

DIACRITICS OF ARABIC NATURAL LANGUAGE PROCESSING AND
ITS QUALITY ASSESSMENT

ABDULRAHMAN AHMED KHUDHUR

A thesis submitted in partial fulfillment of the requirements for the award of Master
of Computer Science (Software Engineering)

The Department of Software Engineering
Faculty of Computer Science and Information Technology
University Tun Hussein Onn Malaysia

JANUARY 2014

ABSTRACT

Arabic is a language unique to the intended pronunciation of the word written. It cannot be determined exactly according to the orthographic level of representation. The development of a different word formation gives a different meaning. To translate the words into English with the addition of this configuration in a single word, this research looks at using an equation for the formation of the translation after that to see the performance and accuracy of the system by using metrics for ease of use. This research is looking at morphological model of Arabic language and which will then uses the equation to put diacritics according to Arabic grammatical rules. Based on this research a system was developed. The input of the system is Arabic word. The system used morphological Arabic natural language processing and translation Arabic word into English. The output of the system will show percentage of translated words successfully with high precision. The result shows that the Quran translation is translated using 11 words; 9 with high accuracy the result 69% and literature language used 7 arabic words; 4 words from these words that have been translated with high accuracy, which results in 31%, and while every other input was between the successes rates of the program compiled by 100%. Furthermore, quality assessment is performed to calculate efficient and effective usability metrics based on the ANLP developed. Based on the result, the system can be used as a translator from Arabic language to English.

ABSTRAK

Bahasa Arab merupakan bahasa yang unik samada untuk sebutan ataupun perkataan bertulis. Perwakilan untuk setiap perkataan atau ayat yang digunakan tidak dapat ditentukan dengan tepat mengikut tahap ortografik. Pembinaan dan pembentukan ayat yang diolah akan memberi makna yang berbeza. Penterjemahan perkataan ke dalam Bahasa Inggeris adalah dengan menambahkan konfigurasi di dalam satu perkataan. Berdasarkan penyelidikan, ianya memfokuskan penggunaan persamaan bagi pembentukkan terjemahan. Justeru itu, ianya memperlihatkan prestasi dan ketepatan sistem dengan menggunakan metrik yang dapat memudahkan pengguna. Projek ini adalah untuk membina dan membangunkan model morfologi Bahasa Arab dan seterusnya menggunakan persamaan dengan meletakkan tanda diakritik mengikut syarat tatabahasa Bahasa Arab. Oleh itu, bagi setiap perkataan dengan tanda diakritikal akan secara terus diterjemahkan dari Bahasa Arab ke Bahasa Inggeris yang mana kemudiannya akan menggunakan metrik kebolegunaan. Justeru itu, perkataan dari Al-Quran dan sastera arab diambil dan diaplikasikan di dalam program ini dengan proses membentuk dan menterjemah. Oleh yang demikian, paparan peratusan perkataan yang berjaya diterjemahkan dengan ketepatan yang tinggi akan ditunjukkan. Kesimpulannya, penterjemahan Al-Quran dengan hasil 69% daripada mana-mana 11 perkataan; 9 dengan ketepatan yang tinggi dan kesusasteraan bahasa yang menggunakan Bahasa Arab 7 perkataan; 4 perkataan telah diterjemahkan dengan ketepatan yang tinggi, iaitu 31%, dan 100% bagi setiap kadar kejayaan program yang disusun.

CONTENTS

THESIS STATUS CONFIRMATION	I
TITLE	II
DECLARATIONS	III
DEDICATION	IV
ACKNOWLEDGEMENT	V
ABSTRACT	VI
ABSTRAK	VII
CONTENTS	VIII
LIST OF TABLES	XI
LIST OF FIGURES	XII
LIST OF SYMBOLS AND ABBREVIATIONS	XIII
LIST OF APPENDICES	XIV
Chapter 1 INTRODUCTION	1
1.1 Background Study	1
1.2 Purpose Of Project	2
1.3 Problem Statement	3
1.4 Objectives	3
1.5 Scope Of Project	4
1.6 Thesis Outline	4
Chapter 2 LITERATURE REVIEW	5
2.1 Introduction	5
2.2 Overview Of Natural Language Processing (NLP)	6
2.3 Challenges In Arabic Natural Language Processing (ANLP)	6
2.4 History of Arabic Language	8
2.5 Grammar of Arabic Language	9

2.5.1	They are listed	13
2.6	Overview of Usability Metric	13
2.7	QUIM: Quality in Use Integrated Measurement	15
2.7.1	QUIM: a Roadmap for a Consolidated Model	17
2.7.2	10 Usability Factors of QUIM	18
2.7.3	Metrics	20
2.7.4	Measurable Criteria	21
2.8	Related Work	25
2.9	Summary	29
Chapter 3	RESEARCH METHODOLOGY	30
3.1	Introduction	30
3.2	The Proposed Methodology for ANLP	31
3.2.1	Word in Arabic Language	32
3.2.2	Morphological Language Model	34
3.2.3	Arabic Morphological Diacritization	34
3.2.4	Morphological and Syntactical Diacritization	35
3.2.5	Arabic Syntactic Diacritics	36
3.2.6	Output Arabic Word Diacritics	36
3.3	Usability Metric Using (QUIM) Techniques	37
3.4	Summary	38
Chapter 4	THE DESIGN AND DEVELOPMENT OF (ANLP) ALGORITHM	39
4.1	Introduction	39
4.2	Word in Program	39
4.3	The Algorithm of (ANLP)	40
4.3.1	The Implementation of (ANLP)	41
4.4	Algorithm for ANLP System	44
4.4.1	The Old Algorithm:	45
4.4.2	The New Algorithms Use in Project	47
4.4.3	Comparison Between Two Algorithms	48
4.5	The Development of ANLP System	48
4.6	Summary	49

Chapter 5	RESULTS AND DISCUSSION	52
5.1	Introduction	52
5.2	Experimental Result for ANLP	53
5.3	Usability Metric Analysis	56
5.4	Summary	59
Chapter 6	CONCLUSION AND RECOMMENDATION	60
6.1	Achievement of Objectives	60
6.2	Contribution	61
6.3	Future Work	59
6.4	Conclusion	62
References		63
Appendix		66
Vitae		

LIST OF TABLES

2.1	Arabic diacritics set	7
2.2	Essential Usability metric	13
2.3	Usability criteria in the QUIM model	20
2.4	Relations between factors and criteria in QUIM	20
2.5	Examples of calculable metrics in QUIM	21
2.6	Current researches in Arabic natural language processing are they translate to English	27
3.1	Arabic POS tags set	31
4.1	same of characters mapping table	40
4.2	the different steps between the old and new algorithm	47
5.1	Distribution word domains	51

LIST OF FIGURES

2.1	Structural Analysis of ANLP	9
2.2	structural analysis ANLP (Chomsky)	10
2.3	equation of chomsky	11
2.4	Tree simplify process	12
2.5	Quim Framework	18
2.6	Example QUIM Components Relationship	24
3.1	Chart Shows The Steps The Project Work	29
3.2	Chart Shows Separate The Word To Letter	30
3.3	Morphological And Syntactical	32
3.4	The Arabic– Morphological And Syntactical Diacritization	33
3.5	Disambiguation Lattice For Morphological Disambiguation, Syntactic Diacritization	33
3.6	The Architecture Of Arabic Diacritics Statistically Disambiguating Arabic Text	35
4.1	Arabic Morphological Analysis As An Intermediate Is Arabic Morphological Analysis As An Intermediate Ambiguous Language.	38
4.2	Dictionary Structure	40
4.3	Dictionary Searching Criterion	42
4.4	Old Algorithm by Mohammad Ahmed Sayed (2009).	43
4.5	Algorithm Use In ANLP System	45
4.6	Path Algorithm For ANLP	47
4.7	Finely Search Algorithms	48
4.8	New Interface for the system	49
5.1	The Words Used And The Percentage Of Right Wrong In It	51
5.2	Coupling Ratio Between Results	52
5.3	The Result For ANLP Software	53
5.4	The Result For Usability Metric “Efficiency”	55
5.5	The Result For Usability Metric “Effectiveness”	56

LIST OF SYMBOLS AND ABBREVIATIONS

- D - Is the “Dictionary” array.
- F - Is the “First Letter Word Group” array.
- S - Is the “Sum Word Group” array.
- W- Is the “Word Diction” array.
- FD - is a function that applies the binary search technique on the “Word Diction” array to find a un diacriticsd word.
- FA- is a ASCII number use in algorithm only 3 (-10,-11,-13).
- u - Is the location of the un diacriticsd word in the “Word Diction” array.
- n - Is the number of letters in the un diacriticsd word.
- i - Is the last letter ID.
- l - Is the summation of all letters IDs in the un diacriticsd word.
- uw -is the un diacriticsd word string.
- dw -is the diacriticsd word string.

LIST OF APPENDICES

A	Table for Related Work Result	64
B	Same Result For Related Work	65
C	Same Of Characters Mapping Table	66

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

The Arabic language is both challenging and interesting. It is interesting due to its history, the strategic importance of its people and the region they occupy, as well as its cultural and literary heritage. It is also a challenging language because of its complex linguistic structure. Historically, classical Arabic has remained unchanged, clear and functional for more than fifteen centuries (Attia, 2008). Culturally, the Arabic language is closely associated with Islam and literature. Strategically, it is the native language of more than 330 million speakers living in an important region with huge oil reserves (control the world economy) and home to the sacred sites of the world's three Abrahamic religions. It is also the language in which 1.4 billion Muslims perform their prayers five times daily. Linguistically, it is characterized by a complex diglossic situation (Abdel, 2009). The Classical Arabic represents the language spoken by the Arabs more than fourteen centuries ago, while Modern Standard Arabic is an evolving variety of Arabic with constant borrowings and innovations proving that Arabic reinvents itself to meet the changing needs of its speakers. At the regional level, there are as many Arab dialects as there are members of the Arab league. The diglossic nature of the Arabic language is discussed (Khaled Shaalan, 2010).

Therefore, the Arabic natural processing language applications must deal with several complex problems pertinent to the nature and structure of the Arabic

language. For example: (علم science, عَلِمَ flag, عَلَّمَ taught, عَلَّمَ knew). Arabic is written from right to left. Like Chinese, Japanese, and Korean, there is no capital letter in Arabic. In addition, Arabic letters change shape according to their position in the word. Modern Standard Arabic does not have orthographic representation of short letters which require a high degree of homograph resolution and word sense disambiguation. Like Italian, Spanish, Chinese, and Japanese, Arabic is a pro-drop language, that is, it allows subject pronouns to drop (Farghaly, 1982). A language that is subject to recoverability of deletion (Chomsky, 1965), as a natural language, Arabic has much in common with other languages such as English.

1.2 Purpose of the Study

The purpose of this study is to describe the solutions that would solve problems related to Arabic natural language processing (ANLP, hereafter). Writing Arabic text is typically made without any diacritics, which may generate some common spelling mistakes, such as ((أ-أ), (ة-هـ) (ي-ى)), due to the highly derivative and inflective nature of Arabic. Thus, it is very difficult to produce a complete compilation of vocabulary that covers all (or even most of) the Arabic general words, and hence, the morphological analyser is used to solve the problem of coverage instead of using a dictionary, as well as to discover defects, remove disambiguation, and validate words. After that, the word would be translated into English, and usability metrics were used to look into the performance and the accuracy of work and to increase speed. About two third of Arabic text words have syntactically dependent case-ending, which invoke the need of a syntax analyser, which is a complex problem.

1.3 Problem statement

Arabic is a unique language based on the intended pronunciation of a written word that cannot be completely determined by its standard orthographic representation. By putting different diacritics words, it can give different meaning. For example:

- عِلْم science
- عَلَم flag
- عَلَّمَ taught
- عَلَّمَ knew

To translate Arabic words into English with the addition of these diacritics per word, this research used equations for both diacritics and translations to look into the performance and accuracy of the system using quality assessment.

1.4 Objectives

The objectives of this research are:

- i. To design and develop a morphological ANLP.
- ii. To compare with other ALPS algorithms in order to validate the design algorithm of ANLP.
- iii. To validate the ANLP developed using usability metrics (the efficient and the effective).

1.5 Scope of the Project

This project focused on natural language processing of Arabic, but with diacritical marks (Fatha, Kasra, Damma) using the ASCII code for the Arabic language. Measures of usability were focused on the efficiency and effectiveness. As for the words that were chosen, 11 words were from the Quran, 7 words from the Arabic language, and every word containing 3 letters from the literature.

1.6 Thesis Outline

The thesis consists of five chapters. Chapter 1 is an overview of the project, and presents the main objectives of the project. It consists of the scope of work covered and methodology of the project.

Chapter 2 illustrates the ANLP concept from the point of literature review. It also gives a brief explanation on the general information about the rules of Arabic Language in this project.

Chapter 3 discusses the suitable methodology to satisfy the objectives of this project. This project used work equation that converted selected words to the case of Diacritical, and then translated into English with usability metric.

Chapter 4 is about design algorithms that were designed to work the programme. The design of the three mathematical equations was related to the ASCII code.

Chapter 5 discusses the analysis obtained from the experiment and laboratory testing from the previous chapter. The final part of this chapter explains the results obtained.

Chapter 6 concludes the thesis based on the results and discussion obtained from this project, and suggests recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Arabic is a difficult language with a grammatical system that is different than English. There is a large potential for errors in interference when Arab learners produce written or spoken English. Arabic word has a three consonant root as its basis. All words in the parts of speech are formed by combining the three-root consonants with fixed vowel patterns and, sometimes, an affix. Arabic learners may be confused by the lack of patterns in English that would allow them to distinguish nouns from verbs or adjectives (Paul, 2012).

In term of alphabet: Arabic has 28 consonants (English 24) and eight vowels/diphthongs (English 22). Short vowels are unimportant in Arabic, and indeed do not appear in writing. Texts are read from right to left and written in a cursive script. No distinction is made between upper and lower case, and the rules for punctuation are looser than in English.

English has about three times as many vowel sounds as Arabic, so it is inevitable that beginners will fail to distinguish between some of the words they hear, such as ship / sheep or bad / bed, and will have difficulties saying such words correctly.

Problems in pronouncing consonants include the inability to produce the sounds in words such as ‘this’ and ‘thin’, the swapping of /b/ and /p/ at the beginning of words, and the substitution of /f/ for /v/. Consonant clusters, such as in words split, threw or lengths, also cause problems and often result in the speaker adding an extra vowel: spilit, ithrew or lengthes.

In Arabic, word stress is regular. It is common, therefore, for Arab learners to have difficulties with the seemingly random nature of English stress patterns. For example, the word ‘yesterday’ is stressed on the first syllable and ‘tomorrow’ on the second (Husni, 2008).

2.2 Overview of Natural Language Processing (NLP)

Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things. NLP researchers aim to gather knowledge on how human beings understand and use language so that suitable tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform desired tasks (Gobinda, 2005).

2.3 Challenges in Arabic Natural Language Processing (ANLP)

There are many challenges for learning the Arabic language; the most prominent are to understand the characters and the diacritical of the Arabic language. Hence, in order to understand the rules of the Arabic language very well, one needs to know knowledge of the details in the Arabic language (Ali, 2009).

Table 2.1: Arabic diacritics set

Diacritic's type	Diacritic	Example on a letter	Pronunciation
Short vowel	Fatha	ﺃ	/b//a/
	Kasra	ﺃِ	/b//i/
	Damma	ﺃُ	/b//u/
Doubled case ending (Tanween)	Tanween Fatha	ﺃَﺃَ	/b//an/
	Tanween Kasra	ﺃِﺃِ	/b//in/
	Tanween Damma	ﺃُﺃُ	/b//un/
Syllabification marks	Sukuun	ﺃْ	No vowel: /b/
	Shadda	ﺃّ	Consonant doubling: /b//b/

The diacritics shown in Table 2.1 are a core group of Arabic diacritics, but there is another set of forms that may look like combinations between the pairs of short vowel intensity such as ‘b’ (pronounced as / b / //b//a/), and severity, such as the pairs in Tanween (pronounced as / b / / b / / UN /). In fact, as the Arab has rich vocabulary on its full form words, scattering the resulting data was easier when it is considered parts of words (Clemet) separately because conformation in Arab is systematic and very rich.

Thus, reliable Arab morphological analysis is crucial in forming the Arabic text and this is likely the case for audio versions of the text input (Attia, 2008; Mark, 2012). While this methodology prefers excellent coverage of the language, the drawback of it is that the search space for the correct configuration using the word components is much larger than the original area to search for the full form of words.

This requires more space to find the largest volume of training data, which is expensive and takes a long time to build and validate (Faiza, 2008). Moreover, this approach requires longer time due to address the large size of the search lattice built.

2.4 History of the Arabic language

Arabic is a Semitic language (Ibrahim, 2000), such as Syria, Aramaic, and Hebrew, which are the languages of the Arabs the population of the Arabian Peninsula between the Persian Gulf and the Red Sea. Arabic contains 28 written characters and is written from right to left - after many languages of the world - and from the bottom to the top of the page. It is considered one of the most widely used languages of the world because there are more than 250 million inhabitants in Arab. Besides, Arabic is the official language of many countries in the Arab world, including Egypt, Algeria, Iraq, Jordan, Lebanon, Saudi Arabia, Syria, and Kuwait. In the middle of the twentieth century, a number of Arab states play an important role in the international relations and as a result, it has become a major language in the Arab as far as business and politics are concerned. Besides, Arabic has been adopted as one of the official languages in the United Nation.

In addition, grammar in Arabic is derived from the expression of things. Furthermore, Arabic is the language of the Quran. The Arabians are found vastly in the Arab states, Israel, the Arab world has a huge number of speakers; 350 million (with Natakayaa as a second language) (Ibrahim, 2000).

Arabic grammar is related to the origins of the composition and rules of a sentence. The aim is to determine the composition of sentences, words, and places, where the function also determines the properties acquired by the word of that position, and there are three grammatical properties: 'Kalaptda' (effective), 'Mufaulah' (grammatical sentences), 'Kalkadim' (delays, express, and construction).

Arabic languages that are amended as derivative (distracted) source (an act past) and models (weight) are to derive certain luminosity close to the source. The exchange is to analyse the words in terms of installation and type of call, for example, the names and words are divided into deeds, returned to the roots, and the weights are measured.

2.5 Grammar Structural Speaking for Arabic language

Arabic language is one of the most difficult languages in the world because they rely on a set of rules and the basic grammar. When speak Arabic the sentence must be based on structural such as the same language (names, verbs, adjectives, etc.) to be corrected and clear sentence when a pronunciation:

- i. Noun + verb + Character
- ii. Verb + Actor + Object
- iii. Verb + Actor + Time-Loc
- iv. Type of Plural
 - Masculine
 - Feminine
 - Cracking

- i. Noun + verb + character:

In the first rule shows that speech in the Arabic language is built on the basis of the noun and verb and character. As example in Figure 2.1, the sentence all linked to each other if tried to remove one of them to become the sentence is clear and understandable.

Example: Ismail plays with a cat

Ismail	Plays	With a cat
Noun (N)	Verb (V)	Character (C)
N	V	C
N + V + C		
Ismail plays with a cat		

Figure 2.1: Structural Analysis of ANLP

Figure 2.1 portrays the rules and did the noun and verb and character and how the adoption of the sentence on the consistency between them, where if one of them to delete the sentence is understandable.

ii. Verb + Actor + Object:

The beginning of the process of analysis, = c =, is divided into two components: the actual compound (verb with the Actor), symbolized by (MF), and the nominal compound (object), symbolized by the B (M S). Then, dissect (MF) to do (P) and actor (m), and then, (P) to do verb (effect) and time (g). Lastly, nominal compound (PG) is dissected to (a) Definition Tool (define). So the final result of the analysis is the status of bilateral sports arranged and coordinated to form a sentence: 'the student opened the door'.

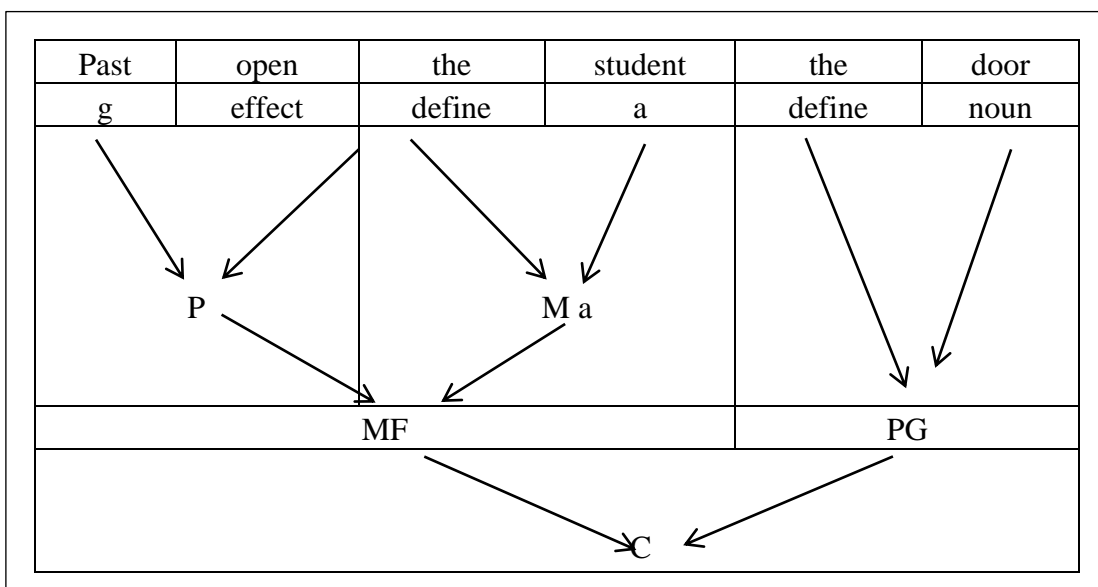


Figure 2.2: Structural Analysis of ANLP (Chomsky, 2005)

Figure 2.2 portrays the reconstruction rules: a set of rules that branches the sentence "as an initial" symbol "Chomsky tried to explain the analytical levels of inters beginning with the analytical level, which divides (c) to the following equation:

iii. Verb + Actor + Time-Loc

Example: my son went to school in the morning

This one Arabic grammar when speaking that depend on time and place, as in the example above where see the presence of a time (morning) and place (School) When you delete one of these words be sentence is understandable.

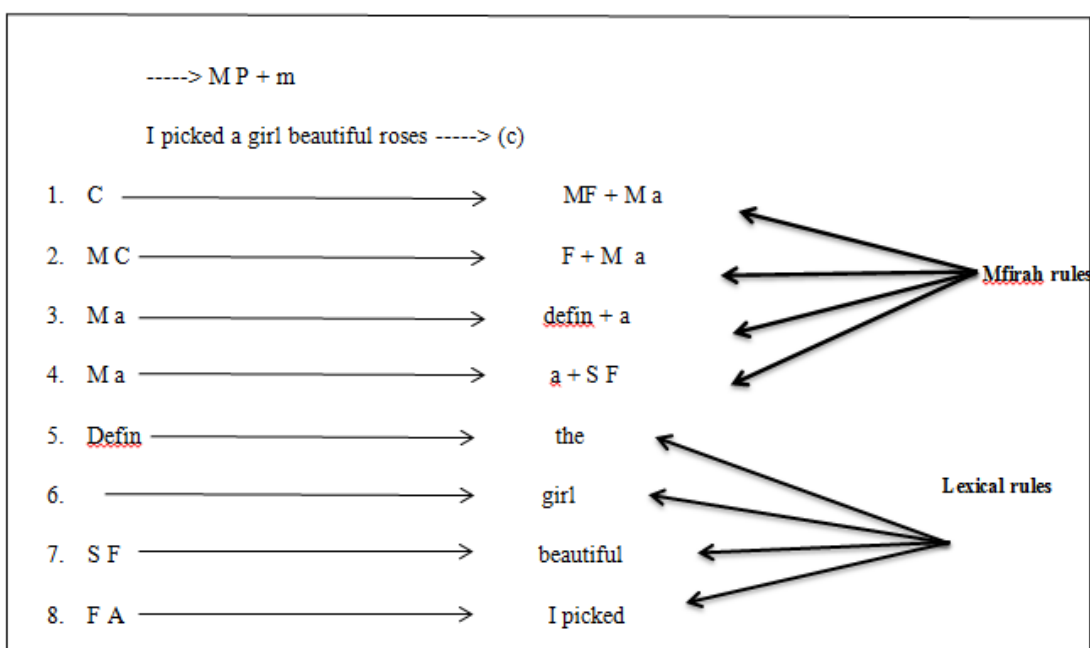


Figure 2.3: The Equation of Chomsky (Chomsky, 2005)

Figure 2.3 is the equation of Chomsky that gives two rules 'Mfirah' as the branching levels of linguistic, but the rules of lexical provide the levels of language vocabulary, and after it ends, an analyst from the application of the rules Mfirah, as prescribed in the application of the rules of lexical, generates the chains of the language. The goal is to present the evolution generative of the sentence according to the rules described above, for example, rule No. 3, the natural result of a series of language is changed to (defin + a), namely:

Definition + Name

Chomsky (2005) also found discharges in the form of a tree is to simplify the process, and this is what a Dingle composition is called, which aims to draw the hidden structure of a sentence, as shown in Figure 2.4.

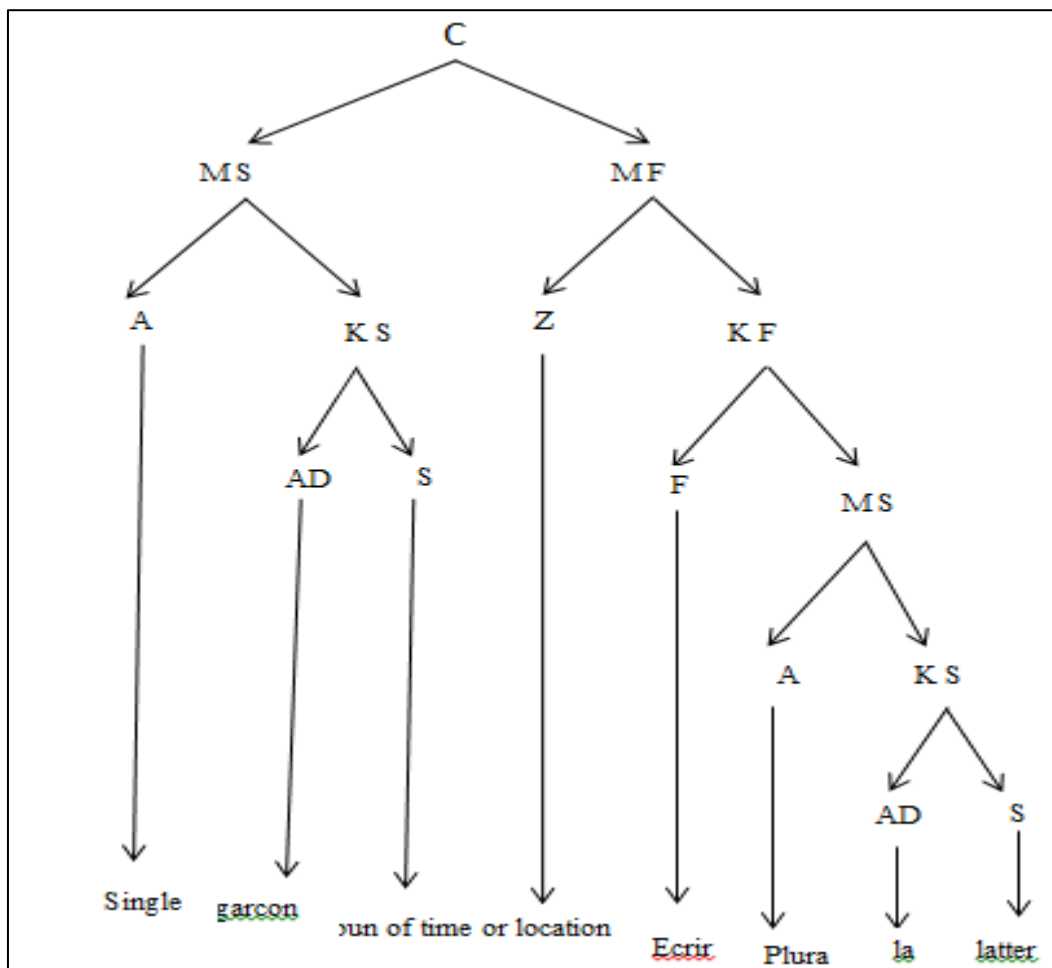


Figure 2.4: Tree Simplify Process (Chomsky, 2005)

Figure 2.4 simplifies the process of the structural transformation. The first is the ability of any deep structure, and the second surface highlights apparent pronunciation, and this goes through transformational rules.

2.5.1 They are listed as follows:

- i. There are general rules for transformative work in developing rules.
- ii. Special transformational rules operate in one entrance, and the development of these rules are divided into two:
 - Transformational JAAZIMA rules include rules for passive and interrogative Articles and Imperatives.
 - Transformative mandatory rules are rules that include accessories, time, and boundaries.

JAAZIMA is an expression in the Arabic language for the coordination of speech and discrimination some meaning for others. It is a case of expression, as well as for lifting, monuments, traction and specializes in the present tense; one is not to reveal the names or letters in past tense. The sign of the original assertion is asleep, and the vowel is deleted if present tense is in use (Hefny, 2008).

Chomsky (2005) asserts that mathematical formulation in the rules of grammar through mathematical equations helps in computing because computing greatly facilitates language through mathematical models.

2.6 Overview of Usability Metric

Nowadays, it is very common to apply metrics in the development of systems. Metrics are used as mechanisms for evaluating the quality of the product in terms of efficiency, portability, usability, maintainability, reliability, and functionality (ISO, 2001; Landauer, 1995). Hence, software development and maintenance projects can be understood, controlled, supervised, guessed, and predicted using metrics (Briand, 1996) and, in many cases, the difference between two systems can be something so simple and so important like applying quality.

Usability metrics are important in order to produce a product that is easy to use. Usability can help to make a system nearer to the final user. If a system is usable, it is easy to learn how to use it productively (Constantine, 1999).

A metric is a way of measuring or evaluating a particular phenomenon or thing. One can say something is longer, taller, or faster because it's able to measure or quantify some attributes of it, such as distance, height, or speed. The process requires agreement on how to measure these things, as well as a consistent and reliable way of doing it. An inch is the same length regardless of who is measuring it, and a second lasts for the same amount of time no matter what the time-keeping device is. Standards for such measures are defined by a society as a whole and are based on standard definitions of each measure.

Metrics exist in many areas of our lives. Familiar with many metrics, such as time, distance, weight, height, speed, temperature, volume, and so on. Every industry, activity, and culture has its own set of metrics. For example, the auto industry is interested in the horsepower of a car, its gas mileage, and the cost of the materials. The computer industry is concerned with the processor speed, memory size, and power requirements. Measuring the user experience involves collecting, analysing, and presenting usability metrics (Albert, 2008).

The usability metrics are used as follows:

- i. Compare usability of two products
- ii. Classify the magnitude of a problem
- iii. Make predictions about the actual use of the product
- iv. Provide management with facts and figures

The essential usability metrics include Completion Rates, Usability Problems, Task Time, task Level Satisfaction, Test Level Satisfaction, Errors and Expectation, Page Views/Clicks, Conversion, and Single Usability Metric, as discussed in Table 2.2.

Table 2.2: Essential Usability Metrics (Seffah, 2006)

Essential Usability Metrics	
<p>Expectation: Users have an expectation about how difficult a task should be based on subtle cues in the task-scenario. Asking users how difficult they expect a task to be and comparing it to actual task difficulty ratings (from the same or different users) can be useful in diagnosing problem areas.</p>	<p>Page Views/Clicks: For websites and web-applications, these fundamental tracking metrics might be the only thing you have access to without conducting your own studies. Clicks have been shown to correlate highly with time-on-task which is probably a better measure of efficiency. The first click can be highly indicative of a task success or failure.</p>
<p>Test Level Satisfaction: At the conclusion of the usability test, have participants answered a few questions about their impression of the overall ease of use. For general software, hardware and mobile devices consider the System Usability Scale (SUS), for websites use the SUPR-Q.</p>	<p>Errors: Record any unintended action, slip, mistake or omission a user makes while attempting a task. Record each instance of an error along with a description. For example, "user entered last name in the first name field". The later add severity ratings to errors or classify them into categories. Errors provide excellent diagnostic information and, if possible, should be mapped to UI problems. They are somewhat consuming to collect as they usually require a moderator or someone to review recordings (although my friends at Web non-graphic have found a way to automate the collection).</p>
<p>Completion Rates: Often called the fundamental usability metric, or the gateway metric, completion rates are a simple measure of usability. It is typically recorded as binary metric (1=Task Success and 0=Task failure). If users cannot accomplish their goals, not much else matters.</p>	<p>Usability Problems (UI Problems) encountered (with or without severity ratings): Describe the problem and note both how many and which users encountered it. Knowing the probability a user will encounter a problem at each phase of development can become a key metric for measuring usability activity impact and ROI.</p>
<p>Task Time: Total task duration is the de facto measure of efficiency and productivity. Record how long it takes a user to complete a task in seconds and or minutes. Start task times when users finish reading task scenarios and end the time when users have finished all actions (including reviewing).</p>	<p>Task Level Satisfaction: After users attempt a task, have they answered a few or just a single question about how difficult the task was. Task level satisfaction metrics will immediately flag a difficult task, especially when compared to a database of other tasks.</p>

<p>Conversion: Measuring whether users can sign-up or purchase a product is a measure of effectiveness. Conversion rates are a special kind of completion rate and are the essential metric in e Commerce. Conversion rates are also binary measures (1=converted, 0=not converted) and can be captured at all phases of the sales process from landing page, registration, checkout and purchase. It is often the combination of usability problems, errors and time that lead to lower conversion rates in shopping carts.</p>	<p>Single Usability Metric (SUM): There are times when it is easier to describe the usability of a system or task by combining metrics into a single score, for example, when comparing competing products or reporting on corporate dashboards. SUM is a standardized average of measures of effectiveness, efficiency of satisfaction and is typically composed of 3 metrics: completion rates, task-level satisfaction and task time.</p>
---	---

Table 2.2 discussed measurements to serve as a general index of quality. These standards to make the various factors that make for the accuracy of the results. The purpose of the Table 2.2 is to develop design standards that are easy to use, and correct in theory, and can be clear and transparent and linked to the principles of good design.

In this project used (Completion Rates) and (Usability Problems) to find out the problems faced by the users in each stage of the system and the impact of that activity and return on investment.

2.7 QUIM: Quality in Use Integrated Measurement

QUIM is a repository of 10 factors, 26 criteria, and 128 metrics for assessing usability in the use of software systems. Most of the existing usability models/standards may be seen as specific instances of the QUIM model. The underlying practical motivation for the development of QUIM is to make usability measurement practices and knowledge easily accessible to software developers unfamiliar with usability concepts.

2.7.1 QUIM: A Roadmap for a Consolidated Model

The proposed QUIM is a consolidated model that can be used for usability measurement. Similar to the existing software engineering models and most usability and measurement, QUIM is hierarchical in that it decomposes usability into factors, then into criteria, and finally into specific metrics.

The main application for QUIM at this time is to provide a consistent framework and repository for usability factors, criteria, and metrics for educational and research purposes. After empirical validation of the hierarchical relationships implied by QUIM, it may be possible to create an application-independent ontology about usability measurement (Jarrar, 2003). By instantiating such an ontology, it may be possible to create a knowledge base that can be used for usability prediction, that is, as an automated quality assessment tool that reduces design, testing, and maintenance time.

The QUIM framework serves basically as a consolidated model under which other models for usability measurement, the QUIM model decomposes usability into factors, criteria and metrics.

In contrast to other hierarchical models, QUIM has two explicit supplementary levels, the data, and data collection methods. Data are elements of usability metrics, that is, they are quantities that are combined in the function that define the metric, by themselves, data are not generally interpretable as a measure of some facet of usability.

A usability metric is based in part on this datum could be the proportion of these objects that are actually relevant to a particular task.

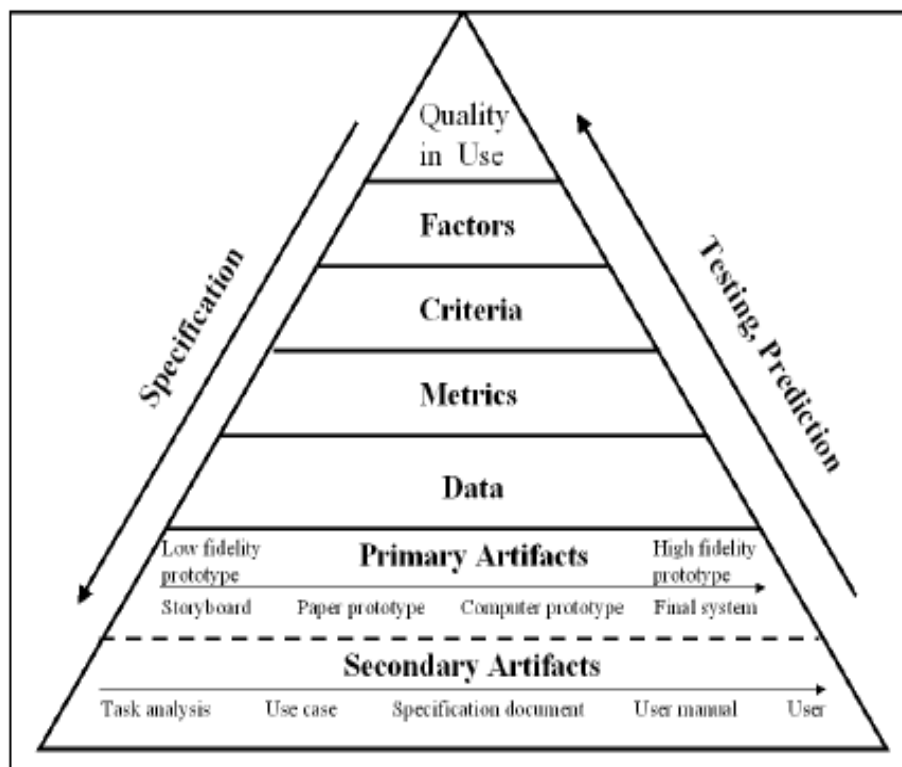


Figure 2.5: QUIM Structure (Seffah, 2006)

Figure 2.5 shows the structure for QUIM is a hierarchical model .Contains four levels called factors, criteria, metrics and data. There is relationship between these layers.

The rest of the levels (Primary Artifacts, Secondary Artifacts), it is not inside part of the project because of the difficulty of linkage between them and the rest of the layers being different performance levels with the first four.

2.7.2 10 Usability Factors of QUIM

The 10 usability factors briefly described next are included in the QUIM consolidated model (Seffah, 2006):

- i. **Efficiency:** the software product to enable users to spend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.

- ii. **Effectiveness:** the software product to enable users to achieve specified tasks with accuracy and completeness.

- iii. **Productivity:** This is the level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system. In contrast with efficiency, productivity concerns the amount of useful output that is obtained from user interaction with the software product.

- iv. **Satisfaction**, which refers to the subjective responses from users about their feelings when using the software (i.e. is the user satisfied or happy with the system?). Responses from users are generally collected using questionnaire.

- v. **Learnability** or the ease with which the features required for achieving particular goals can be mastered. It is the capability of the software product to enable users to feel that they can productively use the software product right away and then quickly learn other new (for them) functionalities.

- vi. **Safety**, which concerns if a software product limits the risk of harm to people or other resources, such as hardware or stored information. It is stated in the ISO/IEC 9126-4 (2001) standard that there are two aspects of software product safety, operational safety, and contingency safety. Operational safety refers to the capability of the software product to meet the user requirements during normal operation without harm to other resources and the environment.

- vii. **Trustfulness** or the faithfulness a software product offers to its users. This concept is perhaps most pertinent concerning e-commerce websites (e.g., Ahuja, 2000; Atif, 2002), but it could potentially apply to many different kinds of software products.

- viii. **Accessibility**, or the capability of a software product to be used by persons with some type of disability (e.g., visual, hearing, psychomotor). The World

Wide Web Consortium (Caldwell et al., 2004) suggested various design guidelines for making Web sites more accessible to persons with disabilities.

- ix. **Universality**, which concerns if a software product accommodates a diversity of users with different cultural backgrounds (e.g., local culture is considered).
- x. **Usefulness** or if a software product enables users to solve real problems in an acceptable way. Usefulness implies that a software product has practical utility, which in part reflects how closely the product supports the user's own task model. Usefulness obviously depends on the features and functionality offered by the software product. It also reflects the knowledge and skill level of the users while performing some task (i.e., not just the software product is considered).

2.7.3 Metrics

Based on usability measurement standards, a total of 127 specific usability metrics have been identified. Some metrics are basically functions that are defined in terms of a formula, but others are just simple countable data. Countable metrics may be extracted from raw data collected from various sources such as log files, video observations, interviews, or surveys. Examples of countable metrics include the percentage of a task completed, the ratio of task successes to failures, the frequency of programme help usage, the time spent dealing with programme errors, and the number of on-screen user interface elements.

Calculable (refined) metrics are the results of mathematical calculations, algorithms, or heuristics based on raw observational data or countable metrics. For example, a proposed formula by Bevan and Macleod (1994) for calculating task effectiveness is:

$$TE = \text{Quantity} \times \text{Quality}/100 \quad (2.1)$$

Where Quantity is the proportion of the task completed and Quality is the proportion of the goal achieved. The proportions mentioned are the countable metrics that make up the calculable TE metric. Listed in Table 2.5 are examples of additional calculable metrics, including in QUIM.

2.7.4 Measurable criteria

Each factor in QUIM is broken down into measurable criteria (sub-factors). A criterion is directly measurable via at least one specific metric. Presented in Table 2.3 are definitions of the 26 criteria in QUIM. These definitions assume a particular context of use or stated conditions for a software feature. Summarized in Table 2.4 are the relations between the 10 usability factors in QUIM.

Table 2.3: Usability Criteria in the QUIM Model (Seffah, 2006)

Criteria	Description
Time behavior	Capability to consume appropriate task time when performing its function.
Resource utilization	Capability to consume appropriate amounts and types of resources when the software performs its function (ISO/IEC 9126-1, 2001).
Attractiveness	Capability of the software product to be attractive to the user (e.g., through use of colour or graphic design; ISO/IEC 9126-1, 2001).
Likeability	User's perceptions, feelings, and opinions of the product (Rubin, 1994).
Flexibility	Whether the user interface of the software product can be tailored to suit users' personal preferences.
Minimal action	Capability of the software product to help users achieve their tasks in a minimum number of steps.

Table 2.3 shows usability criteria in the QUIM model, criteria are some factors. The difference is that they are measurable through a set of metrics.

Table 2.4: Relations between Factors and Criteria in QUIM (Seffah, 2006)

Criteria	Factors									
	Efficiency	Effectiveness	Satisfaction	Productivity	Learnability	Safety	Trustfulness	Accessibility	Universality	Usefulness
Time behavior	+			+		+		+		+
Resource utilization	+	+		+	+		+		+	+
Attractiveness			+			+		+		
Likeability	+		+		+	+	+	+	+	+
Flexibility	+	+	+							
Minimal action			+		+		+		+	+

Table 2.4 shows the relations between factors and criteria in QUIM. Factor represents the behavioral characteristic of a system. For examples: Efficiency, Effectiveness, Satisfaction, and Productivity. A criterion is an attribute of a factor that is related to software development. For example: Modularity is an attribute of the architecture of a software system.

Relationship between factors and criteria each factor is positively influenced by a set of criteria and the same criterion impacts a number of factors. Some factors positively impact others. An effort to improve the correctness of a system will be increase its reliability.

Table 2.5: Examples of Factors Metrics in QUIM (Seffah, 2006)

Metric	Description
Essential Efficiency (EE; Constantine & Lockwood, 1999)	<p>The relationship between the inputs of the production process on one hand and between the outputs resulting from this process.</p> $EE = \frac{S_{Essential} - T}{S_{Essential}} * 100$ <p>S_{enacted} S_{essential} = The number of user steps in the essential use case narrative, T=The amount of the product to be tested.</p>
Effectiveness (Constantine & Lockwood, 1999)	<p>The system's ability to achieve the goals is calculated with the increase in costs is an indicator of the efficiency of the system, it is calculated with the following equation</p> $= \frac{W * \text{Years of use of the system}}{T} 100\%$ <p>W= The proportion of actual production of the product, T = The amount of the product to be tested.</p>
Task Concordance (TC; Constantine & Lockwood, 1999)	$TC = 100 \times D/P$ $P = N (N - 1)/2$
Measures how well the expected frequencies of tasks match their difficulty, favours a design where more frequent tasks easier are made easier (e.g. fewer steps)	<p>N = The number of tasks being ranked, D = Discordance score, i.e., the number of pairs of tasks whose difficulties are in the right order minus those pairs whose difficulties are not in right order</p>
Task Visibility (TV; Constantine & Lockwood, 1999)	$TV = 100 \times (1/S_{total} \times \sum V_i) \forall i$ <p>S_{total} = Total number of enacted steps to complete the use case</p>
The proportion of interface objects or elements necessary to complete a task that are visible to the user	<p>V_i = Feature visibility (0 or 1) of enacted step i (i.e., how to count enacted steps and allocate a visibility value to them is defined by some rules in the reference)</p>

Hence, data should be collected in order to quantify the criteria. For example, a developer could, within the QUIM framework, devise a testing plan and benchmark reports that can be developed during the requirements phase and used later during the

evaluation phase. If requirements for usability should change (e.g. an additional factor is deemed necessary), then a new usability measurement plan could be derived under QUIM. Compared to the original measurement plan, the modified measurement plan would indicate the additional data that should be collected in order to evaluate the new usability criteria. Both the original and modified usability measurement plans would have consistent definitions under QUIM, which may facilitate the integration of usability and its measurement in the software development life cycle. This goal is especially important in a software quality assurance model.

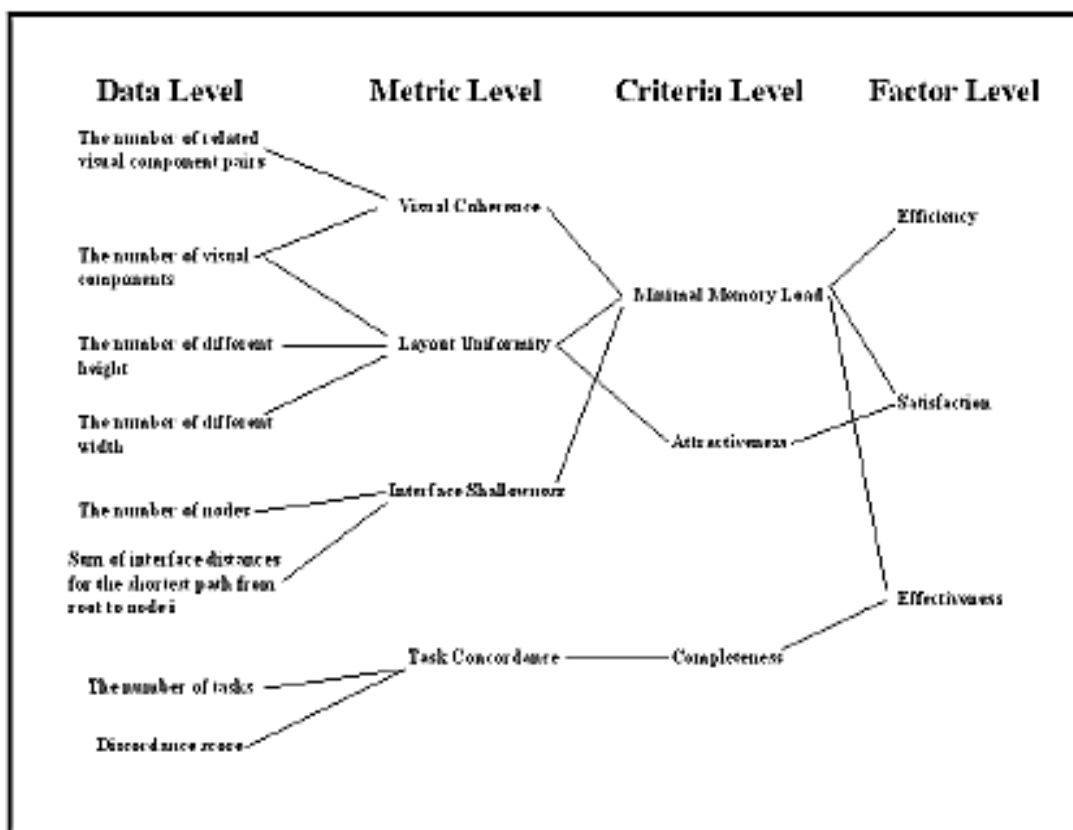


Figure 2.6 Example QUIM Components Relationship (Seffah, 2006)

Figure 2.6 shows those relationships, the data is an input to two different metrics “Visual Coherence” and “Layout Uniformity” (Constantine, 1999). QUIM is not exactly a tree. A specific metric could affect more than one criterion and then it is connected to more than one criterion. This is also the cases at every level.

References

- Ahuja, V. (2000). Building Trust In Electronic Commerce, IEEE Educational Activities Department Piscataway, IT Professional 2: 61–63, Volume 2, Issue 3. May, 2000, USA.
- Atif, Y. (2002). Building Trust In E-Commerce, 6: 18–24, Volume 6, Issue 1. Feb., 2002.
- Attia, M. (2008). Handling Arabic Morphological and Syntactic Ambiguities Within The Lfg Frame-Work With A View To Machine Translation. Phd Dissertation, University of Manchester. Badawi, Carter, And Ulliy, 2007. Modern Written Arabic: A Comprehensive Grammar. Routledge, London.
- Attia, M. (2008). A Compact Arabic Lexical Semantics Language Resource Based On The Theory Of Semantic Fields, Lecture Notes On Computer Science (Lncs): Advances In Natural Language Processing.
- Badrashinya, A. R., Attia1, M., Rashwan, B., & Basoumy, A. (2009). Compact Arabic Lexical Semantics Language Resource Based On The Theory Of Semantic Fields. M the Engineering Company for The Development Of Computer Systems, Egypt.
- Barlow, M. (2006). Monoconc And Corpus Analysis Using MP 2.2. University of Auckland, NZ.
- Briand, L.C. (1996). Property-Based Software Engineering Measurement. IEEE Transactions on Software Engineering, Pp 68-85.
- Chomsky, N. (1965). Aspects of The Theory Of Syntax. Mit Press. Language Arts & Disciplines, Pp 251. Cambridge, Ma.
- Chowdhury, G. (2005). Natural Language Processing Dept. Of Computer And Information Sciences University Of Strathclyde, Glasgow.
- Dane, M. (2012). System And Method For Automatically Processing Candidate Resumes And Job Specifications Expressed In Natural Language Into A Normalized Form Using Frequency Analysis Us.

- Dichy, H. (2005). The Dinar.1 (Dictionnaire Informatisé De L'arabe, Version Arabic Lexical Resource, an Outline of Contents And Methodology, The Elra Newsletter.
- Elkateb, F. (2008). Modifying A Natural Language Processing System For European Languages To Treat Arabic In Information Processing And Information Retrieval Applications Gregory Grefenstette, Nasredine Semmar, Multilingual Multimedia Knowledge Engineering Laboratory.
- Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Trans. Asian Lang. Inform. Process* 8, 4, Article 14 (2009, Dec), from <http://doi.acm.org/10.1145/1644879.1644881>.
- Fraser, A., & Wong, W. (2008). The Language Weaver Statistical Machine Translation Software System. *Arabic Computational Linguistics*, CSLI Publications, Chapter 9, pages 257-288.
- Habash, N., & Rambow, O. (2005). A Morphological Analysis and Generation For Arabic Dialects. Center for Computational Learning Systems. Columbia University, USA.
- Habash, N., & Rambow, O. (2007). Morphophonemic and Orthographic Rules in a Multi-Dialectal Morphological Analyzer and Generator for Arabic Verbs, International Symposium on Computer and Arabic Language (ISCAL), Riyadh, Saudi Arabia.
- Habash, N., & Rambow, O. (2007). Arabic Diacritization through Full Morphological Tagging, In Proceedings of the North American chapter of the Association for Computational Linguistics (NAACL), Rochester, New York.
- Landauer, S. (1995). *The Trouble with Computers: Usefulness, Usability And Productivity*. Cambridge, MA.
- Mcenery, A., Garside, J., & Geoffrey, L. (1997). *Corpus Annotation: Linguistic Information from Computer Text Corpora*. Longman Isbn. London
- Mohammad, A. S., & Badrashiny, A. A. (2009, June). Diacritizer For Arabic Texts, Faculty Of Engineering, Cairo University Giza, Egypt.
- Muhtaseb, H. A. (2008). *Some Differences between Arabic and English: A Step Towards an Arabic Upper Model*, Fredonia. Department of Language Learning and Leadership, New York

- Nabil, J. (2006). The Arabic Language Site Within The Family Of Semitic Languages Using Timelines And Linguistic Maps.
- Nassef, H. (2008). Arabic Grammar developed. Chapter 3, pages 120-135, University of Cairo.
- Nelson, A. (2007). A Corpus-Based Study Of Business English And Business English Teaching Materials Unpublished, Manchester: University Of Manchester.
- Samarrai, I. A. (2000). History Of The Arabic Language And This Reference Speaks About The History Of The Language And Its Components And Large Signs In This Language.
- Seffah, A. (2006). Usability Measurement and Metrics: A Consolidated Model. *Software Qual J*, 14: 159–178. Concordia University, Montreu.
- Oudah, M. M., & Shaalan, K. (2012, Dec). A Pipeline Arabic Named Entity Recognition Using a Hybrid Approach. *Technical Papers*, Pp 2159–2176, COLING 2012, Mumbai.
- Shaalan, K., & Amer, J., (2009). Named Entity Recognition For Arabic. *Soc. Inform. Sci. Technol. Text In Multilingual Speech-To-Speech Machine Translation Framework, Machine Translations*.
- Zaidan, O. (2012). Crowd Sourcing Annotation For Machine Learning In Natural Language Processing. Tasks A Dissertation Submitted To The Johns Hopkins University In Conformity With The Requirements For The Degree Of Doctor Of Philosophy. Baltimore.