**Chapter 1: Introduction**

In this chapter, we introduce the research context for our work, problem statement, the research objectives, and thesis contributions that we have proposed to resolve the required issue. We conclude this chapter by describing the structure of this report.

* 1. **Research context**

The adaptation of the interfaces has become a challenge for developers to improve the interactions between the user and the system. It aims to reduce the rate of errors and to enhance the satisfaction of the users.

In such a way, (Lavie and Mayer, 2010) define the adaptation of user interfaces as the process that adapt the displays and the actions of the user interfaces to the user’s current goals and abilities.

In this context, many works are proposed about context modeling, design of adaptive user interface and tools for adaptive user interface generation. But, there exist very few proposals in the literature about adaptive user interface evaluation.

So, the evaluation of adaptive user interface is considered as a new defy for several researchers in the field of adaptation.

According to (Biancalana et al, 2013), the evaluation assesses the effectiveness of the user interface based on preferences of users in the context of use. So, it tests the capacity of adaptation of these user interfaces to deduct the adequacy between the adaptation of these interfaces and the description of users’ needs.

Nowadays, many research approaches in the field of adaptation (Alshaali, 2011; Heukelman, 2012; Navarro, 2014; Fez et al, 2014 and Marcus, 2015) are focused on adaptive user interface taking in consideration different profiles. In fact, the wide-spread use of the adaptive systems became a need for the users, because it allows them to look for their information anywhere and at any time. In addition, adaptive user interface intend to create an interaction between human and computers by customizing the context of use. Moreover, it has been applied in many areas such as education, transport, medical treatment …

In this chapter we are going to introduce the research question of this master’s thesis along with the organizational structure of this document.

* 1. **Problem statement**

The adaptation of the interface drew the attention of several researchers to generate a personalized user interface. Then, evaluating this adaptive user interface becomes a new challenge for many researchers who try to improve the quality of adaptive interfaces.

In this master thesis, we highlight the different problems and challenges that are mainly related to the evaluation of user interface based on several quality metrics to generate a set of evaluation rules.

Problem 1: *Difficult to provide a rigorous definition for the context of use*: In general, it is necessary to define primarily the context of use because there are:

* A diversification of user's preference (age, motivation, experience…) and of its token values (Low, Medium, High) that differs from user to another.
* A diversification of platform of interaction between user and system (PDA, Smartphone…).
* A diversification of environment of interaction (location, temperature …).

Problem 2: *Difficult to define a consensual about evaluation metrics*: the most of existing criteria devoted to ergonomic evaluation. However, there is a difference between detecting ergonomic defect and adaptation problem. In fact, the metrics of evaluation of the user interface is diversified and devoted to ergonomic evaluation, but there is no metrics of evaluation denoted for the adapted user interface. Thus it is important to improve the existing metrics and to adapt them to the context of the adapted user interfaces.

Problem 3: *difficult to define defects of adaptive interfaces:* There is no consensus about the possible problem and there equivalent evaluation metrics. In fact, the same evaluation metrics could be associated to many problems.

Problem 4: *Difficult to define evaluation rules*: There is a consensus that the generation of the evaluation rules is important to assess the quality of such interface. Indeed, most of existing approaches propose to define manually a set of evaluation rules by an expert. However, the manual generation of evaluation rules is difficult according to the diversification of the context of use and it can take a lot of time. Thus, it is necessary to propose an automated process to extract these rules.

The following sentence represents the question of this research:

*How to generate an evaluation rules for adaptive user interface?*

**1.3 Research objectives and main contributions**

**1.3.1 Objectives**

The aim of our research is to generate automatically the evaluation rules of adaptive user interface. This generation is performed by a heuristic approach based on a set of evaluation metrics. So, we propose an evaluation method to verify the quality of the interface by considering their context of use. In addition, this context is modeling by three main dimensions: User, Platform, and Environment.

The purpose of this evaluation is to verify the adaptive systems if it corresponds to the specific needs defined by the user and to validate the feasibility of the adaptive user interface.

**1.3.2 Contributions**

Our work includes two contributions:

* ***Proposal of a set of quality metrics for adaptive user interface***: to assess adaptive interface we propose to adapt a set of exciting evaluation metrics inspired from (Buanga, 2011).
* ***Generation of evaluation rules***: Due to the variety of context criteria and the large list of quality metrics, we propose to extract an interesting evaluation rules basing on a heuristic algorithm.

**1.4 Thesis organization**

As mentioned, the chapter 1 is an introduction of this master report. It describes the research context, problem statement, the objectives and the mains contributions.

The chapter 2 presents a review of literature about adaptive user interface. It defines also an evaluation of adaptive user interface.

The chapter 3 presents the overview of our proposal. It starts by proposing a set of quality metrics.

The chapter 4 presents the experimental result to validate our approach.

The chapter 5 is the conclusion of our thesis and it presents the limitation of our contribution and the perspective of ours research.

**Chapter 2: Related Work**

In this chapter, we will present some definition about the adaptation of user interface. Then, we will define the context of use in adaptive user interfaces, and we will conclude by a definition of the evaluation of user interfaces, the existing method and metrics of evaluation of this user interfaces.

* 1. **Use of adaptation**

The new generation of interaction tools and the multiplicity of information source give rise to several problems for users. These problems have arisen in how to access to the right information at the right time and in the right way. So to solve this issue, we need to adapt application to user’s perspectives (Vargin et al, 2015).

The adaptation consists in simplifying the use of the computer systems by presenting to the users the expected information and by reducing the complexity of this system in order to make it more usable (Rouillard, 2008).

According to (Dyche, 2002), adaptation is the abilities of systems to provide adapted interfaces to user’s needs and expectations. This adaptive user interface based on the preferences and behavior of the user at the time of interaction with the system. So adaptive systems has many benefits for users:

* It meets the need of users.
* It improves the interaction between users and the system.
* It enhances the usability of systems by making it more efficient, effective and easy to use.

There are several factors that have influence on the process of adaptation, (Van Setten, 2001) regroups it into four factors:

* User: Concerns the specific information of users such as: their centers of interest, their preferences, their physical characteristics, their purposes, etc. (Mnasser et al, 2014)
* Content: Describes the characteristics and properties of the contents which define the adaptation of information (Garido and Morales, 2014).
* Context: Defines the technical environment of system as well as the environment of the user which have an influence on the process of adaptation ( Lemos et al, 2013).
* Method of adaptation: Present the method and the tools used to realize the adaptation which has an influence on the possibilities and the results obtained from this adaptation (Walker et al, 2013).

As we are mentioned, the adaptation of user interface depends on the context of use which is considered as one of the major factor which has impact on the adaptation.

* 1. **The context of use:**

The notion of context is defined in (Dey et al, 2000), (Dey, 2001):“*The context represents all the information which can be used to characterize a situation or an entity. An entity is a person, a place or an object which is considered as significant at the level of the interaction between the user and the application, by including the user and the application itself there.”*

According to the definition of adaptation proposed by (Ledoux, 2001), the process of adaptation modifies systems to perform adequately in a specific context, which means that it suits perfectly user needs in this context.

(Brossard, 2008) classify the context in the field of interactive application in two levels:

* The adaptation of the interface, it is described by the plasticity. (Oliveira et al, 2015) define plasticity, by *« the capacity of the interfaces to adapt themselves to their context of use in the respect for their use. The context of use defines themselves as the triplet user, platform and environment»*
* The content adaptation or *awareness context*. The awareness context is "*The use of the context to supply appropriate information and/or services(departments) to the user; the context was any information which can be used to characterize the situation of an entity which can be a user, an environment, a physical or IT object*".(Motti and Vanderdonckt, 2013).

According to (Van Setten, 2001), we can present the context in three levels:

* An application level: wherein the proposed adaptation can depend on the performance of the adaptive system’s application.
* A technical level: wherein many technical constraints (such as: the used platform of consultation, the networks capacities and technical infrastructure of the application) are considered during the process of adaptation (Jeffrey et al, 2011).
* An environmental level: wherein many environmental factors (the physical environment in which the application is used, the place of access to the application and the moment when the application is performed) can impose several constraints on the process of adaptation (Jeffrey et al, 2011).

In our work, we consider the context of use as a triplet ***“User, Environment, Platform”***:

* The user represents the person who interacts with the system that can have an impact on the context of use in terms of his needs (Akiki, 2014).
* The platform establishes the set of the material and software resources used by user. Variability in screen size is an example of aspects that simulate the behavior of platform (Bouchelligua et al, 2010).
* The environment represents the set of objects, people and events which can have an impact on the system, such as distance from display devices that require an adaptation of user interface (Chapuis et al, 2014).

**2.2.1 User Profile:**

Each user has specific characteristics, which may be associated to physical and cognitive factors. So it is fair to say that users are not homogeneous. To highlight these differences, (Lemos et al, 2013; Browne, 2014 and Thalmann, 2014) consider user as the basis of the process of adaptation, and they classify his characteristics to identify their influence in the user’s performance during his interaction with the system.

Thus, the user profile is represented by attributes and functions (such as age, experience, education level, competence.etc.) which describe the user preference of the system (Parru-Arnau, 2014).

The use profile is collected into five categories that describe the interaction between user and system, (Bacha et al, 2011):

* Demographic information: include the demographic data of the user which do not change in time (UMO, 2003), (Kostadinov, 2008) and (Jrad et al, 2007).
* Contact information: contains personal data that can be changed (Rousseau et al, 2004) and (Lin et al, 2007).
* User preferences: contains the preference and the interests of user .Some authors use only the term “preference” ( Kostadinov, 2008), (Preuveneers et al., 2004 ) and (UMO, 2003) while others use the term “interest” (Rousseau, 2008).
* User state: represents the state of user during the interaction with the system. This state may be emotional, physiological (Schmidt et al, 1999) and (UMO, 2003) or an activity practiced by the user (Kim and Choi, 2006).
* User abilities and proficiencies : contains the knowledge, skills and abilities of users (UMO, 2003)

**Table 2.1. Concepts of User Profile**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Categories** | **Demographic information** | **Contact information** | **User preference** | **User state** | **User abilities** |
| **Descriptions** | -date of birth  -Gender  -occupations  -affiliations | -the city  -country  -email  -phone number | -preference profiles  -Interest… | -physiologic state  -emotional state  -mental state… | - ability to talk  - ability to see  -ability to hear … |

**2.2.2 Environment:**

This Environment describes all information where the interaction between the user and the platform takes place. Most of the information related to this model are dynamic and can impact the content to be presented.

(Korpipää et al., 2003; Preuveneers et al., 2004; Arabshianand and Schulzrinne, 2006 and Kostadinov, 2008), noted that this dimension is composed of two main categories. The first is “*Location”*, refers to the place where the user is located at the time of interaction with the platform. The second considers the “*time”*, which indicates the moment of interaction with the platform (Korpipää et al., 2003; Arabshianand, 2006and Hobbs and Pan, 2006).

**Table. 2.2. Concepts of Environnement.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | **Location** | **Time** | **Excremental event** |
| **Descriptions** | -Geometric (GPS)  -Geographic place  -Absolute position  -relative position | -Temporal entity  - Interval  - Duration  -Description | - Sound  - Light  -Temperature values |

**2.2.3 Platform:**

It describes the platform that the user will interact with it. According to( FIPA, 2001; Preuveneers et al., 2004; UsiXML, 2007; Kostadinov, 2008 andW3C, 2009) the platform include two parts hardware and software:

* The *hardware* represent the physical aspect of the platform and it is composed into four parts (Memory, CPU, Network, User interface) (FIPA, 2001; Preuveneers et al., 2004 and Taconet and Aoul,2008 )
* The *Software* part defines the software side of the platform and is composed of four subparts ( virtual machine, application system , operating system, rendering engine) (Preuveneers et al., 2004).

**Table. 2.3. Concepts of Platform**

|  |  |  |
| --- | --- | --- |
| **Categories** | **Hardware** | **Software** |
| **Descriptions** | -Operating System  -Virtual machine  - Runtime Environment  -Edition, Version | -Memory  -CPU  -Connection  -Keyboard types, network interface, screen size |

* 1. **. Adaptive User Interface:**

The user interface is the important part of a computerized system; it is the components of software applications which are localized between the user and the system. (Browne,2014 and Akiki et al, 2015) define the adaptive user interface by the software entity that improves its ability to interact with a user by constructing a user model based on past experience with that user.

The adaptive user interfaces interacts with users in order to reach their needs more easily, faster and with a higher level of satisfaction.

(Faria et al, 2012) classifies the knowledge obtained by an adaptive interface in four distinct domains: knowledge of the user, knowledge of the interaction (modalities of interaction and dialogue management), knowledge of the task/domain and Knowledge of the system characteristics (Norcio and Stanley, 1989).

(Wooldridge, 2002) classifies the adaptive user interfaces into three main classes:

* The first class involves the addition of adaptation to an existing direct manipulation interface. For example we can design an interface with several commands. (Faria et al, 2012)
* The second class is established of interfaces which are considered as an intermediary between the user and the interface of direct manipulation while filtering information (Ross, 2000).
* The third class is composed by the agent interfaces which provide support to the user by making suggestions and giving advice (Faria et al, 2012).

Thus, in the review of state, there are another class of adaptive interfaces which is designated as Programming by Demonstration since it produce commands with arguments (Cypher and Halbert, 1994) (Ross, 2000).

There are different examples of adaptive user interface in the literature, which focus on the task of *information filtering* (Langley, 1999):

* S YSKILL & WEBERT (Pazzani et al, 1996) presents the user with a list of web pages and it encodes each user model as probabilities that calculate the occurrence of certain words that determine if the person likes or dislikes the topic proposed by the search engines. Moreover, it lets the user mark pages as desirable or undesirable, and the system records the  
  marked pages as training data for learning the user’s preferences
* RINGO (Shardanand and Maes, 1995) is an interactive system that recommends movies a person might enjoy, it requires the user to rate a series of sample movies, from which it constructs a simple profile. RINGO then finds other people who have similar profiles to the current user and recommends films that they liked but that the current user has not yet rated.
* FAB (Balabanovic, 1998) retains profiles both for individual users and for topics, and that combines their predictions to give both content-based and collaborative behavior.
  1. **Adaptability/ Adaptativity:**

(Jameson, 2003) classifies adaptation into two principal categories which are:

* Adaptation to the environment or also *Plasticity* “*Denote the capacity of system to adapt itself to the material and environmental constraints in the respect for its usability*.” ( Thévenin et al, 2003).
* Adaptation to the user profile that can be done in two ways:
* Static adaptation: denote the adaptability of a system that is based on a static model of user which is used to give proposed options for the user.
* Dynamic adaptation: denote the adaptativity of a system that is based on transparent adaptation proposed by the system itself which has the initiative of evolve the behavior of user interface.

According to (Moisuc, 2007), the adaptability is the capacity of the interface to adapt itself to the customizations expressly asked by the user, whereas the adaptativity indicates its capacity to meet the needs of the user without an explicit intervention from him.

The term adaptability refers to the interface that is customized by an explicit intervention of the user which can act on the parameters fixed by the designer. In this type of system, the decisions of adaptation are taken by the user because he changes the features of the system. (Brusilovsky and Milán, 2007)

The term adaptativity refers to the interface that performs adaptation automatically without intervention of the user. In this interface, the decisions of adaptation are taken by the system itself. (Brusilovsky and Milán, 2007)

**2.5. Evaluation of adaptive user interface:**

**2.5.1. Definitions of evaluation:**

In this section, we try to present the definition of adaptive user interface evaluation and the set of criteria that are used to evaluate it. Moreover, we going to extract form the literature the set of metrics which are used as the measure of evaluating user interface.

Nowadays, the evaluation of adaptive user interface became a challenge for many researchers ( Verbert et al, 2012; Pu et al, 2012 and Preece et al, 2015). They try to develop their own method to evaluate interactive systems based on their context of research .As a result, we find in the review of literature a diversity of definition and method of adaptive user interface evaluation.

(Paramythis, 2009) considers that adaptive systems were evaluated using methodologies, techniques and assessment tools intended for general interactive systems. The evaluation of user interface consist to verify whether the user is able to achieve his task using a given communication system.

(Preece et al, 2015) define evaluation as “*a mechanism that verifies if interactive systems correspond to the specifications stemming from the definition of user’s needs. It is validated if it corresponds to needs by respecting the constraints of the field of application; otherwise, its inadequacies with regard to criteria identified in priori must be put in evidence*”.

(Gena and Weibelzahl, 2007) classifies evaluation into two types:

- **Summative evaluation:** this type of evaluation aims at implementing the evaluation approach during the cycle of development of an adaptive system. It is considered as an empirical validation of the final results of the execution of this system.

-**Formative evaluation:** this type of evaluation aims at verifying the choice of conception from the beginning of cycle of development before the real implementation of an adaptive system and to obtain indications to revise the conception during the iterative process of development.

According to (Treu, 2012) the evaluation of adaptive interface can serve three goals:

* Verifying the quality.
* Detecting problems.
* Supporting decisions.

(Senach, 1990) classifies the evaluation of user interface in two dimensions:

* **The utility:** it concerns the functional adequacy by answering the question: the interface does allow the user to reach his goals?
* **The Usability**: it concerns the adequacy of the Human-machine interface by answering the question the interface is it easy to learn and to operate?

These notions establish the pillars of the ergonomically evaluation of user interface.

**Utility**

* **functional ability**
* **system performance**
* **quality assistance**

***Evaluation of user interface***

**Usability**

* **ease of learning**
* **ease of use**
* **quality of documentation**

**Fig 2.1. Evaluation of user interface (Senach, 1990)**

**2.5.2. Criteria of evaluation:**

In this context, (Bastien et al, 1993) present eight ergonomic criteria to evaluate and organize the user interface by measuring their utility and usability.

These different criteria are organized as following:

* **Adaptability** “*refers to its capacity to behave contextually and according to the users ‘needs and preferences*”.

The given interface must be adapted to the potential users to achieve their goal.

* **Consistency** “*Certain aspects of an interface should behave in consistent ways at all times for all screens; terminology, icons, colors… should be consistent between screens or within a screen*”.

(Bastien et al, 1993) indicate that the lack of consistency can increase the search time and cause the rejection of users.

* **Compatibility** “*refers to the match between users‘characteristics (memory, perceptions, customs, skills, age, expectations, etc.) and task characteristics on the one hand, and the organization of the output, input, and dialogue for a given application, on the other hand*”.

The given interface must be presented in a directly usable form to enchance the best performance of system.

* **Error Management** “*refers to the means available to prevent or reduce errors and to recover from them when they occur. Errors are defined in this context as invalid data entry, invalid format for data entry, incorrect command syntax, etc*”.

The given interface must increase the number of interruption which have a negative impact on user’s tasks.

* **Explicit Control** “*concerns both the system processing of explicit user actions, and the control users have on the processing of their actions by the system”.*

The given interface will be better if user can have control on it during the interaction.

* **Guidance** *“refers to the means available to advise, orient, inform, instruct, and guide the users throughout their interactions with a computer (messages, alarms, labels, etc.), including from a lexical point of view”.*

The given interface must have a good guidance to facilitate the interaction between user and system and to lead the better performance.

* **Workload** *“concerns all interface elements that play a role in the reduction of the users‘ perceptual or cognitive load and in the increase of the dialogue efficiency”.*

The given interface must have a less level of density to let user accomplish their goal efficiently because the higher level of density causes the higher probability of errors.

* **Significance of Codes** *“qualifies the relationship between a term and/or a sign and its reference. Codes and names are significant to the users when there is a strong semantic relationship between such codes and the items or actions they refer to”.*

The given interface must have a meaningful code to be easier in use.

**2.5.3. Methods of evaluation:**

In this section, we present an overview of methods that can be used to evaluate adaptive user interface and to provide a reviews that modify the adaptation of this user interface.

(Gena et al, 2013) presents a classification of evaluation methods as following:

* **Collection of user’s opinions:** that gathers a several method that is used to extract qualitative and quantitative information from the real user of adaptive user interface in order to evaluate the success of user interface’s adaptations. This methods are :
  + Interview: that collect user’s experience, opinions such as their satisfaction with the adaptive user interface and it’s considered as qualitative information (Park et al, 2013).
* Questionnaires: Those arrange information for user by collecting knowledge from user and the used adaptive user interface (Harrison et al, 2013).
* **Observing and monitoring usage:** that gathers a several method that collects information from usage of adaptive interface and to analyze it in qualitatively and quantitatively manner:
* The systematic observations: this method used to quantifying the user’s behavior in a real context (Almaliki et al, 2014).
* Logging use: it aims to analyze quantitatively the log files that are considered like a register of all the actions of users during their accomplishing task (Haradio, 2012).
* User Observations: it is characterized by the direct and indirect observations of users during their usage of user interface. It is considered like a qualitative method (Flores et al, 2012).
* Verbal protocol: it aims to record the reactions of user when they accomplish a task in experimental sessions (Pankwska, 2012).
* **Predictive evaluation:** That gathers a several method that predicts the usability of adaptive user interface from models or specifications.
* Expert review: it is an evaluation method that considers experts like a less experienced user. It aims to identify the usability issues of adaptive user interface ( Dirin et Neiminen, 2015) .
* Heuristic evaluation: it is an evaluation method that invites users to examine the adaptive user interface in order to extract problems from it (Gena et al, 2013).
* Parallel design: it consist on exploring different solutions for adaptive system proposed by several designers before implementing the further proposal (Ickin et al, 2012).
* Cognitive walkthroughs: it is an evaluation method in which an evaluator proposes task scenarios then play the role of user (Preece et al, 2015).

**2.5.4. Overview of evaluation metrics:**

Usually, many researchers use a set of metrics, which may be adopted in their own research, to evaluate user interface.

The set of metrics that we are going to consider are proposed by (Perlman, 1987; Vanderdonckt and Gillo, 1994; Shneidermanand et al.1994; Ngo et al, 2000; Miyoshi.al, 2001 and Hartmann et al, 2005) .

* (Hartmann et al, 2005) proposed several metrics of aesthetic attributes these are used to enhance the visual balance of an initial label layout through an efficient real-time optimization process (Buanga, 2011).
* (Perlman, 1987) proposed an “axiomatic model” of information layout for alphanumeric displays (Buanga, 2011).
* (Shneidermanet al.1994) proposed thirteen metrics for checking the consistency of a graphical user interface. He works with General Electric Information Services and a group of researchers at the University of Maryland to generated a set of 40 metrics (Buanga, 2011).
* (Vanderdonckt and Gillo, 1994) proposed thirteen visual techniques grouped in five categories (Physical, Composition, Association and dissociation, Ordering, Photographic techniques) to help designers to classify elements of layout in such a way that it is visually attractive (Buanga, 2011).
* (Ngo et al, 2000) proposed fourteen metrics presented in his works (e.g. (Ngo et al., 2000); (Ngo and Byrne, 2001) and (Ngo, Teo and Byrne, 2003)) in which we focus our studies. Ngo proposed initially four metrics (Ngo et al. 2000), (Balance, Equilibrium, Symmetry and Sequence) and he try to evaluate them in an empirical study to prove if these metrics correlate with users’ perception of aesthetics. Then he extended his measure to fourteen (balance, cohesion, density, economy, equilibrium, homogeneity, order and complexity, proportion, regularity, rhythm, sequence, simplicity, symmetry, unity) (Purchase, 2011).
* (Miyoshi et al, 2001) who consider that in GUI design, the measure of complexity of screen layout is achieved by size, local density, grouping and alignment which are influenced on the screen usability. (Fu et al, 2007).

We have just placed our work in the context of the adaptation of the adaptive interfaces by using metrics of quality quoted to develop metrics adaptable to the user interfaces adapted to realize the evaluation of these interfaces.

**2.6. Conclusion:**

In this chapter, we introduce the review of literature to define the adaptation and evaluation of adaptive user interface.

First, we present the definition of context of use in the field of adaptation by describing the user profile, platform, environment .Then we indicate the concept of adaptive user interface by presenting a example of it occurred in several reference .Moreover, we determine the comparison between adaptativity and adaptability. Then, we define the notion of evaluation presenting in the literature by determining the criteria of evaluation used to evaluate user interface. Finally, we present asset of metrics used to measure the utility and usability of user interface.

In the next chapter, we will discuss the set of metrics adaptive to our study. Moreover, we will describe the process of evaluation by defining the rule evaluation form and to present the meta-heuristic approach that will be used to evaluate a set of adaptive interfaces.

**Chapter 3: Extraction of evaluation rules using MOAE/D**

**3.1 Introduction:**

We notice that the evaluation of adaptive user interface is often neglected. So, it is necessary to envisage new specific evaluation methods devoted to adaptive user interface evaluation. Indeed, adaptive user interface evaluation studies should benefit from the existing methods in every development phases to reach more rigorous level in terms of statistical analysis, correctness in procedures, experiment settings, etc.

In this chapter we will present the mains contributions of our work about the generation of an evaluation method for adaptive user interface. In addition, we will describe in the next sections the proposed metrics for adaptive user interface and the overview of the evolutionary algorithm that is used to extract the evaluation rules.

**3.2 Proposed Metrics for AI evaluation:**

The ultimate goal of this work is to develop a tool based on the quality metric to evaluate the adaptive user interface.

We aim to generate evaluation rules to assess the quality of adaptive interface. In this context, we propose a set of evaluation metrics that can be used to evaluate the user adaptive interface: density, grouping, homogeneity, regularity, sequence, simplicity, symmetry, and unity. In fact, these metrics are inspired from (Ngo et al, 2002) and based on several ergonomic criteria proposed by (Bastien et al, 1993). So according to the definition of such criteria, we will group these metrics into two criteria:

* Guidance: grouping, regularity, sequence and simplicity.
* Coherence: unity, symmetry, density and homogeneity.

The values of these metrics are in [0,1]. Where, 0 present the lower value, 1 present the higher value and for the medium we use the average of this interval (0.5) that as the threshold.

* + 1. **Guidance:**

***Guidance*:** user guidance refers to the means available to advice, orient, inform, interact, and guide the users throughout their interaction with computer (message, alarm, label, etc.). This criterion is subdivided into two metrics: Regularity [Ngo et al, 2002] and grouping [Miyoshi et al, 2001].

* + - 1. **Regularity:**

Regularity refers to the consistency of the adaptive interface. This metric can be accomplished by providing a uniformity space between adaptive interface objects, and by minimizing the alignment point which is the number of row and columns of the interface. It aims to organize the structure of these components. The regularity of the adaptive interface is related to many user profile criteria such as age, motivation, education level, etc. In fact, the regularity of the interface provides guidance for novice users having a low education level. This well arrangement of components helps also these users to read in an easier way the content of adaptive interface (AI).So, adaptive interfaces must provide a high regularity level for this kind of users. In order, to measure this metric we will base on the formula proposed by [Ngo et al, 2002]:

RMalignment: is the extent to which the alignment points are minimized.

RMspacing: is the number of distinct distances between column and row starting points.

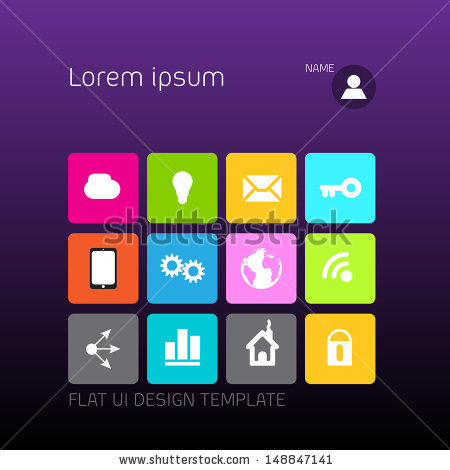
nv = the numbers of vertical alignment points ( number of row).

nh = the numbers of horizontal alignment points( number of column).

nspacing= the number of distinct distances between column and row starting points.

n: is the number of the component of the adaptive interface.

HigherRMalignment values are refers to how well the components are aligned in the adaptive interface. Higher RMspacing valuemeans that the space between components is normalized.



1. (b)

Fig.1: Two versions of adaptive interface having different regularity. (a) Low regularity. (b) High regularity.

* + - 1. **Grouping / composition**

Grouping is a guidance technique which aims to orient and guide user throughout their interaction with adaptive interface. In fact, the component of this adaptive interface that is semantically linked should be seems visually in some way. It calculates the number of objects that own a clear boundary by line, color, etc. This metric enhance the visual clarity which is achieved when the interactive objects are organized and presented in meaningful and understandable manner. It is related to experience and age of user.For example, grouping makes adaptive interface easier to understand by users having lower experience.

The formula of this metric is given by [Miyoshi et al, 2001]:

gi: is the number of groups with clear boundary by line, background, color, or space

: is the total number of groups.



1. (b)

Fig.2: Two versions of adaptive interface in different grouping. (a) The bad grouping. (b) The good grouping.

* + - 1. **Sequence / storing:**

Sequence is the metric that describe the way of how organize the objects in the adaptive system in relation to the eye movement that progress sequentially from dark area to light area, from big object to little object, etc. So it is providing to spruce up the component to lead the eye of user through the adaptive interface in a logical and sequential ordering that refer to the user’s needs. This metric is related to the age, education level and interests of user. For example, Sequence may facilitate the use of adaptive interfaces to user who have old age and lower level of education.

We can calculate it as follow [Ngo et al, 2002]:

With:

UL: upper-left

UR: upper-right

LL: lower-left

LR: lower-right

: is the total weight of quadrant j

: is the area of object i on quadrant j.

Each quadrant is given a weighting in q.So.



Fig.3: good adaptive interface with a high sequence.

* + - 1. **Simplicity:**

Simplicity refers to the optimization of the number of interactive object in the adaptive interface and the minimization of the alignment point. Simplicity helps user to appreciate easily the meaning of expected information. Moreover, only the information related to user’s tasks or needs is to be presented in the adaptive interface.

This metric is related not only to the age, motivation and education level of user but also to the type of platform used by them. For example, simplicity helps old people by giving them a simple interface’s layout, at variance to motivated person that having a higher education level that prefers an interface with low simplicity.

We can calculate it as follows [Ngo et al, 2002]:

nvap= number of vertical alignment points.

nhap= number of horizontal alignment points.

= number of object on the frame.

SMM is conversely proportional to the addition of the numbers of alignment points and the number of interactive objects of the interface so when these summations are decreased, SMM tend to increase. A high simplicity is achieved with the smaller number of component which they are aligned in the adaptive interface.



Fig.4: the same interface in different level of simplicity.

* + 1. **Coherence**

Coherence is the means that used to provide established interaction between users and interface, and secure the efficient use of the interface. This criterion is subdivided into four metrics: unity, density, size and symmetry [Ngo et al, 2002].

* + - 1. **Unity :**

Unity is grouping all interface elements to appear like a one piece. It concerns the adaptation to the platform capacities by which the users interact with the system ( PC, Smartphone, etc.) this property is related to the term plasticity [Thévenin and Coutaz, 1999], [Calvary et al, 1990]. Authors define this term as the capacity of the interface to adapt the context use by respecting its usability. This criterion is related only on to information support (container). In fact, the interface surface of the digital personal assistant is not the same as of a computer that is why it is necessary to adapt the information quantities and form of information, the navigation in the interface, graphic objects placement according to the visualized support. The measure of this metric is determined by the extent to which the components are related in size, and the relative measure of the space between groups (groups of component) and that of margins [Ngo, 2002]. It helps to secure centered element on the interface and avoid its fragmentation.

Well unity obtain by using the optimum number of size component (minimize the uses of different sizes in adaptive interface) and leaving less space between objects. So when the level of unity increases, the adaptive interface is not centered as well.

This metric is given by (Ngo, 2000)

UMform: is the extent to which the objects are related in size.

UMspace: is a relative measure of the space between groups and that of the margins.

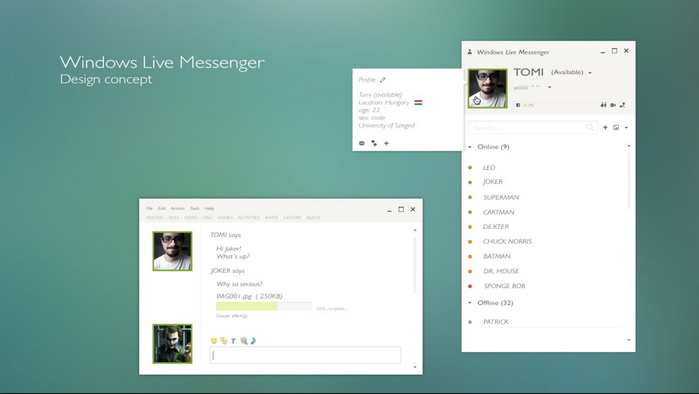
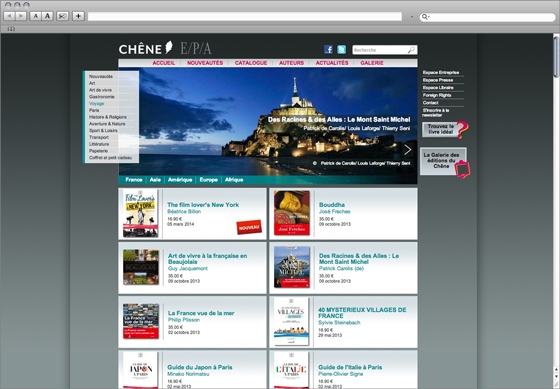
nsize:the number of size used.

n : the number of object.

aAI:the area of the adaptive interface.

aframe: the area of the screen.

ai: the area of the interactive object i.

****

1. (b)

Fig.5: two versions of screens in unity study. (a) Interface with high unity.

(b) Interface with low unity.

* + - 1. **Density:**

Density concerns the workload of adaptive interface. It is a set of information presented to the user. Density is the extent to which the adaptive interface is covered with objects. This metric is related to the motivation, experience and center of interest of users. In fact, user with a low motivation prefers an interface with a low density level. Density can also depend on the type of platform employed.

Density measure (DM) is given by [Ngo et al, 2002]:

ai= area of the interactive object i.

aframe= area of the frame ( screen of interactive platform).

n= number of interactive object.

DM present the percentage of interactiveobject area in the adaptiveinterface that should be equal to 50%, so DM= 1 when density level is 50, and a Higher values are related to how closely the level is 50.



1. (b)

Fig.6: Two versions of adaptive interface with different density levels. (a) Interface with higher density level.(b) Interface with medium density level.

* + - 1. **Symmetry:**

Symmetry is one of metrics who provide the coherence of interface. It gives by an equal distribution of the quantity of elements on the right and the left columns of adaptive interface.

Symmetry consists to duplicate component on the left, right, and radical of the adaptive interface centerline, and avoid the imbalance in the different part of interface.

Adaptive interface should be adapted to the motivation of the users, and it should also stimulate their interests. For example person who has less level of motivation should get an interface with higher coherence.

The Symmetry measure (HM) is given by [Ngo et al, 2002]

SYMvertical, SYMvertical, SYMradial are, respectively the vertical, horizontal and radial symmetries with

X’j, Y’j, H’j, B’j, Θ’j, and R’j are, respectively, the normalized values of

UL, UR, LL, and LR stand for upper-left, upper-right, lower-left, and lower-right, respectively.

Xj: is the total x-distance of quadrant j.

Yj: is the total y-distance.

Hj : is the total height.

Bj: is the total width.

Θj: is the total angle.

Rj: is the total distance

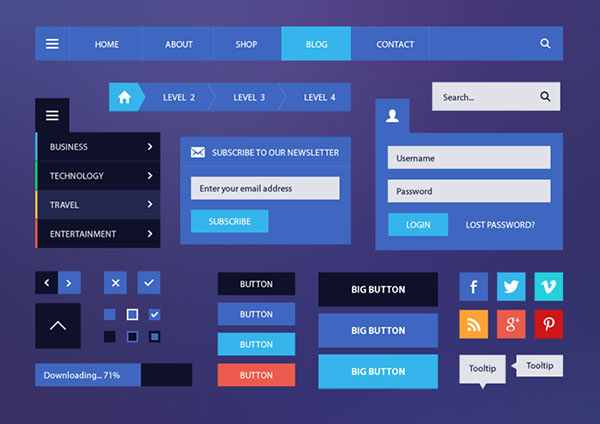
(*xij; yij*) : the coordinates of the centers of object i on quadrant j.

(*xc; yc*) : the coordinates of the frame.

bij: the width of the object.

bij: the height of the object.

nij: the total number of objects on the quadrant.



1. (b)

Fig.7: two versions of screens in symmetry study. (a) Interface with horizontal symmetry.(b) Interface with higher homogeneity.

* + - 1. **Homogeneity :**

Homogeneity is an overall seen of component distribution in adaptive interface, which gives users an equal arrangement of object among the four quadrants. This metric is the comparison between, the numbers of different ways that objects can be organized for the four quadrants, compared to an optimal distribution.

The optimal distribution obtain when the n object are evenly allocated with the quadrants of adaptive interface. However, for n component there are n! different ways to organize them. And in each quadrant (upper-left, upper-right, lower-left, lower-right), nj object can be organized with nj! different ways.

This metric is related to the level of user experience. In fact, adaptive interface should propose an optimal distribution for novice users in order to help them to navigate through it.

The homogeneity measure (HM) is given by [Ngo et al, 2002]

W: is the number of different ways of n object can be arranged for the four quadrants.

Wmax: W is maximum when the n objects are evenly allocated to the four quadrants of the adaptive interface.

n: is the number of object on the adaptive interface.

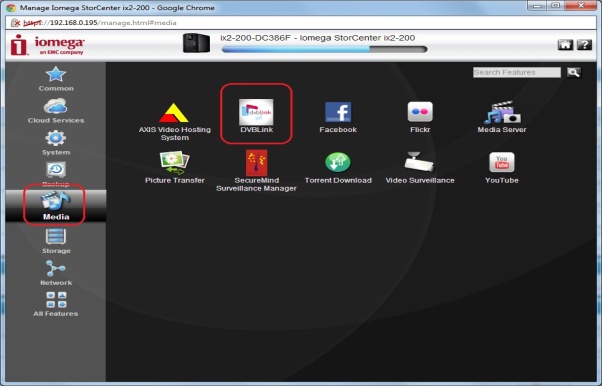
nj: is the total number of objects in quadrant j.

nUL: is the number of object on the upper-left.

nUR: is the number of object on the upper-right.

nLL: is the number of object on the lower-left.

nLR: is the number of object on the lower-right.



1. (b)

Fig.8: two versions of adaptive interface with different level of homogeneity. (a) Interface with low homogeneity. (b) Interface with high homogeneity.

* + 1. **Complexity of this metrics:**

The complexity of this metrics is calculated as follow:

COMG: is the complexity of guidance.

COMC: is the complexity of coherence.

* + 1. **Correlation between metrics and user profile**:

Each metrics is related with set of context, for example, regularity is related with age, motivation and education level. These context and criteria can have the following value: low (-), medium (±) and high (+).

When the values of age and education level are low, adaptive interface should have high regularity level. But when motivation is high regularity is not important and can be with low regularity level.

**Table. 3.1: Mapping between adaptive user interface criteria and context.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Pref**  **MIA** | **age** | **motivation** | **User experience** | **Interest** | **Education level** |
| **Regularity** | **(+,-)** | **(-,+)** |  |  | **(+,-)** |
| **Grouping** | **(+,-)** |  | **(+,-)** |  |  |
| **Sequence** | **(+,-)** |  |  | **(+,+)** | **(+,-)** |
| **Simplicity** | **(+,-)** | **(-,+)** |  |  | **(+,+)** |
| **Unity** |  |  | **(+,-)** |  |  |
| **Density** |  | **(+,+)** | **(+,+)** | **(+,+)** |  |
| **Symmetry** |  | **(+,-)** |  | **(+,-)** |  |
| **Homogeneity** |  |  | **(+,-)** |  |  |

* 1. **Approach overview:**

We propose an approach that uses knowledge from previous empirical study of a several adaptive interface examples to generate a set of evaluation rules. Fig9 shows the general structure of the proposed approach.

Fig.9. Approach overview

Our approach takes as inputs a set of information extracted from a several adaptive user interface used for the evaluation process:

-**The trace** is a base of examples that contains the profiles of the users which we invited to test the various adaptive user interfaces and estimate the problems proven by them during their uses of these interfaces.

-**The context of use** contains the various preferences of users: age, motivation, education level, user experience, interest, and their taken values: low, medium, high.

-**Threshold of metrics** corresponds to the results of the calculation of the proposed metrics for each adaptive user interface generated by the parser.

-**Problems** are the different type of problems of the adaptive interface seen by the users and they are classified according to the values taken by the metrics of these interfaces.

In this context, we fitted the threshold of metric’s values to 0.5. Thus, every interface has a value of metrics superior to 0.5 is an interface which has problems of high quality, while an interface with a value of the metrics lower than 0.5 is considered as an interface which has problems of low quality.

In our study, we consider two types for problems that correspond to high quality or low quality of adaptive user interface basing on the values of the calculated metrics.

This table below contains the proposed metrics and the correspond problems that can be detected according to the values of each metric.

**Table 3.2. Metrics and correspond problems**

|  |  |
| --- | --- |
| **Metrics** | **Correspond problems** |
| **Regularity** | -irregular interface : low quality  -regular interface : high quality |
| **Grouping** | -low guidance : low quality  -high guidance : high quality |
| **Sequence** | -Worst arranged interface: low quality  -well arranged interface : high quality |
| **Simplicity** | -Complex interface : low quality  -Simple interface: high quality |
| **Unity** | - not centered interface: low quality  -Centered interface: high quality |
| **Density** | - Not charged interface : low quality  -Workload: high quality |
| **Symmetry** | -incoherent interface : low quality  -coherent interface : high quality |
| **Homogeneity** | -unequal arrangement interface : low quality  -equal arrangement interface: high |

As outputs, our approach derives a set of evaluation rules from the genetic algorithm. Thus, the generation process can be viewed as a search-based that combines the inputs of this approach to generate these rules. So the best solution of our study is a set of rules that detect the maximum of problems and contains the minimum of rules.

In addition, the quality of this solution is calculated by the fitness function that compare the different generated rules with the examples from the trace.

As many inputs combinations are possible, we have also a huge number of possible solutions derived from these combinations, so we need to use a meta-heuristic search to explore this large number of these combinations.

***-The UI Parser:***

This parser is a set of java classes that used to parse different type of code source of adaptive interface to extract the dimension of each component. It applies the formula of each metrics to generate the threshold of metrics. These thresholds are taken as input for the evolutionary algorithm to generate evaluation rules.

We describe the conception of this parser in a class diagram:

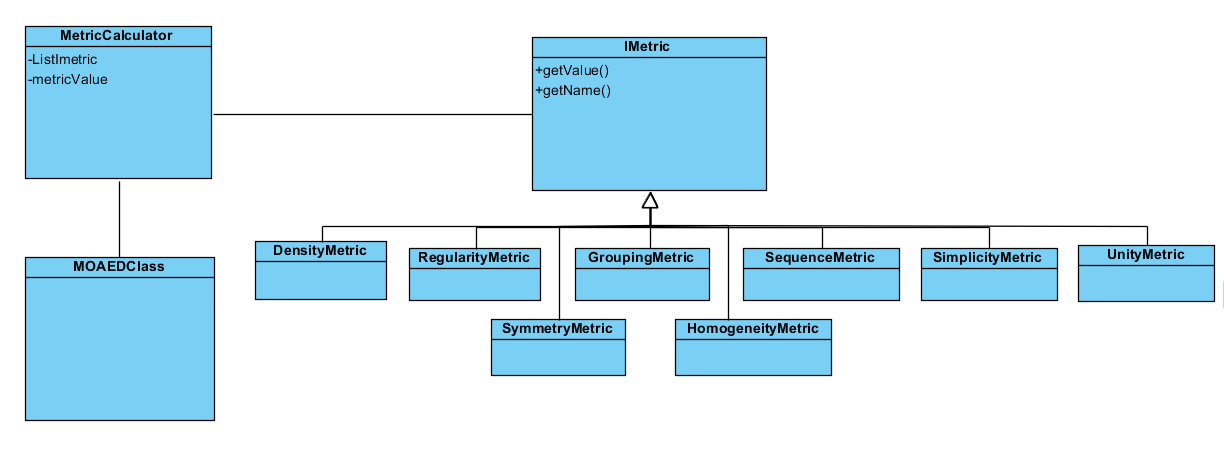


Fig10. Diagram of uses cases of the parser

We can summarize our approach on the following representation:

Fig.11. Global representation of the approach

* 1. **Evaluation rule extraction using MOAE/D:**

This section illustrates the evolutionary algorithm that we are used. In addition, it describes the appropriate adaptations of this algorithm to our work.

* + 1. **Evolutionary algorithm overview:**

An evolutionary algorithm is a stochastic optimization method based on the Darwinian theory of evaluation. It aims to explore a large search space of a specific problem by producing a population of solutions, develop toward the optimal solution that optimizes a fitness function of this problem.

The development of an evolutionary algorithm takes place around three successive stages which are;

* **Selection:** is an essential operator which aims to allowing the best individuals of a population to reproduce. It compares between them based on the values of their fitness functions.
* **Crossover:** is an binary operator acting on a set of individual, it allows the transmission of the characteristics of the best individuals parent to the new individuals children by replacing the randomly chosen dimensions of the individual parent with those of another individual parent to obtain two different children.
* **Mutation:** is the operator which allows the modification, with a certain probability, of one or several nodes of the selected individual, to introduce some variability into the population.

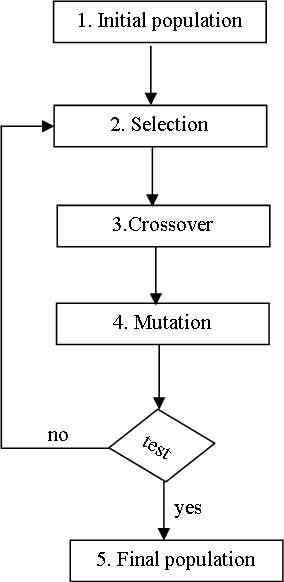


Fig.12. Evolutionary algorithm’s functioning

During the comparison of the population’s individuals, the evolutionary algorithm taken into account four fundamental concepts:

**-Pareto optimality:** the definition of Pareto optimality states that *x*\* is Pareto optimal if no feasible vector *x* exists which would improve some objectives without causing a simultaneous  
worsening in at least another one. ( Bechikh, 2013)

**-Pareto dominance:** a solution 1 is said to dominate another solution 2 if and only if the fitness functions of solution 1 is partially less than the fitness function of solution 2.

**-Pareto set**: is the set of solutions which called non-dominate solutions because they aren’t dominated by any others. The image of this set of solution in the objective space called **Pareto front**.

The evolutionary algorithm has a crucial advantage when it comes to solve a multi-objective problem since it treats a population of solutions instead of a single solution. Indeed, the use of evolutionary algorithm in our work results from the need to generate a set of evaluation rules and to reach this goal we use one of the multi-objective evolutionary algorithms which is MOEA/D.

MOEA/D a multi-objective evolutionary algorithm based on decomposition. It decomposes a multi-objective optimization problem into a number of single objective optimization sub problems and optimizes them simultaneously. (Zhang and Lui, 2007)

To convert this multi-objective problems to sub problems, MOAE/D required the Tchebycheff approach to reach this goal. In this approach, the problem has the form : (Zhang and Lui, 2009)

|}

Where: X is the solution of the sub problem i.

K is the reference point: the vector composed with all the desired values of the objective.

During every generation, the population is a collection of the best solution found for every sub problem. The optimal solutions of two neighboring sub problems should be similar. Every sub problem is optimized basing on information received from its neighboring sub problems.

According to (Zhang and Lui, 2007), MOEA/D has several features:

* MOEA/D introduces the decompositions approaches into multi-objective evolutionary computation.
* MOEA/D solve many issues of multi-objective evolutionary algorithm such as fitness assignment and diversity maintenance because it optimizes many sub problems rather than a whole problem.
* MOEA/D has many advantages then the others evolutionary algorithm in terms of complexity and solution quality.
* MOEA/D can incorporate the objective normalization technique.
  + 1. **Adaptation of MOEA/D:**

Our approach consists on generating a set of evaluation rules for adaptive user interfaces. So to reach this goal, we propose an approach based on a multi-object optimization problem using MOEA/D.

The principal of MOEA/D consists on decomposition of the multi-object optimization problem into N single objective sub problems. This algorithm composed into four steps: (Zhang and Lui, 2009)

* The first step is the initialization wherein MOEA/D computes for each sub problem i H(i) which is the Euclidean distances between any two weight vectors and then find the T closest weight vectors to each weight vector. Moreover, it generates randomly an initial population *x 1,…,xN* . Then, it initialize the reference point *K= (K1,…,Km)* by setting Qi = min{fi(x1), fi(x2),…,fi(xN)}. Finally, it sets the generation size = 0.
* The second step is the selection of sub problems to form a population.
* The third step is the update of the population. So for each i=1…N, MOEA/D selects randomly two indexes *F,P* from *H(i)* , and then it generates a new solution Q from*xF* and *xP* by using crossover and mutation. Then, it applies the fitness function of the problem on *Q and update K =fj(Q’)* if *Kj < fj(Q’)*. In addition, for each index, and it sets.Finally, it removes from each population all the vectors dominated by *F (Q’)* and it adds *F (Q’)* to each population if there are no vectors in each population dominate *F (Q’).*
* The fourth step is the stopping criteria. If there are no vectors dominate *fN (xi) or* the generation size was exceeded, then MOEA/D generates as output the best solutions. Otherwise, it increments the generation size and returns to step 2.

**Inputs:** values of metrics, preference of user, problems of AUI, the trace

**Inputs**: Population-size=P, Generation, Number of sub problem N, Number of weight vector W.

**Outputs**: evaluation rules

1. Initialization of the population.

* Calculate for each i = 1, . . . , N, H(i) = {i1, . . . , iW } where λ i1 , . . . , λiW are the T closest weight vectors to λ i.
* Generate randomly an initial population *x1…xN* .
* Initialize *K=min {fi(x1),fi(x2),…, fi(xN)}.*
* Set Generation-size= 0.

1. Selection of population.
2. Update population*: For i=1,…,N Do*

* Reproduction: Crossover and Mutation
* Evaluation.
* Update of *K.*
* Update of Neighboring Solutions.

1. Stopping Criteria: If there are no vectors dominate *fN (xi) or* Generation-size> Generation, then stop and output each population. Otherwise, Generation-size= Generation-size+1 and go to 2.

Fig.13.Algorithm of Multi-Objective Evolutionary Algorithm/Decomposition *MOEA/D*

The four following subsections describe our adaptation of MOEA/D to our problem that generates a set of evaluation rules for adaptive user interface.

* + - 1. **Individual representation:**

Our research consist on elaborate a set of evaluation rules for adaptive user interface so we need to have a formative form for this rules.

In fact, we can consider that our population is composed of a set of solutions that are also a collection of rules. Our rule presented as a set of IF-THEN.

Consequently, a detection rule has the following representation:

**IF** (Context = *ValueContext*) **AND** (Metric (<,>) *ValueMetric*) **THEN** Problem.

The IF clause correspond to the combination of some preference of users (age, motivation, etc.) with their following possible values (low, medium, high), and a metric with its threshold value using the logic operator AND. Besides, the THEN clause highlights the type of problem detected on the adaptive user interface.

* + - 1. **Generation of an initial population:**

This step takes in input the preference of users (experience, age, motivation, etc.), the calculated metrics resulting from the parser (regularity, density, simplicity. etc.), the trace that contains the bases of examples and the list of possible problems. Note that each attribute of the preference can take several values: low, medium and high and the metrics can take a number between 0 and 1. This step allows extracting an initial population of rules in a random manner from a possible combination of inputs.

**-*Encoding:*** the MOAE/D needs a significant coding of individuals to facilitate the processing of the algorithm. Indeed, the coding can have a significant impact on the way in which examples are processed. The rules will undergo a process of improvement; they must then be presented as an Array List of a new type Rule.

|  |
| --- |
| **R1:** IF (Interest=Low) and ( Sequence<=0.31 ) THEN (not\_arranged\_Interface) |

Fig.14. Encoding rule.

* + - 1. **Evaluation of population:**

In this step, we evaluate the solutions of the initial population in order to estimate and discover the interesting rules. This evaluation is based on the calculation of the fitness function which will determine the quality of each rule by referring to the trace. The quality of generated solutions is evaluated using the fitness function F (x) given in equation (1). The idea is to improve the quality of evaluation rules by reaching the two objectives of the problem:

-maximizing the number of covered GUIs by each solution

- minimizing the number of rules in each solutions.

Our fitness function is calculated using the following equation 1:

Where, x is the decision variables of the problem.

N: number of solution by population.

Nb: number of rules by solution.

Rate: is the rate of problem by solution and it is calculated as following:

* + - 1. **Genetic operators:**
         1. **Crossover:**

The genetic operator crossover used is the single-point crossover that is first applied to the two solutions and generates one child solution. The crossover operator swaps rules from one solution to another.

**Solution1**

|  |
| --- |
| **R1:** IF (Interest=Low) and ( Sequence<=0.31 ) THEN (not\_arranged\_Interface) |
| **R2:** IF (Age=High) and ( Regularity<=0.40 ) THEN (Irregular\_Interface ) |
| **R3:** IF (User Experience=Meduim) and ( Regularity<=0.11 ) THEN (Irregular\_Interface) |
| **R4:** IF (Motivation=Low) and ( Density>=0.82 ) THEN ( charged\_Interface ) |

**Solution2**

|  |
| --- |
| **R1:** IF (User Experience=Meduim) and ( Simplicity<=0.56 ) THEN (complex \_Interface) |
| **R2:** IF (Motivation=Meduim) and ( Grouping<=0.5 ) THEN (Low\_guidance ) |
| **R3:** IF (Age=High) and ( Simplicity<=0.3 ) THEN (complex \_Interface) |
| **R4:** IF (Age=Meduim) and ( Sequence<=0.06 ) THEN (not\_arranged\_Interface ) |

**Child 1**

|  |
| --- |
| **R1:** IF (Interest=Low) and ( Sequence<=0.31 ) THEN (not\_arranged\_Interface) |
| **R2:** IF (Age=High) and ( Regularity<=0.40 ) THEN (Irregular\_Interface ) |
| **R3:** IF (Age=High) and ( Simplicity<=0.3 ) THEN (complex \_Interface) |
| **R4:** IF (Age=Meduim) and ( Sequence<=0.06 ) THEN (not\_arranged\_Interface ) |

**Child2**

|  |
| --- |
| **R1:** IF (User Experience=Meduim) and ( Simplicity<=0.56 ) THEN (complex \_Interface) |
| **R2:** IF (Motivation=Meduim) and ( Grouping<=0.5 ) THEN (Low\_guidance ) |
| **R3:** IF (User Experience=Meduim) and ( Regularity<=0.11 ) THEN (Irregular\_Interface) |
| **R4:** IF (Motivation=Low) and ( Density>=0.82 ) THEN ( charged\_Interface ) |

Fig.15. An example of the crossover process

**3.4.2.3.2 Mutation:**

The genetic operator mutation used is the standard mutation operator. It mutates one rule of solution to produce a new solution.

**Solution X**

|  |
| --- |
| **R1:** IF (Interest=Low) and ( Sequence<=0.31 ) THEN (not\_arranged\_Interface) |
| **R2:** IF (Age=High) and ( Regularity<=0.40 ) THEN (Irregular\_Interface ) |
| **R3:** IF (User Experience=Meduim) and ( Regularity<=0.11 ) THEN (Irregular\_Interface) |
| **R4:** IF (Motivation=Low) and ( Density>=0.82 ) THEN ( charged\_Interface ) |

**Child**

|  |
| --- |
| **R1:** IF (Interest=Low) and ( Sequence<=0.31 ) THEN (not\_arranged\_Interface) |
| **R2:** IF (Age=Low) and ( Regularity<=0.40 ) THEN (Irregular\_Interface ) |
| **R3:** IF (User Experience=Meduim) and ( Regularity<=0.11 ) THEN (Irregular\_Interface) |
| **R4:** IF (Motivation=Low) and ( Density>=0.82 ) THEN ( charged\_Interface ) |

Fig.16. An example of the mutation process

Once reproduction and crossover have been applied according to given probabilities, parents and children are merged and the newly created generation of individuals is evaluated by the fitness function F (x). This process is repeated iteratively, usually for a fixed number of generations. The result of MOAE/D (the best solution found) is the fittest individual produced along all generations.

* 1. **Conclusion**

In this chapter, we present our two contributions of our research which are the proposed metrics of adaptive user interface evaluation and the meta-heuristic used to generate a set of evaluation rules that evaluate a several adaptive interfaces.

In the next chapter, we will discuss the experimentation of our MOAE/D. Moreover, we will present a comparison study of our used multi-objective evolutionary algorithm with another mono-objective evolutionary algorithm.

**Chapter 4: Validation**

* 1. **Introduction**

To evaluate our work for adapting systems’ interfaces using MOEA/D, we conducted a set of experiments based on one large system (HandcraftWomen). Each experiment is repeated 31 times, and the obtained results are subsequently statistically analyzed with the aim to measure the performance of our MOEA/D proposal with various given criteria. In this section, we first present our research questions and then describe and discuss the obtained results.

* 1. **Research Questions**

We assess the performance of our approach by finding out whether it could generate meaningful sequences of rules that improve the adaptivity of interfaces while reducing the number of rules needed. Our validation is conducted by addressing the following research questions outlined below. We also explain how our experiments are designed to address these questions:

**Research Question 1**: To what extent can the MOEA/D help in improving the adaptivity of interfaces in the system?

**Research Question 2**: To what extent can the use of multiple metrics improve the adaptivity of interfaces in the system?

**Research Question 3**: To what extent can the proposed approach minimize the number of needed rules?

**Research Question 4**: How does the proposed MOEA/D perform compared to a mono-objective approach?

To answer Research Question 1, we emphasize on the reason behind using a metaheuristic to solve the problem of generating the best sequence of adaptive rules. To do so, we simulate the growth of our input problem i.e. number of interfaces, and we increase the number of possible metrics to use and then we compare the performance of the exhaustive search against MOEA/D in terms of runtime needed to generate the rules.

To answer Research Question 2, we focus on the importance of using various metrics to achieve better coverage of all the possible interfaces properties that we want to capture. We carry on an experiment that starts with generating rules using only one metric and calculate its code coverage then gradually increment the number of metrics used to investigate its impact on the coverage.

To answer Research Question 3, we compare between two simulations of MOEA/D, in which one uses one objective (maximizing the coverage) and the other adds the second objective (minimizing the number of used rules). We investigate the possibility of empirically achieving the same results with less number of used rules.

To answer Research Question 4, we experiment the benefit of using a multi-objective algorithm by comparing MOEA/D against a mono-objective algorithm in which the two objectives, used in MOEA/D, are being aggregated in one single fitness function.

* 1. **Studied Project**

We applied our approach to evaluate the quality of adaptive user interfaces. To reach this purpose, we use an open-source java projects called *HandicraftWomen*. This project takes place in a Higher Institute of Management in Tunisia as part of a graduation projects. It aims to support handcraft women in their business activities. This project consists on adapting the current technologies to the profile of those handcraft women.

* 1. **Experimental Setting**

The evaluation of the adaptive user interfaces of the studied project should improve the adaptivity of his application and enhance the satisfaction of their users.

* + 1. Subjects:

Our work involved 250 subjects who have different age, motivation, level of education, experience and interest. They are composed from 100 handcraft women and 150 different users who are invited to test the adaptive user interfaces of our studied project.

All the subjects are volunteers and they have several profiles and preferences.

* + 1. Scenario

The subjects were invited to fill a questionnaire that aims to evaluate our user interfaces of our studied project. This questionnaire was divided into two parts:

* User profile: Contains the profile of the suggested user.
* User evaluation: Contains the answer of the user to the demanded questions after testing the project.

The user interface to evaluate depends on the preferences of users. So, the subjects were first asked to fill out the first part of the questionnaire that contains seven  
questions. Then, we collect the profile of all the users. Moreover, we propose for each user the appropriate interface according to his profile. According to (Vargin et al, 2015), the profile of user is composed from five attributes that can take three values (Low, Medium, and High). However, the subjects have different values for the attribute “age”. So, we classify the user according to their age into three interval: the first is in [18,30], the second is in [30,55] and the third is in [55,80], and we accord the values low, medium and high respectively.

After answering the first part, the subjects should test their appropriate interface and evaluate it by answering at eight questions.

The subjects are asked to express their satisfactions after testing their appropriate interface. So, they are invited to select for each question one of the possibilities: *"Yes"*, "*No"*, or *"May be"* (if not sure).Their results of evaluation are reviewed by an expert on the field of adaptation to extract the different problems of the tested interface detected by users.The questionnaire is organized as presented in Annex.

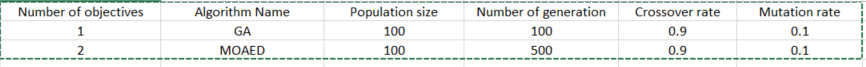
Since we collect the result of evaluation from the questionnaire, we organize it into a survey (trace) to validate the evaluation rules generated from our approach. The table.4.1 describe a part of the organization of our survey contained in the excel file.

**Table.4.1 The organization of the survey**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Range of user | Age | Motivation | Education level | User experience | Interest | Problem | Tested Interface |
| 1 | Low | Medium | Medium | High | High | Regular interface | Interface3 |
| 2 | Meduim | Medium | High | High | High | High guidance | Interface5 |
| 3 | Low | Low | High | Low | Low | Simple interface | Interface1 |
| 4 | Meduim | Medium | Low | Medium | High | Ordered interface | Interface3 |
| 5 | Medium | High | High | High | Medium | Complex Interface | Interface2 |
| 6 | High | Low | High | Low | Medium | Workload | Interface4 |
| 7 | High | Medium | High | Medium | Low | Centered interface | Interface1 |
| 8 | High | Low | High | Low | Low | Low guidance | Interface2 |
| 9 | Low | Medium | Medium | High | High | Well arranged interface | Interface6 |
| 10 | Low | Meduim | High | High | High | Coherent interface | Interface7 |

* + 1. Parameter setting

Parameter setting influences significantly the performance of a search algorithms on a given search problem. It is usually difficult to preemptively set the best tuning setting. For this reason, we perform a set of experiments using several population sizes: 10, 20, 40, 80, 160, and 320 for our 2 objectives. The maximum number of generations used is 100, 200, 400, 8000 and 1600. For each algorithm, to generate an initial population, we start by defining the maximum vector length (maximum number of rules per solution). As a higher number of operations in a solution do not necessarily mean that the results will be better, we empirically determine the best set of setting through the above mentioned trials. Ideally, a small number of rules should be sufficient to provide a good trade-off between the fitness functions. This parameter can be also specified by the user or derived randomly from the sizes of the program and the used operations list. During the creation, the solutions have random sizes inside the allowed range. We use the trial and error method (Walker et al, 2013) in order to obtain a good parameter configuration. Since we are comparing different search algorithms, we classify parameters into common parameters and specific parameters. Table X depicts the important common parameters. For MOEA/D, the neighborhood size is set to 20



* 1. **Research Questions**
     1. **Results for Research Question 1**

During the rules generation process, our approach combines randomly structural metrics with context criteria within logical expressions (union OR; intersection AND) to create rules. In this case, the number n of possible combinations is very large. The rule generation process consists of finding the best combination between m structural metrics and k context criteria. In addition, for three threshold values (Low, medium, High) that each matric/criterion can take, a huge number of rules can be generated. In this context, the number NR of possible combinations that have to be explored is given by: NR = (3nk). Table X shows the following experiment of considering the runtime of the brute force search, in which, all the possible combinations of rules are exhaustively explored, against the use of a meta-heuristic.

Table 1: Runtime in micro-seconds of Brute Force and MOEA/D over a given number of metrics/criteria.

|  |  |  |
| --- | --- | --- |
| Number of metrics/criteria | Brute Force runtime (μs) | MOEA/D runtime (μs) |
| 6 | 830 | 25411 |
| 10 | 60000 | 25948 |
| 15 | 14349007 | 26966 |
| 20 | 3.49E+09 | 27109 |
| 25 | 8.47E+11 | 27982 |
| 30 | 2.06E+14 | 28611 |

As noticed in Table 1, the runtime of Brute Force quickly becomes huge for a higher number of metrics while MOEA/D’s runtime does not significantly increase. We conducted a further experiment in which we increase the population size along with the number of iterations (generations) and we noticed a jump in the runtime of MOEA/D. We empirically validated that MOEA/D generated near optimal results in a relatively reasonable runtime. Consequently, to ensure the scalability of our solution, we considered the rule generation process as a combinatorial optimization problem. As any solution must satisfy two criteria (complexity and coverage).

|  |  |  |
| --- | --- | --- |
| population size | generation size | MOEA/D runtime (μs) |
| 10 | 100 | 2421 |
| 100 | 1000 | 5625 |
| 500 | 10000 | 7688 |
| 1000 | 100000 | 97226 |
| 1100 | 1000000 | 116535 |
| 1500 | 10000000 | 206049 |

To this end, we propose an adaptation of the MOEA/D for the problem of rules generation and the results are described in the next sub-section.

* + 1. **Results for Research Question 2**

Sometimes it is important to normalize fitness values to a given range. This might for instance be the case if outliers exist that are very far from the other fitness values. If a fitness proportional selection operator is applied, these outliers could easily dominate all others, which is not desired. To cope with this issue, we normalize the fitness values. Figure 1 summarizes the results of median values of the non-normalized and normalized fitness functions of the solution belonging the Pareto front over 31 independent simulation runs.

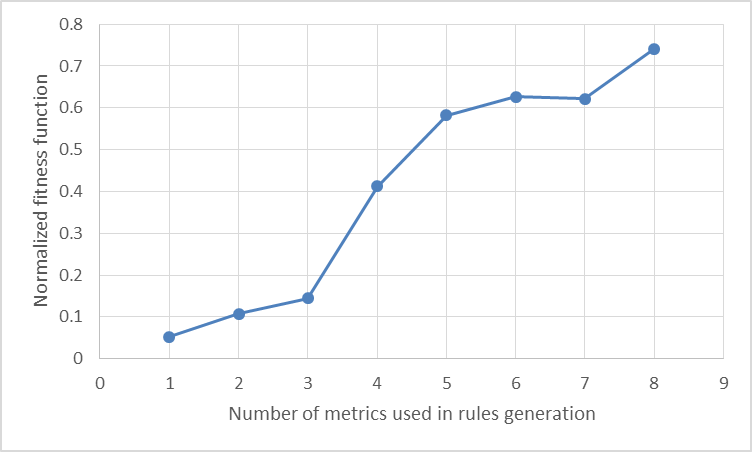
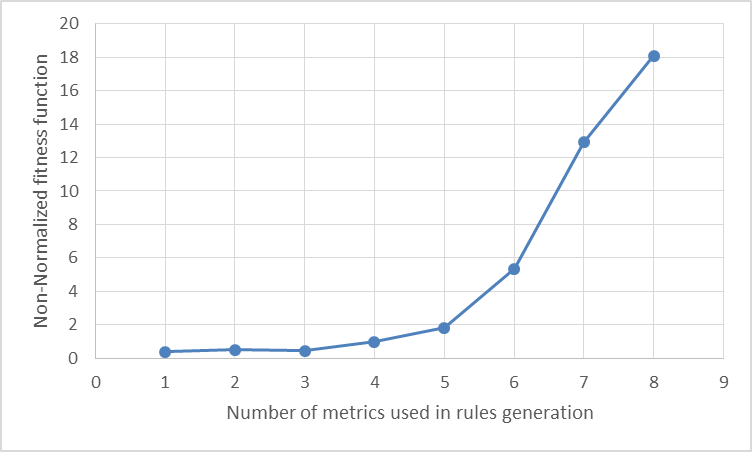


Figure 1 :

During the normalization process, the best fitness value receives a predefined value of 1.0, while the worst one gets 0.0. All other fitness value components are then mapped into the given range.

To study the impact of using multiple structural metrics and context criteria, we run the process of generating rules using initially one single metric, then we measure average coverage score of the execution of the generated rules on our system. Then we keep incrementing the number of used rules during the generation and we report the evolution of their relative coverage scores over the GUIs in the following Figure X

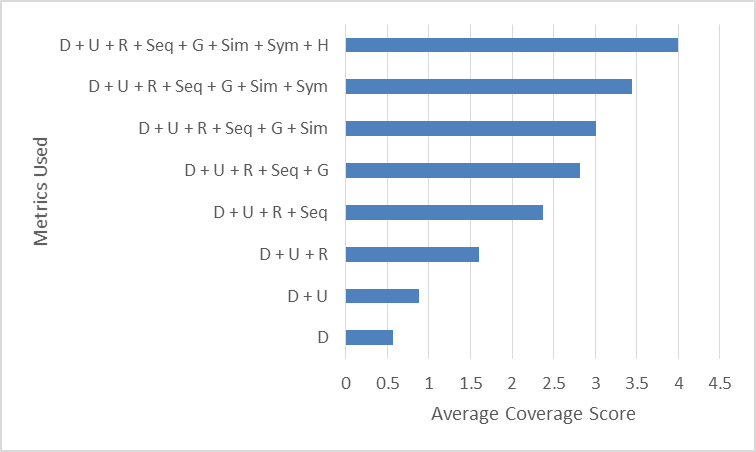


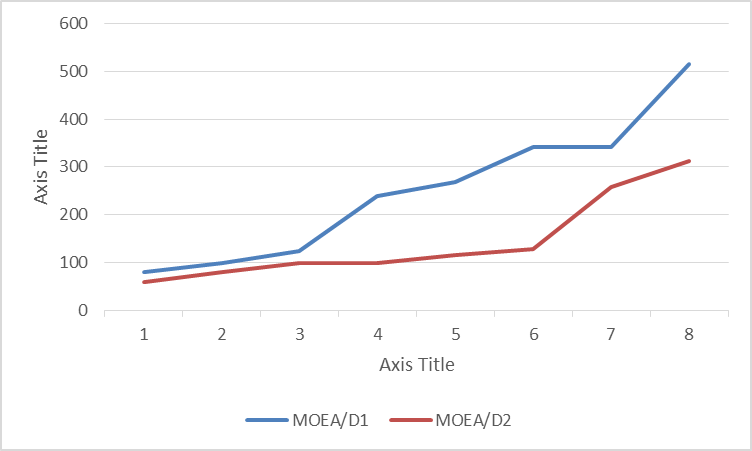
Figure 2 : …

The results of Figure X are based on the gradual consideration of all the 8 metrics for the multi-objective algorithm MOEA/D. Since the metrics used are independent so the order in which we added is not important and have no impact on the results which have been showing that, by increasing the number of used metrics, the median fitness function of the last generation of the algorithm is also increasing. Thus, the use of various metrics and criteria help the algorithm in better exploring the search space and generate near optimal solutions. Figure Y gives the detailed measures of all metrics overall system GUIs.

|  |  |
| --- | --- |
| It is noticeable that some interfaces are metric-independent due to measures inapplicability. For example, let’s consider *Cadre.java*, the class structure cannot be measured by Grouping and Symmetry, that’s why their relative values in the graph is absent. This explains the need of deploying various metrics to allow the rules to capture more system properties and so achieve better coverage. | Figure 3:… |

* + 1. **Results for Research Question 3**

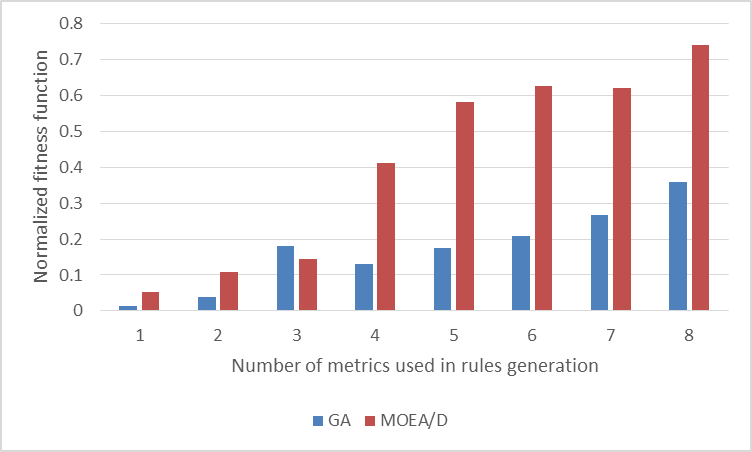
To test the effectiveness of the minimizing the number of rules, we consider two instances of MOEA/D in which, the first instance called MOEA/D1 runs without taking into account the second objective i.e. minimizing the number of rules while the second instance MOEA/D deploys it. Figure X summarizes the results of median values of the number of generated rules over 31 independent simulation runs after applying the proposed rules by the best solution selected using the knee-point strategy



We notice that the second instance of MOEA/D was able to generate less number of rules while maintaining good coverage. This can be explained by the fact that increasing the number of metrics allow the search heuristic to reach more rules combinations and so create more rules. Since the use of various metrics enhance the coverage quality, any mono-objective heuristic will eventually end up deploying as many rules as possible as it leads to better fitness. Since the use of many rules is not recommended as it increases the solution complexity, it is necessary for a good solution to find the tradeoff between maximizing the coverage while maintaining a relatively low number of rules. Since these two objectives are conflicting, the mono-objective MOEA/D is unable to handle such conflict as it aggregates the two fitness functions into one. While MOEA/D treats each fitness function separately so it is capable of generating high quality solutions while lowering the needed number of rules to execute.

* + 1. **Results for Research Question 4**

Figure X summarizes the results of median values of the structural metrics over 31 independent simulation runs after applying the proposed rules by the best solution selected using the knee-point strategy. The results of Figure X are based on the gradual consideration of all the 8 metrics for the GA in which the two fitness functions i.e. maximizing the coverage and minimizing the number of rules, have been aggregated into one objective, while for MOEA/D, these two objectives have been considered separately.



As described in Figure X, we found that MOEA/D algorithm provides better structural coverage over GA when the number of used metrics increases. In order for the GA to generate better solutions, it requires more metrics to be deployed per rules and this will automatically increase the number of generated rules. The structural improvement scores of multi-objective and mono-objective algorithms are relatively different especially when the number of metrics used increases. This is an interesting result confirming that our algorithm can find very good compromises between multiple objectives that are dissimilar and it outperforms those that are produced by existing approaches using only mono-objective or exhaustive search (for a large scale system). We believe that improving metrics coverage overs various GUIs is a difficult and very important objective to reach. We consider that out algorithm’s performance is a very promising since the main goal of this work is to be able to find the best coverage rules for any given set of metrics and we have shown that it is feasible over 8 different input metrics.

* 1. **Conclusion**

(…)

**Annex: Questionnaire**

**Part1 : User Profile**

1. Name : ………………….
2. Age :

* Under [18, 30]
* Under [30,55]
* Under [55,80]

1. Motivation :

* Low
* Medium
* High

1. Experience :

* Low
* Medium
* High

1. Education level :

* Low
* Medium
* High

1. Interest :

* Low
* Medium
* High

**Part2 : Evaluation**

1. Is the components of the interface are organized on a regular basis? Is the spacing between the components is equal?

* Yes
* No
* Maybe

1. Is the grouping of components is semantically?

* Yes
* No
* Maybe

1. What do you think of the sequencing interface? Is that the components are organized according to their importance?

* Yes
* No
* Maybe

1. What do you think of the simplicity of the interface?

* Yes
* No
* Maybe

1. Is the interface components are centered in the screen?

* Yes
* No
* Maybe

1. What do you think of the density of the interface ? Does the interface is workload?

* Yes
* No
* Maybe

1. If we divide the interface into two parts vertically, is that the number of components that are in the first portion is equal to those of the second part?

* Yes
* No
* Maybe

1. Does the components of the interface are arranged equally in all four quadrants of the screen?

* Yes
* No
* Maybe