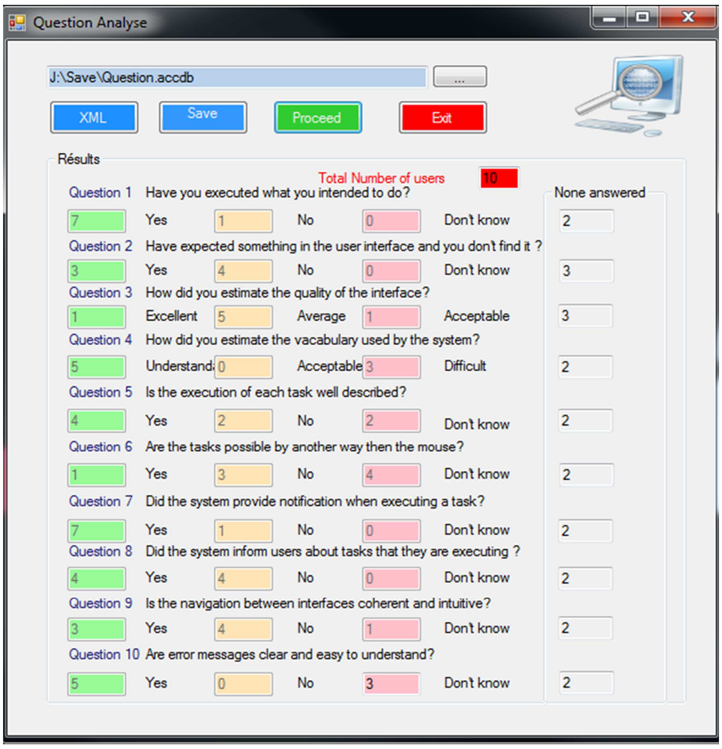

Evaluation Data Capture Module
UI Data Parser
EISEval Electronic Informer
Questionnaire Generator
Question(s)
Questionnaire
HTML
Users
Answers
Database
Evaluation Engine
UI Inspector
Interaction Data Analyzer
Data Interpreter
Questionnaire Analyzer
Graphs
Numerical Data

Figure 14: UI evaluation questionnaire Principle

For each question, the analyzer calculates the number of users that answer to it and the breakdown of the various answer alternatives (Figure 15).

3.2.3.3. The UI Inspector

The UI Inspector is used to detect the UI design problems based on the EG specified by the evaluator in the first evaluation process module, as shown in Figure 16. The inputs for this sub-module come from the XML file directory (modeling the EGs in question) and an XML file with the values of the UI graphical controls attributes (e.g. controls‘ dimension, font and color.), as shown in Figure 1. The interface inspector then executes methods according to comparison operators under consideration in order to verify the consistency between the recommended value(s) and the UI attributes. The inspector provides two messages as outputs: the errors and the recommendations pertaining to the non-respected EG, as shown in Figure 17. As an example, we cite information readability problems.


Question Analyse
...
XML
Save
Proceed
Exit

J:\Save\Question.accdb

Résults

Total Number of users 10

None answered

Question 1 Have you executed what you intended to do?

7 Yes 1 No 0 Don't know 2

Question 2 Have expected something in the user interface and you don't find it ?

3 Yes 4 No 0 Don't know 3

Question 3 How did you estimate the quality of the interface?

1 Excellent 5 Average 1 Acceptable 3

Question 4 How did you estimate the vacabulary used by the system?

5 Understand: 0 Acceptable 3 Difficult 2

Question 5 Is the execution of each task well described?

4 Yes 2 No 2 Don't know 2

Question 6 Are the tasks possible by another way then the mouse?

1 Yes 3 No 4 Don't know 2

Question 7 Did the system provide notification when executing a task?

7 Yes 1 No 0 Don't know 2

Question 8 Did the system inform users about tasks that they are executing ?

4 Yes 4 No 0 Don't know 2

Question 9 Is the navigation between interfaces coherent and intuitive?

3 Yes 4 No 1 Don't know 2

Question 10 Are error messages clear and easy to understand?

5 Yes 0 No 3 Don't know 2

Figure 15: A screenshot of a sample numerical synthesis of user questionnaire answer data

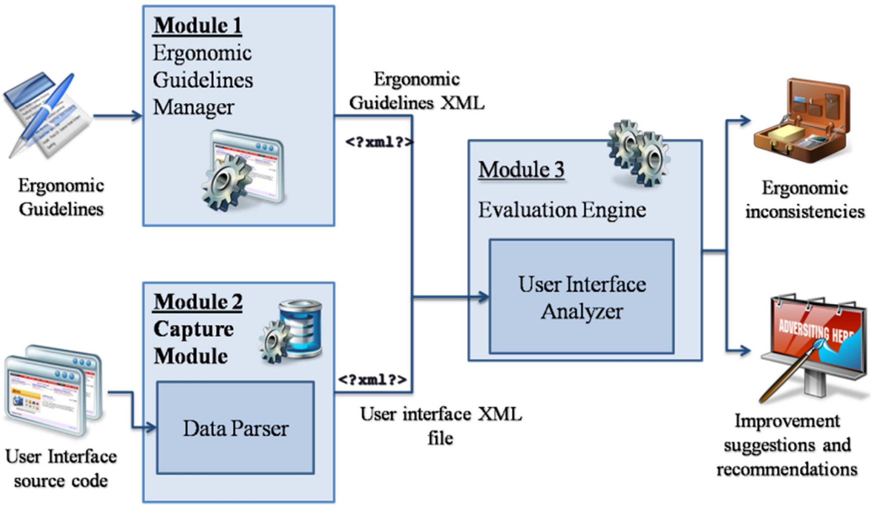
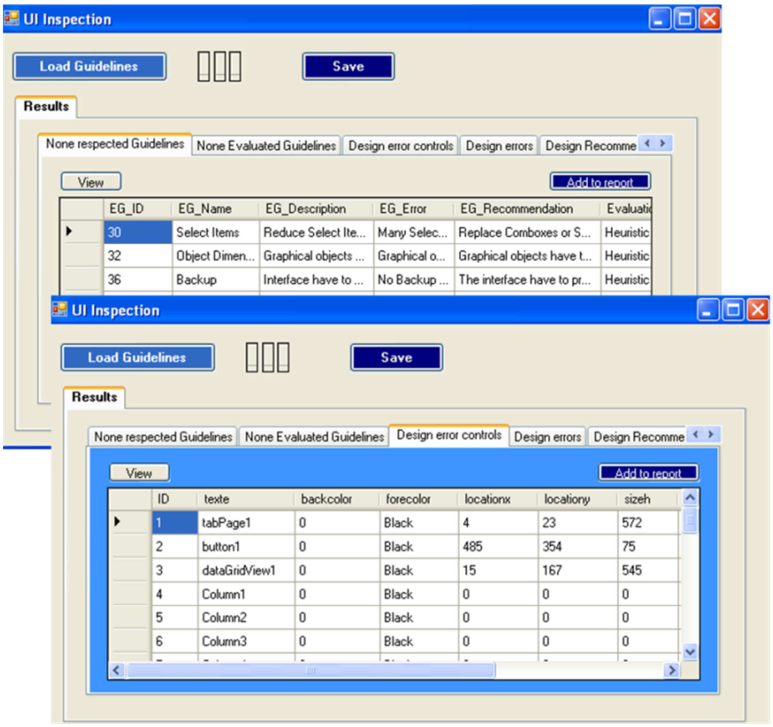

Module 1
Ergonomic Guidelines Manager
Ergonomic Guidelines
Ergonomic Guidelines XML
<?xml?>
Module 3
Evaluation Engine
User Interface Analyzer
Ergonomic inconsistencies
Module 2
Capture Module
Data Parser
User Interface source code
<?xml?>
User interface XML file
ADVERSITING HERE
Improvement suggestions and recommendations

Figure 16: The adopted ergonomic quality inspection principle

3.2.3.4. The data Interpreter

Once the evaluator has received the evaluation data of the other three submodules, that evaluator must synthesize the data to obtain a final evaluation result. Data synthesis involves grouping the data according to the system supported user tasks. Three coefficients are associated with each task. The coefficient is inspired maily from [Joshi, 10]. These coefficients are numerical values between 0 and 1:

- The first coefficient C1 (task completion rate) corresponds to the ratio of users that have performed the task successfully (e.g., 0.5, or half of the users were able to perform the task successfully)
- The second coefficient C2 (positive feedback rate) corresponds to the ratio of positive responses from users on the questionnaire related to this task, and
- The third coefficient C3 (control invocation rate) involves the proportion of the UI graphical controls used to perform this task in accordance with the EG specified using the first module.


UI Inspection
Load Guidelines
Save
Results
None respected Guidelines
None Evaluated Guidelines
Design error controls
Design errors
Design Recomme
View
Add to report

EG_ID	EG_Name	EG_Description	EG_Error	EG_Recommendation	Evaluati
30	Select Items	Reduce Select Ite...	Many Selec...	Replace Comboxes or S...	Heuristic
32	Object Dimen...	Graphical objects ...	Graphical o...	Graphical objects have t...	Heuristic
36	Backup	Interface have to ...	No Backup ...	The interface have to pr...	Heuristic

UI Inspection
Load Guidelines
Save
Results
None respected Guidelines
None Evaluated Guidelines
Design error controls
Design errors
Design Recomme
View
Add to report

ID	texte	backcolor	forecolor	locationx	locationy	sizeh
1	tabPage1	0	Black	4	23	572
2	button1	0	Black	485	354	75
3	dataGridView1	0	Black	15	167	545
4	Column1	0	Black	0	0	0
5	Column2	0	Black	0	0	0
6	Column3	0	Black	0	0	0

Figure 17: A screenshot of sample results of an ergonomic inspection

In this evaluation framework, we adopted the design error classification proposed by Nielsen (1993): cosmetic (Between 1.0 and 0.9), minor (Between 0.9 and 0.8), average (Between 0.8 and 0.7), major (Between 0.7 and 0.5), critical (Between 0.5 and 0.0).

In the adopted evaluation process, the electronic informer coefficient is considered as having the heaviest weight. Interaction data capture and analysis is the most reliable evaluation technique of the three used techniques. It is based on quantitative numerical data. Furthermore, it does not require any subjective judgments from the evaluator. The results of the ergonomic quality inspection and the questionnaire complement those of the electronic informer. The Table 1 illustrates an example of coefficients.

Table 1: Sample coefficients.

Task	C1	C2	C3	Result
T1	0.90	0.50	0.20	- Good quality - Some cosmetic errors to correct
T2	1.00	0.80	0.60	- Excellent UI quality - Some cosmetic errors to correct
T3	0.50	0.90	0.36	- Many utility and usability problems detected

Note that the correlation between the different types of data is established through grouping the different data task by task. A task can be executed though many page screens. The correspondence between the controls and the tasks is established through the list of the required controls for each task. The association between the tasks and the question is established manually by the evaluator.

3.2.4 Evaluation Report Generator

This module builds the evaluation report in an understandable and readable format, as shown in Figure 1. The report mainly includes the detected design problems, the recommendations for improving the UI and an overview of the evaluation process. The provided report can be exported in three different formats: PDF, HTML, and TXT, as shown in Figure 18.

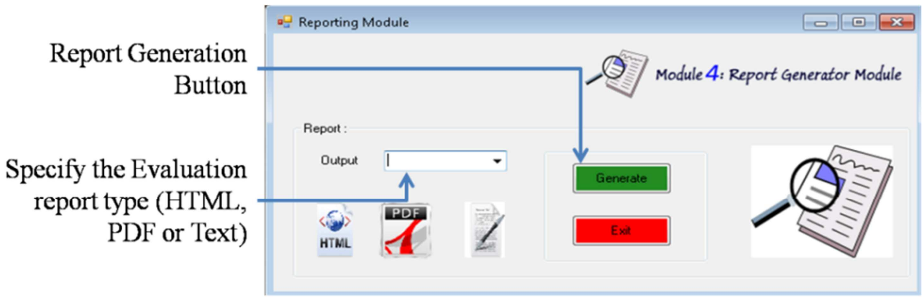

Report Generation Button
Specify the Evaluation report type (HTML, PDF or Text)
Reporting Module
Module 4: Report Generator Module
Report :
Output
Generate
Exit
HTML
PDF

Figure 18 : A screen shot of the Evaluation Report Building Module interface

4 Proposed evaluation process

The adopted UI evaluation process using the RITA framework requires the participation of three stakeholders: (1) the evaluator (to manage and assist in the evaluation process), (2) a Human Factors expert (to select and define the EG), (3) users (to participate in the interaction sessions), and (4) the designer (to inject code into the evaluated system source to enable the electronic informer to gather data about the interaction).

4.1 The pre-evaluation phase

During this phase, as shown in Figure 19, the evaluator proceeds with: (1) specifying evaluation objectives, (2) identifying the evaluated system use context, (3) specifying a list of EG to follow when inspecting UIs conformity, (4) specifying a list of the questions to ask users, (5) generating a questionnaire, (6) designing a scenario for the interaction session, (7) preparing the material required for the evaluation, and (8) conducting the interaction session with users.

4.2 The evaluation phase

This phase involves extracting the data needed to identify UI usability and utility problems. The collected data are: graphical UI control attributes, user responses, and interaction data.

Next, the collected data are processed and synthesized to help the evaluator evaluate UIs, Figure 20.

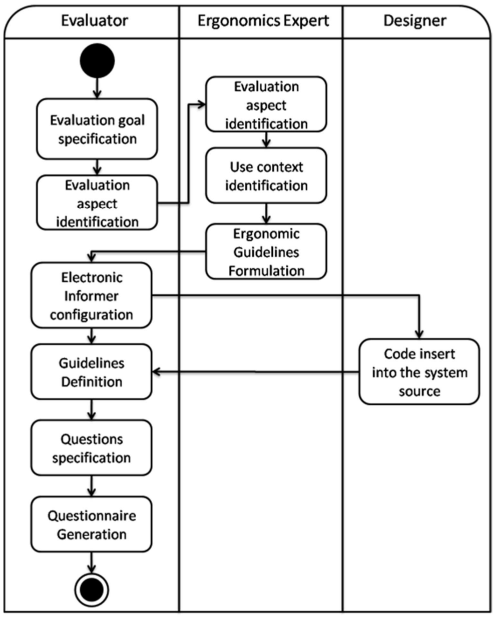

Evaluator
Ergonomics Expert
Designer
Evaluation goal specification
Evaluation aspect identification
Evaluation aspect identification
Use context identification
Ergonomic Guidelines Formulation
Electronic Informer configuration
Code insert into the system source
Guidelines Definition
Questions specification
Questionnaire Generation

Figure 19: Activity diagram of the pre-evaluation process using the RITA Framework

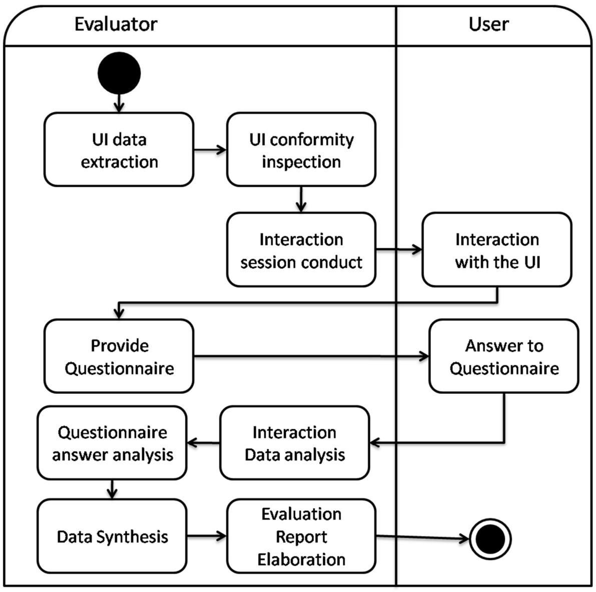

Evaluator
User
UI data extraction
UI conformity inspection
Interaction session conduct
Interaction with the UI
Provide Questionnaire
Answer to Questionnaire
Questionnaire answer analysis
Interaction Data analysis
Data Synthesis
Evaluation Report Elaboration

Figure 20: Activity diagram of the evaluation process using the RITA Framework

4.3 Post-evaluation phase

First, an evaluation report is generated, as shown in Figure 1. The evaluation report includes the evaluation result: (1) a list of the design problems detected in the evaluated UI, (2) a list of suggestions and recommendations to improve the evaluated UI, and (3) a report about the evaluation process progress. It covers the problems encountered during this process.

Once the report is generated, the designer implements the various suggestions and recommendations to improve the evaluated UI. Depending on the result of the evaluation, and the project constrains (time and cost), the evaluator can proceed to iterative evaluation.

4.4 Conclusion about the RITA framework

In this section, we present the adopted evaluation process supported by the RITA Framework. Like most existing evaluation tools, it includes three main phases (i.e., the electronic informer, the questionnaire, and ergonomic quality inspection.). Note that the evaluator and the UI designer manually establish the first phase. The second and the third phases are supported by the evaluation framework. Note that, with the exception of the different data evaluation syntheses, these two phases are established automatically.

In order to validate the framework, we proceed to the evaluation of a transportation network system. This case study is described in the following section.

5 Case study: application of RITA to the evaluation of an interactive transportation network supervision system

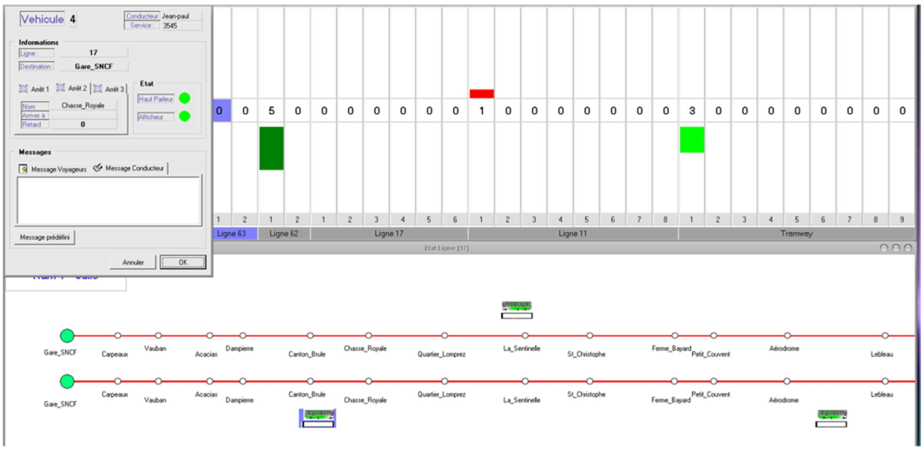
This section presents a case study of the proposed framework. This study focuses on evaluating a system dedicated to supervising the urban transportation network supervision system in Valenciennes (France). This system is called the IAS (Information Assistant System); in its current version, it is available on PCs, see Figure 21. This system is described in the next section.

Figure 21: A picture of a user using the IAS

5.1 The IAS: Presentation

This system was designed as part of the SART project of the Regional (Nord-Pas-de-Calais) Grouping for Transport Research (partners: LAMIH, LAGIS, IFSTAR, Transvilles). It aims to help travelers by improving the quality of urban transport network connections by keeping these travelers informed of the traffic situation in real time. It was suggested to assist human regulators in a control room manage and supervise urban transportation networks. The purpose of the system is to ensure that regulators perform their tasks optimally during normal and abnormal situations [Ezzedine, 08]. In other words, its purpose is to minimize passenger-waiting time and ensure continuity of travel in multimodal networks.

The IAS is responsible for presenting different types information about the transportation network to passengers and regulators, as shown in Figure 22. Its main goal is to inform supervisors about the positioning of different vehicles in the transportation network. In addition, the IAS also enables supervisors to communicate with different vehicles drivers. The IAS has 25 page screens.


Vehicule 4
Conducteur Jean-paul
Service : 3545
Informations
Ligne : 17
Destination : Gare_SNCF
Nom Chasse_Royale
Arriver à
Retard 0

Arrêt 1
Arrêt 2
Arrêt 3
Etat
Haut Parleur
Afficheur
Messages
Message Voyageurs
Message Conducteur
Message prédéfini
Annuler
OK
0 0 5 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0
1 2 1 2 1 2 3 4 5 6 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 9
Ligne 63
Ligne 62
Ligne 17
Ligne 11
Tramway
Etat Ligne [17]
Gare_SNCF
Carpeaux
Vauban
Acacias
Dampierre
Canton_Brule
Chasse_Royale
Quartier_Lomprez
La_Sentinelle
St_Christophe
Ferme_Bayard
Petit_Couvent
Aérodrome
Lebleau

Figure 22: A screenshot of the IAS UI

5.2 The adopted evaluation process

As mentioned above, RITA is a framework that incorporates three different evaluation techniques: the electronic informer, the questionnaire, and ergonomic quality inspection. In this section, we briefly describe the evaluation process used. Some subprocesses for comparison with the presented process, such as determining the evaluation objectives, are missing in this section because our goal is not to evaluate the system, but rather to perform a case study to test the proposed evaluation Framework.

5.2.1 Inspection of UI ergonomic quality

To evaluate the IAS UIs, we selected ten EGs: as explained previously, the main goal of this experimental study is not to evaluate the IAS system it-self; rather, it is to test the proposed framework; as a result, we only choose a set of ten ergonomic guidelines for the feasibility study. Note that the evaluator can add new guidelines. There is not any constraint about the guidelines number. They pertain to information legibility (e.g., font color, control size and number). The evaluation is established page screen by page screen. This evaluation is achieved by automatically comparing the UI control attribute values to the values specified in the EGs. For example, one EG is, "*Given the unpredictability of color displays, users, and viewing situations, the choice can get very complicated. Color is often best used to highlight key information. As a general rule, use no more than three colors for primary information*" [Watzman, 02, p. 347]. In other words, RITA ensures that the designer did not use more than three primary colors (i.e., red, green, and blue) for the interface design. This EG is modeled as two EGs. The first one is devoted to the number of used colors based on the gathered data from parsing the UI source. The graphical UI ergonomic quality inspector counts the number of used colors per page screen. It reports the compliance of the obtained number with the recommended number (three). The second EG pertains to the chosen colors. The proposed tool determines whether the used colors belong to the following set or not: {Red, Green, Blue}. The evaluation Framework proceeds to extract the used colors in the UI and to check if each one belongs to the specified set. It reports the conformity of the chosen colors with the specified set.

5.2.2 Capturing the user interaction with the interface

This evaluation involves establishing interaction sessions with users to capture data on the interaction between the user and the interface (in our case, the IAS). This data is processed to assist the evaluator in UI evaluation.

5.2.2.1. Choice of users for the interaction session

There are many recommendations for choosing the number of users for interaction sessions [Hwang, 10; Whiting, 08]. In this feasibility study, we decided to involve ten users in the interaction sessions. These users were master's degree students aged 20 to 30 years. Their areas of expertise included computer science and the science and technology for physical and sports activities.

5.2.2.2. Experimentation scenario

During the interaction session, the user acted as a network supervisor (also known as a regulator). The user was confronted with a simulated network characterized by several disturbances (i.e., 48 disturbances are incorporated into the network during the simulation). Whenever there was a disruption in the network (e.g. a late or early passage of a bus or a tram with respect to the schedule), the regulator had to notify the driver and provide instructions to handle this disturbance.

The simulated network was essentially characterized by: (1) a high frequency of network disturbances, (2) an overlap of several instant disturbances, (3) all network lines being affected by disturbances, and (4) a frequent change in the early/late values for different vehicles.

In this evaluation, we conducted work sessions corresponding to a disturbed situation to prompt the user to perform as many tasks as possible.

5.2.2.3. Experimental software devices

The experimental devices were based on three components, as shown in Figure 23: (1) the interactive system to be evaluated (i.e. the IAS); (2) the EISEval Electronic Informer, which captures the actions performed by the user on the interface during the interaction session and stores them in a database [Tran, 08], and (3) the disturbance simulator, which sends frames to the IAS to incorporate disturbances into the transportation network [Charfi, 13].

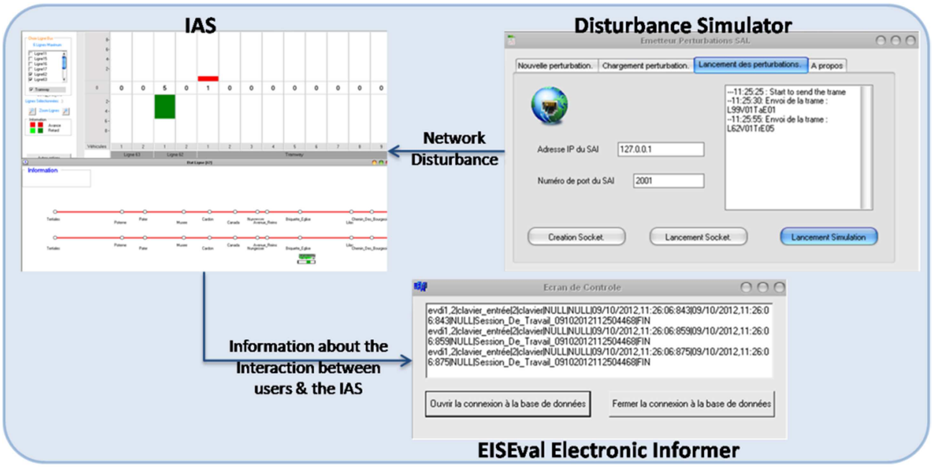
The communication between these three components is established via a TCP/IP network by sending messages through sockets. They operate as follows:

1. The disturbance simulator sends information frames[6] to the IAS in order to create disturbances in the simulated network using a well-defined scenario.
2. The IAS shows the user (regulator) the disturbances on the network.

---

[6] The frames' format is LLLVVVDDEEE where: (1) LLL is the network Line, (2) VVV is the vehicle, (3) DD is the disturbance type (*Ta* for early and *Tr* for late), and (4) EEE is the disturbance duration.

3. The user tries to handle the disturbances that occur on the network (Figure 21).
4. The EISEval informer captures and stores the data related to the user-executed actions.


IAS
Disturbance Simulator
Emetteur Perturbations SAL
Nouvelle perturbation.
Chargement perturbation.
Lancement des perturbations.
A propos
Adresse IP du SAI
127.0.0.1
Numéro de port du SAI
2001
Creation Socket.
Lancement Socket.
Lancement Simulation
Network Disturbance
Information about the Interaction between users & the IAS
Ecran de Controle
Ouvrir la connexion à la base de données
Fermer la connexion à la base de données
EISEval Electronic Informer

```
---11:25:25 : Start to send the trame
--11:25:30: Envoi de la trame :
L99V01TaE01
--11:25:55: Envoi de la trame :
L62V01TrE05
```

```
evdi1,2|clavier_entrée|2|clavier|NULL|NULL|09/10/2012,11:26:06:843|09/10/2012,11:26:0
6:843|NULL|Session_De_Travail_09102012112504468|FIN
evdi1,2|clavier_entrée|2|clavier|NULL|NULL|09/10/2012,11:26:06:859|09/10/2012,11:26:0
6:859|NULL|Session_De_Travail_09102012112504468|FIN
evdi1,2|clavier_entrée|2|clavier|NULL|NULL|09/10/2012,11:26:06:875|09/10/2012,11:26:0
6:875|NULL|Session_De_Travail_09102012112504468|FIN
```

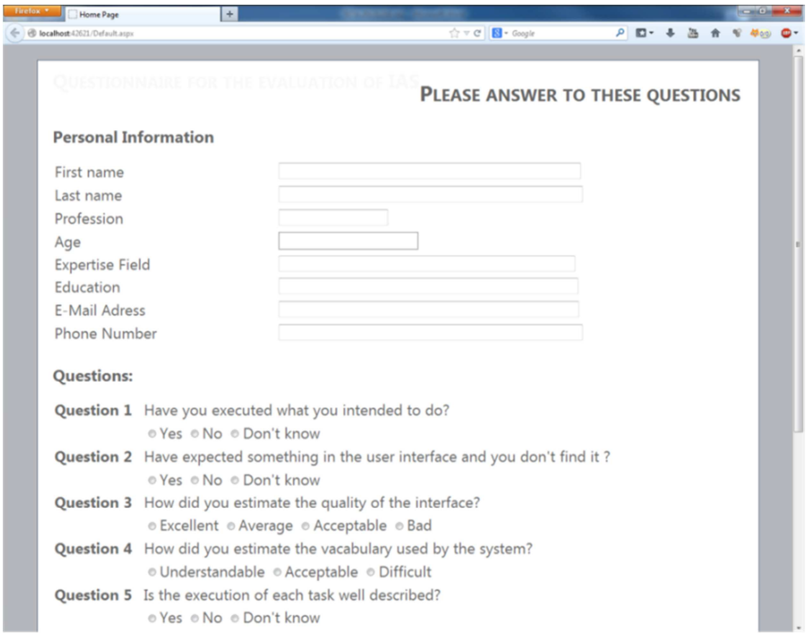
Figure 23: Experimental devices for the interaction session

5.2.3 Questionnaire

After interacting with the evaluated system, a questionnaire was proposed to the users. It included two parts. The first involved questioning the users about user information, including name, age, gender and level of expertise. This part was used to classify answers according to their profiles. The second part included 25 questions[7] specific to using the second module. The questionnaire was proposed to the user via an HTML interface. The collected answers were stored in a database, as shown in Figure 24.

---

[7] As mentioned at the beginning of this section, our aim was to present a feasibility study of the proposed framework and not just to evaluate the IAS system itself. We suggested only 25 questions for this feasibility study. These questions were proposed to examine aspects that cannot be evaluated through guideline inspection and the electronic informer, and to provide complementary results. Note that the evaluator can add new questions. There is not any constraint about the number of questions.



QUESTIONNAIRE FOR THE EVALUATION OF IAS

PLEASE ANSWER TO THESE QUESTIONS

Personal Information

First name
Last name
Profession
Age
Expertise Field
Education
E-Mail Adress
Phone Number

Questions:

Question 1 Have you executed what you intended to do?
○ Yes ○ No ○ Don't know

Question 2 Have expected something in the user interface and you don't find it ?
○ Yes ○ No ○ Don't know

Question 3 How did you estimate the quality of the interface?
○ Excellent ○ Average ○ Acceptable ○ Bad

Question 4 How did you estimate the vacabulary used by the system?
○ Understandable ○ Acceptable ○ Difficult

Question 5 Is the execution of each task well described?
○ Yes ○ No ○ Don't know

Figure 24: A screenshot of the questionnaire to be fulfilled by each user

5.3 Summary of the evaluation results and discussion

After analyzing the different interaction sessions summarized into Table 2 that contains the execution time, the number of executed actions, and the time and task number space (regarding the average of time execution and the executed tasks) per user. We concluded that users encountered some problems:

- Users did not execute the same number of tasks, although they were all confronted with the same evaluation scenario (i.e., 48 disturbances).
- Every time a disturbance value changed (i.e. the value of the advance or the delay change for instance from three to two minutes), users generally did not pay attention to it.
- Many users confused passenger messages with driver messages, and then subsequently sent messages to passengers instead of drivers.
- The default overview of the network did not illustrate the network lines in their entirety. Users needed to include more vehicle lines in order for more lines to be visualized. Not all users did this, and therefore some users failed to handle the disturbances related to these “hidden” lines.
- There were no mechanisms provided to users to allow them to cancel operations and prevent errors.
- The system did not send messages to let network supervisors know whether or not their messages had been correctly sent. Therefore, the network supervisors tended to re-send to same message thinking they had not been sent.

- Some users confused early vehicles with late vehicles.
- Some users did not handle problems with early vehicles because they thought this situation was ok. In fact, green was used to highlight early vehicles and red was used to highlight late vehicles. However, users thought that only red signified disturbances.

The evaluation performed through the ergonomic quality inspection showed that the information provided to the users was clearly presented. However, during the feasibility study, we found that some users were confused between early and late vehicles. This is due to the color choices of green or red. Users were also confused about the message recipient: driver or passenger. After grouping the different coefficients per task, the evaluated UI problems were detected. These problems are listed according to the criteria found in ISO standard 9241-111 [ISO, 99].

The problems pertained to seven areas: (1) Clarity: information should be conveyed quickly and accurately; (2) Discriminability: information should be able to be accurately distinguished; (3) Conciseness: only the information necessary to complete the task should be provided; (4) Consistency: the same information should be presented in the same way throughout the application; (5) Detectability: the user's attention should be directed to the required information; (6) Legibility: information should be easy to read; (7) Comprehensibility: the meaning should be clearly understandable.

Table 2: Evaluation result issued from the interaction data

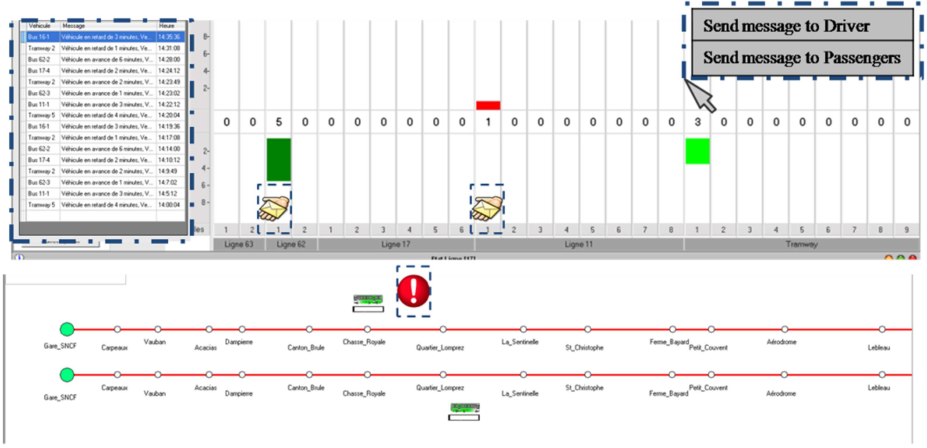
User	**Execution Time**	**Time space**	**Realized actions**	**Task space**
1	17:32	+00.00	32	-15.38
2	14:07	-19.48	36	-07.69
3	17:21	-01.04	27	-30.76
4	17:04	+03.04	48	+23.07
5	19:24	+01.65	42	+07.69
6	17:31	+00.00	43	+07.16
7	18:49	+07.31	22	+43.58
8	17:24	-00.76	42	+07.63
9	18:46	+10.45	44	+12.82
10	18:06	+03.25	57	+46.15
Average	17:36	+00.44	39	+09.43

UI conciseness and homogeneity were respected. The information presented to the user was what was necessary to supervise the transportation network. The consistency factor was validated, except for the green and red, and the detectability factor was also validated. However, we detected some legibility factor design problems. First, the designer used only one color to present the different types of information to users (namely, black). In addition, some interfaces were fairly dense. Moreover, users encountered legibility problems using these interfaces.

To handle the detected design problems, we proposed adding an interface to view and manage the sent message history. We also proposed adding notifications on handled disturbances to the interface so that users would not be confused about disturbances.

The evaluated UI did not support any mechanisms to prevent the user from making mistakes (e.g., from choosing the wrong vehicle to receive a message). The user must confirm some actions before executing them. Furthermore, the evaluated UI does not support any mechanisms for enabling users to undo the most recent actions or return to a previous state, meaning that users cannot correct erroneous actions. Moreover, the supervisor does not receive any confirmation that messages have been sent.

Subsequently, we propose a new IAS prototype to address the detected utility and usability problems, as shown in Figure 25. In this prototype, we added a data grid view table (at the top left of the UI) to display the history of the messages sent by the supervisor. In addition, we added an icon to the network disturbances handled by the supervisor. We also proposed providing an opportunity for the regulator to send a message to a vehicle driver or passengers by directly clicking on the bar associated with the corresponding vehicle in the "Traffic_State" interface. Moreover, on the "Line_State" interface, we added warning markers to indicate that a vehicle was early or late and there was no action performed to address this. These changes are surrounded by dotted blue rectangles in Figure 25.



Vehicule	Message	Heure
Bus 16-1	Véhicule en retard de 3 minutes, Ve...	14:35:36
Tramway 2	Véhicule en retard de 1 minutes, Ve...	14:31:08
Bus 62-2	Véhicule en avance de 6 minutes, V...	14:28:00
Bus 17-4	Véhicule en retard de 2 minutes, Ve...	14:24:12
Tramway 2	Véhicule en avance de 2 minutes, V...	14:23:49
Bus 62-3	Véhicule en avance de 1 minutes, V...	14:23:02
Bus 11-1	Véhicule en avance de 3 minutes, V...	14:22:12
Tramway 5	Véhicule en retard de 4 minutes, Ve...	14:20:04
Bus 16-1	Véhicule en retard de 3 minutes, Ve...	14:19:36
Tramway 2	Véhicule en retard de 1 minutes, Ve...	14:17:08
Bus 62-2	Véhicule en avance de 6 minutes, V...	14:14:00
Bus 17-4	Véhicule en retard de 2 minutes, Ve...	14:10:12
Tramway 2	Véhicule en avance de 2 minutes, V...	14:9:49
Bus 62-3	Véhicule en avance de 1 minutes, V...	14:7:02
Bus 11-1	Véhicule en avance de 3 minutes, V...	14:5:12
Tramway 5	Véhicule en retard de 4 minutes, Ve...	14:00:04

Figure 25: New IAS prototype proposal

The provided evaluation focused on three aspects: system use, information presentation and user appreciation. However, it is also interesting to note that in this initial feasibility study, the evaluation results do not converge as we had expected. For instance, while inspecting ergonomic quality, we detected a set of ergonomic inconsistencies in some interfaces, while the interaction session did not detect any use problems with these interfaces.

During the evaluation, evaluators were faced with a series of inconsistent data (e.g., task execution times, UI aspects that do not comply with EGs, user answers to the questionnaire). We granted priority to data generated by the electronic informer over the two other evaluation techniques. In fact, these two techniques were used as techniques that are complementary to the electronic informer. If the user did not have any difficulties performing a task and the inspection identified some design errors (according to a set of EG), the evaluator could conclude that the interface did not present any design problems. Thus, the detected design problems were ignored.

Different IAS versions have been evaluated to validate the contribution of the research conducted during the PhD theses of Trabelsi [Trabelsi, 06] and Tran [Tran, 08]. Both of the contributions are electronic informers. They only provide users with certain statistics. They do not provide the evaluator with an evaluation report. Tran proposes Petri Nets generated from the captured interaction data. These nets are generally complicated and difficult for the evaluator to explore, especially when the interaction sequence includes many user actions.

At the beginning of this paper, we introduced our framework as generic, configurable and flexible with multiple evaluation techniques. In the feasibility study, we validated the fact that the framework is flexible and does indeed have multiple evaluation techniques. We were not able to verify the generic aspect because the IAS only has a WIMP UI. In addition, we used a general module for the evaluation, and so did not benefit from the modular architecture.

6 Conclusion

The evaluation of the interactive part of systems has been the subject of many studies over the last 30 years. However, UI evaluation remains a difficult task to develop. Among the difficulties of evaluation, we can highlight the choice of the evaluation technique and the tool to be used. Note that each technique has its own characteristics and covers a given number of aspects. There are numerous UI evaluation tools. However, evaluators are still faced with some limitations in using these tools. The encountered limitations enabled us to design the RITA framework. This framework is structured according to a modular architecture so that it can be used to evaluate different UIs. Such an architecture can be partially or fully integrated into other evaluation environments and platforms. Furthermore, evaluators can simply use some of the proposed framework tools for evaluating interactive systems according to the needs, constraints and specifications of the evaluation process.

The limitations of existing evaluation tools include EG hardcoding. The majority of existing tools hardcode EGs into the evaluation engine. In our framework, the EGs, as mentioned below, are modeled as external XML files. This enables evaluators to add new EGs or modify existing ones. However, the proposed editor models the EGs one by one. It does not model several EGs into a single structure. As highlighted by [Farenc, 96], there are many EGs that cannot be modeled, such as those pertaining to the clarity of the error messages. This limit pertains with the proposed framework.

Another limitation is on the choice of the appropriate technique for UI evaluation. There are many techniques, each with specific characteristics and evaluation methods. The evaluation results may converge, depending on the technique used. For instance, an interface can be validated using a set of EGs yet still remain difficult for a user to use.

In this framework, we used three different techniques: the electronic informer, the ergonomic quality inspection and the questionnaire. Subsequently, the proposed framework incorporates many aspects for evaluation. First, it focuses on the static presentations of information to the user (i.e. the compliance of the graphical controls regarding the EG). Second, it considers the interaction with the user (e.g., response time, action sequencing for the execution of a task deduced from the capture of executed actions by the users). Thus, the evaluation process considers the performance of the user interacting with the interface. Third, the Framework incorporates a user appreciation about the evaluated interface by considering the user answers to a questionnaire. The provided evaluation result is a report with the results of the three evaluation techniques. However, using these techniques reveals some technical difficulties. The informer requires adding lines of code to the source code of the evaluated system. It is not obvious that the evaluator can access the system code. The evaluator must also proceed with an ergonomic quality inspection page screen by page screen. Another limitation of this framework is that it does not support all kind of UI (e.g. multi-screen UI and touch-screen UI are not supported for the evaluation).

The evaluation report is proposed to the evaluator in several formats with three main types of results. The first type is on UI design utility and usability problems. The second consists of a list of recommendations and suggestions to improve the UI. The third provides information on performing the evaluation process (e.g., unreadable file, truncated frame received).

In this framework, the capture phase and the analysis phase are automated. The critical phase is partially automated and requires the intervention of the evaluator to produce the evaluation results. There are still difficulties in automating this critique phase. This is due to the fact that, like software engineering, there is no universal solution or canonical approach. Thus, it is difficult, or even impossible, to conduct this phase automatically without any intervention of human operators [Nielsen, 93]. The use of this evaluation framework requires stakeholders: an evaluator, a human factors expert, an end user and a designer. Then, this framework has the same limitation related to automating critique phase as existing evaluation tools.

In the future, we intend to improve the provided evaluation report. The report should be generated using a dedicated format, such as EARL[8] or RDL[9].

In addition, we propose quantifying the quality of UIs. This could be interesting when comparing several design alternatives. This present version of the framework does not enable evaluators to compare between different evaluations. It should be interesting to save the different evaluation results through a model to establish an evaluation benchmarking. Subsequently, we propose adapt some of the techniques used in decision-making. We propose also deploying the framework using service-oriented architecture in to ensure better interoperability [Erl, 07].

---

[8] EARL (Evaluation And Report Language) is a W3C standard for evaluation reports. It essentially aims to present a report in four parts: the evaluator, the test subject, the adopted test and the test results (http://www.w3.org/TR/EARL10-Schema/).

[9] RDL (Report Definition Language) is a report format proposed by Microsoft. It involves a common pattern of reports to facilitate their comprehension and use. Its aim is to improve interoperability. It is based on the XML markup language (http://msdn.microsoft.com/en-us/library/dd297486.aspx).

Today, post-WIMP interfaces are more common in daily life (e.g., multi-touch interfaces, tangible interfaces). Nevertheless, there is no model, technique or tool to validate and test the interactions provided by such interfaces. We propose expanding the scope of the RITA features by supporting the evaluation of post-WIMP UIs (for instance, tangible objects or new types of interactive applications on tabletops) and distributed UIs [Penalver, 13]. Furthermore, we intend to integrate other evaluation techniques such is the case of eye tracking method [Zhou, 14; Raschke, 13] and to consider the accessibility factor in Web user interface [Miñón, 14].

As mentioned in 3.2.3, the evaluation framework considers the tasks' sequence, frequency and control links. We intend to consider the tasks importance while gathering and analyzing the interaction sequence.

Acknowledgements

The present research work is partially supported by the International Campus on Safety and Inter-modality in Transportation (CISIT), the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research, and the CNRS.
The authors would like to thank the anonymous reviewers for their helpful and constructive comments that greatly contributed to improving the final version of the paper. They would also like to thank the Editors for their support during the review process.

References

[Abascal, 04] Abascal, J., Arrue, M., Fajardo, I., Garay, N., Toms, J.: The use of guidelines to automatically verify Web accessibility, Univers. Access Inf. Soc., 3(1), 71-79, 2004.

[Abran, 03] Abran, A., Khelifi, A., Suryyn, W.: Usability Meanings and Interpretations in ISO Standards, Software Quality Journal, 11(4), 325-338, 2003.

[Bastien, 99] Bastien, J. M. C., Scapin, D. L., Leulier, C.: The Ergonomic Criteria and the ISO 9241-10 Dialogue Principles: A pilot comparison in an evaluation task. Interacting with Computers, 11, 299-322, 1999.

[Beirekdar, 02] Beirekdar, A., Vanderdonckt, J., Noirhomme-Fraiture, M.: A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code, Computer-Aided Design of User Interfaces III, Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces, May, 15-17, Valenciennes, France, 337-348, 2002.

[Brinck, 02] Brinck T., Hofer E.: Automatically evaluating the usability of web sites, in 'CHI '02 extended abstracts on Human factors in computing systems', ACM Press, New York, USA, 906-907, 2002.

[Brooke, 96] Brooke, J.: Sus: A "quick and dirty" usability scale, In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), Usability Evaluation in Industry, 189-194, London : Taylor and Francis, 1996.

[Calvary, 03] Calvary, G., Coutaz, J., Thevenin, D, Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for multi-target user interfaces, Interacting with Computers 15(3): 289-308, 2003.

[Charfi, 11] Charfi, S., Ezzedine, H., Kolski C., Moussa, F.: Towards an Automatic Analysis of Interaction Data for HCI Evaluation Application to a Transport Network Supervision System, in Julie Jacko, ed., 'Human-Computer Interaction. Design and Development Approaches', Springer Berlin / Heidelberg, 175-184, 2011.

[Charfi, 13] Charfi, S., Ezzedine, H., Kolski, C.: RITA: a Framework based on multi-evaluation techniques for user interface evaluation, Application to a transport network supervision system, ICALT, International Conference on Advanced Logistics and Transport (29-31 May), IEEE, Sousse, Tunisia, 263-268, 2013.

[Charfi, 14] Charfi, S., Trabelsi, A., Ezzedine, H., Kolski, C.: Widgets dedicated to user interface evaluation, International Journal of Human-Computer Interaction, 30, 408-421, 2014.

[Erl, 07] Erl, T., SOA Principles of Service Design, Prentice Hall; 1 edition, 2007.

[Ezzedine, 08] Ezzedine, H., Bonte, T., Kolski, C., Tahon, C.: Integration of traffic management and traveller information systems: basic principles and case study in intermodal transport system management, International Journal of Computers, Communications & Control (IJCCC), 3, 281-294, 2008.

[Farenc, 96] Farenc, C., Liberti, V. Barthet, M.F.: Automatic Ergonomic Evaluation: What are the Limits?, CADUI'96 2nd International Workshop on Computer-Aided Design of User Interfaces - Namur, Belgique, June 5-7,159-170, 1996.

[Galitz, 07] Galitz, W.O.: The essential guide to user interface design: an introduction to GUI design principles and techniques, 3rd edition, John Wiley & Sons, Inc., New York, 2007.

[Grammenos, 00] Grammenos D., Akoumianakis D., Stephanidis C.: Integrated support for working with guidelines: the Sherlock guideline management system, Interacting with Computers, 12(3), 281- 311, 2000.

[Henninger, 97] Henninger, S.: Creating Organization-specific Usability Guidelines, CHI '97 Extended Abstracts on Human Factors in Computing Systems, ACM, 287-288, 1997.

[Hilbert, 00] Hilbert, D., Redmiles, D.: Extracting usability information from user interface events, ACM Comput. Surv, 32(4), 384-421, 2000.

[Hornbæk, 06] Hornbæk, K.: Current practice in measuring usability: Challenges to usability studies and research, Int. Journal of Human-Computer Studies, 64(2), 79-102, 2006.

[Hwang, 10] Hwang, W., Salvendy, G.: Number of people required for usability evaluation: the 10±2 rule, Commun. ACM, 53(5), 130-132, 2010.

[ISO, 98] International Standard Organisation. ISO 9241-11:1998. Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability, 1998.

[Ivory, 01] Ivory, M., Hearst, M.: The state of the art in automating usability evaluation of user interfaces, ACM Comput. Surv, 33(4), 470-516, 2001.

[Joshi, 10] Joshi, A., Sarda, N.: Evaluating Relative Contributions of Various HCI Activities to Usability, In Bernhaupt, R., Forbrig, P., Gulliksen, J. & Larusdattir, M. (Eds.), Human-Centred Software Engineering, Springer Berlin Heidelberg, 6409, 166-181, 2010.

[Karat, 94] Karat, C.M.: A comparison of user interface evaluation methods in Usability inspection methods, edited by Nielsen, J. and Mack, R.L., John Wiley & Sons, Inc., New York, NY, USA, 203-233, 1994.

[Kirakowski, 96] Kirakowski, J.: The software usability measurement inventory: background and usage, P. Jordan, B. Thomas e B. Weedmeester (eds.), Usability Evaluation in Industry. London: Taylor & Francis, 169-178, 1996.

[Kolski, 91] Kolski, C., Millot, P.: A rule-based approach to the ergonomic "static" evaluation of man-machine graphic interface in industrial processes, International Journal of Man-Machine Studies, 35 (5), 657-674, 1991.

[Koua, 04] Koua, E.L., Kraak, M.J.: A usability framework for the design and evaluation of an exploratory geovisualization environment, In Information Visualisation, London, UK, 2004.

[Lavery, 97] Lavery, D., Cockton, G., Atkinson, M.P.: Comparison of evaluation methods using structured usability problem reports. Behaviour and Information Technology 16(21), 246–266, 1997.

[Leporini, 08] Leporini, B., &Norscia, I. (2008) Fine Tuning Image Accessibility for Museum Web Sites; Journal of Universal Computer Science, 19, 3250-3264, 2008.

[Mariage, 04] Mariage, C., Vanderdonckt, J., Beirekdar, A., Noirhomme, M.: DESTINE: Outil d'aide à l'évaluation de L'ergonomie des sites web, Proceedings of the 16th Conference on Association Francophone d'Interaction Homme-Machine, ACM, 117-124, 2004.

[Mariage, 05] Mariage, C.: MetroWeb: A tool for supporting the evaluation of web sites ergonomic quality, (in French) PhD Thesis, UCL, Louvain-la-Neuve, 2005.

[MacDonald, 14] MacDonald, C. M., Atwood, M. E.: What Does it Mean for a System to be Useful? An Exploratory Study of Usefulness, In Proceedings of the 2014 Conference on Designing interactive systems (DIS 14). ACM, New York, NY, USA, 885-894, 2014.

[McNamara, 11] McNamara, N., Kirakowski, J.: Measuring user-satisfaction with electronic consumer products: The Consumer Products Questionnaire, International Journal of Human-Computer Studies, 69(6), 375-386, 2011.

[Miñón, 14] Miñón, R., Moreno, L., Martínez, P., Abascal, J.: An approach to the integration of accessibility requirements into a user interface development method, Science of Computer Programming, 86, 58-73, 2014.

[Nielsen, 93] Nielsen, J.: Engineering, Usability, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993.

[Nielsen, 94] Nielsen, J.: Usability inspection methods, Conference Companion on Human Factors in Computing Systems, CHI '94, Boston, Massachusetts, USA, 413-414, ACM, 1994.

[Olsen, 07] Olsen, J., Dan, R.: Evaluating user interface systems research, in Proceedings of the 20th annual ACM symposium on User interface software and technology, ACM, New York, USA, 251-258, 2007.

[Peñalver, 13] Peñalver, A., López, J-J., Botella, F., Gallud, J-A.: Defining Distribution Constraints in Distributed User Interfaces, J.UCS (Journal of Universal Computer Science), 19, 831-850, 2013.

[Preece, 94] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: Human-Computer Interaction. Addison-Wesley, Wokingham, 1994.

[Raschke, 13] Raschke, M., Blascheck, T., Burch, M., Huang, W.: Visual Analysis of Eye Tracking Data, In Huang, W. (Ed.) Handbook of Human Centric Visualization, Springer New York, 391-409, 2014.

[Rohrer, 14] Rohrer, C.: When to Use Which User-Experience Research Methods, http://www.nngroup.com/articles/which-ux-research-methods/ on October 12, 2014.

[Rubin, 08] Rubin, J. Chisnell, D.: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, Wiley Publishing, 2nd edition, 2008.

[Rukshan, 11] Rukshan A., Baravalle A.: A quantitative approach to usability evaluation of web sites, in Advances in Computing Technology, London, United Kingdom, 2011.

[Schmettow, 08] Schmettow, M.: Heterogeneity in the usability evaluation process. In: M. England, D. & Beale, R. (ed.), Proceedings of the HCI 2008, BCS, 1,89-98, 2008.

[Shneiderman, 2000] Shneiderman B.: Universal usability, Communications of the ACM, 43(5), 84-91, 2000.

[Senach, 90] Senach, B.: HCI ergonomic evaluation: a state of the art, Technical report, INRIA N°1180, 1990.

[Stone, 05] Stone, D., Jarrett, C., Woodroffeet, M., Minocha, S. User Interface Design and Evaluation, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[Tarby, 09] Tarby J., Ezzedine H., Kolski C.: Prevision of evaluation by traces during the software design of interactive systems: two approaches compared, In A. Seffah, J. Vanderdonckt, and M. Desmarais (Eds.), Human-Centered Software Engineering: Architectures and Models-Driven Integration, Springer HCI Series, 257-276, 2009.

[Trabelsi, 06] Trabelsi, A., Contribution to the evaluation of agent based interactive systems. Application to urban transport supervision system (in French), PhD Thesis, University of Valenciennes and Hainaut-Cambresis, France, 2006.

[Tran, 08] Tran, C., Ezzedine, H., Kolski, C.: Evaluation of Agent-based Interactive Systems: Proposal of an Electronic Informer Using Petri Nets, Journal of Universal Computer Science (J.UCS), 14(19), 3202-3216, 2008.

[Vanderdonckt, 05] Vanderdonckt, J., Beirekdar, A.: Automated Web Evaluation by Guideline Review, Journal of Web Engineering, 4(2), 102-117, 2005.

[Watzman, 02] Watzman, S.: Visual design principles for usable interfaces, In Sears A, Jacko J.A. (Eds.), The Human Computer Interaction Handbook, L. Erlbaum Associates Inc. Hillsdale, NJ, USA, 263-285, 2002.

[Whiting, 08] Whiting, M.A., Haacket, J. Varley, C.: Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software, In Proc. Conference on Beyond Time and Errors: Novel Evaluation Methods For information Visualization, 1-9, 2008

[Wickens, 04] Wickens, C., John, D. L., Liu, Y., Becker, S.: An Introduction to Human Factors Engineering, Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

[Williams, 09] Williams L. Beginning XSLT and XPath: Trans-forming XML Documents and Data, John Wiley & Sons Ltd, 2009.

[Zhang, 99] Zhang, Z., Basili, V., Shneiderman, S.: Perspective-based Usability Inspection: An Empirical Validation of Efficacy, Empirical Software Engineering, 4, 43-70, 1999.

[Zhou, 14] Zhou, S. R.., Jeon, S., Sim, H., Lee, W.: User interface evaluation method using eye tracking, The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014), 1-2, 2014.