# A Genuine Framework

# To Control the Quality of Arabic and Arabized Software

July, 2015

#### Submitted By:

Nawras Ali Othman	Fares Ali Othman	Komalpreet Singh
B.Sc. in Communication	M.A.degree in Library &	B. tech. in Electrical
Engineering Technology	Information science	Engineering
Othman.nawras@gmail.com	2010fares@gmail.com	kpsgne@gmail.com
Amman, Jordan	Amman, Jordan	Punjab, Jordan

#### Abstract

This paper presents a new model and an efficient approach to control the quality of software having Arabic character and solving different issues facing Arabic software. The research consists of two phases, in first phase the previous research papers, interview with developers and personal experience of authors are used to investigate and list out the challenges facing Arabic software with suggested solutions, if any. In the second phase the standards and solutions for problems related to Arabic software are collected and bound together in order to construct the given model. "Internationalization" and "Localization" are the two major challenges faced by developers in Arabization process. Different scenarios and points of views were taken into account to construct this model which lists out the specifications and requirements to make the software more conductive to Arabic language. At last, Arabized version of French integrated library system PMB was taken to check the implementation of this model.

#### Introduction:

There are approximately 295 million native speakers of Arabic language, making it one of the most spoken languages in this world. Countries where first language is

Arabic, whole work in all contexts is made in Arabic and for this one needs software of good quality which supports Arabic characters. But there are some challenges before Arabic software industry and as soon there will be problems in Arabic software and theone will not be able to use them efficiently. To get rid of problems and challenges facing Arabic software industry, the responsibility of creating a model which controls the quality of the software is taken in this research paper. In order to control the quality, first of all problems in software having Arabic character must be identified and then solved. Most of the few researches that investigate the challenges facing Arabic software presented them as internationalization and localization problems of Arabic language. In general, W3C defines internationalization of software as "the design and development of a product, application or document content that enables easy localization for target audiences that vary in culture, region, or language [1]" And localization as "the adaptation of a product, application or document content to meet the language, cultural and other requirements of a specific target market (a locale) [1]" Bentahar and Mrabet define internationalization and localization as "the process of designing software to support different languages and cultures without having to change the program. In contrast, localization is process of modifying an internationalized program so that it behaves correctly in a given language and culture [2]". In particular, Global Partners International (GPI) defines Internationalization (I18n) of Arabic as "the process of enabling back-end technologies to function or support the Arabic language and the locales in which it is used [3]" and localization (110n) of Arabic as "localization deals primarily with front-end or linguistic and cosmetic aspects of an Arabic-language software application or Website; this includes locale-specific content, cultural correctness, translations, and design. [3]". The challenges facing software having Arabic character are discussed in some previous researches. For example, Engström presented the main processing problems in the Arabic script in five points which are "Arabic is a cursive, context-dependent, and bidirectional language, many different character set standards are used in the Arabic-speaking locales, there are many different encoding methods in use within the Arabic-speaking locales, there is no standard keyboard layout and typesetting methods are not well established for Arabic, and special rules for issues such as text direction and linguistic boundaries are needed [4]". Galdo and Nielsen also listed all possible problems facing Arabic language as following: "Character codeset and standard encoding, character

Shaping and text direction algorithms, character fonts, global Screen direction and mirror effect, numerals and Hindi Digit shapes, Arabic vowels and collating sequences, neutral characters, dual keyboard management and optical Character Recognition[5]". Problems related to Arabic standards and culture as following: "Handwriting, codeset uses, local differences, no abbreviations and justification of text [5]" were also listed by them.

In an effort to find a way to solve problems of Arabic localization, Abufardeh and Kenneth stated that we need to understand the aspects of the localized language in order to get localized software of high quality. They also had addressed some topics to be highlighted by future researches: "many of the technical limitations which existed in the past including: limited keyboard support; limited Arabic in ISO Unicode; and a limited number of fonts are no longer problems. However, the absence of clear and comprehensive Arabization process, the lack of standards, and the lack of tools available which can handle Arabic script continues to be major obstacles that need to be addressed by researchers in this field [6]". An ISO solution for Arabic character sets is ISO/IEC 8859-6 [7]. However, it does not handle the different shapes of Arabic letters depending context. More sufficient character encoding is represented by UTF8 which uses 8-bit code units and it covers all possible character in Unicode. Moreover, UTF8 encodes all contextdependent of each Arabic letter [8]. Unicode organization support bidirectional languages; it is stated in Unicode standard that "in the case of bidirectional text, there are circumstances where an implicit bidirectional ordering is not sufficient to produce comprehensible text. To deal with these cases, a minimal set of directional formatting characters is defined to control the ordering of characters when rendered. This allows exact control of the display ordering for legible interchange and ensures that plain text used for simple items like filenames or labels can always be correctly ordered for display [9]". Unicode standards also support many Arabic fonts.

There were almost no researches related to quality of Arabic or Arabized software. However, ISO/IEC 25010 defines a product quality model: "A product quality model composed of eight characteristics (which are further subdivided into subcharacteristics) that relate to static properties of software and dynamic properties of the computer system. The model is applicable to both computer systems and software products. [10]". In general, ISO/IEC 25000 series defines the

process for software product evaluation; it consists of five divisions which are: quality management division, quality model division, quality measurement division, quality requirements division and quality evaluation division [11]. Arabic as supported language can be an element of software accessibility measures according to ISO/IEC 25023. ISO/IEC 25023 defines accessibility as "degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use. [12]". Although this single measure can quantify the internationalization of software including Arabic internationalization, it doesn't represent the percentage of internationalization done for each language, it is a number of supported languages which actually software supports over the languages needed to be supported with clarification what kind of support is this. Finally, all previous researches proposed specific solutions for some challenges facing Arabic software, but software industry in Arab world still lack the models and standards which consider all challenges created by Arabic language in both technical and nontechnical aspects in order to enhance the quality of produced software which have Arabic character.

# Problems, objectives and limitation of research:

#### Problem statement:

The absence of the frameworks which control the quality of Arabic and Arabized software and lacking for standardization and models regarding to Arabic software industry in general.

#### Objectives:

The main objective of this research is to create a new model which controls the quality of software having Arabic character in order to solve the lack of standardization and modeling regarding to Arabic software in general with taking different points of views and scenarios of software into consideration conducting the experience of authors, interviews and literature review. The objectives are also listed as following:

- 1. To investigate the challenges facing software having Arabic character.
- 2. To review and collect the standards related to software having Arabic character.

3. To suggest a new integrated model to control the quality of Arabic and Arabized software based literature review and authors' experience.

Limitations:

- 1. Target software: the suggested quality model of this study specifically targets software having Arabic character.
- 2. This model does not include direct solutions for challenges facing Arabic software except some suggested solutions; however, it states the quality factors which have to be standardized and their stages.

# Methodology

The sequence of operations which was followed to build a model which ensures the quality of Arabic and Arabized software is shown in figure (1). The first step was reviewing literature to gain knowledge about challenges facing software with Arabic character and their proposed solutions by other researchers. The second step was interviewing developers and specialists to get to know all possible scenarios and points of views which must be considered in upcoming model. Third; related international standards were projected on the new model. Finally, all information mentioned in previous steps is collected and then a new model is created. In order to show how this model can be applied, Arabized version of French integrated library system PMB is selected as a case study.

# **Challenges of Arabic software industry**

Challenges of Arabic software industry can be classified into technical and non-technical shown in table (1):

Challenges of Arabic software industry					
Technical aspects		Non-Technical aspects			
Arabic character issues	Other issues of Arabic language	<ul> <li>Social challenges.</li> <li>Economic challenges.</li> <li>Absence of qualified staff.</li> <li>Political challenges.</li> </ul>			

# Table (1) Challenges of Arabic software industry



Figure (1): the procedure followed to create a new quality model for Arabic and Arabized software.

First, Technical Challenges of Arabic scripts:

Technical challenges can be divided into two groups; first one related to Arabic characters and the second concerns the other issues of Arabic language

I. Arabic characters:

a) Arabic character shaping: Arabic has 28 letters and each letter has four forms: Initial, Medial, Final and Isolated and the letter shape changes depending on its position in word.

b) Diacritics (Tashkil): they are short vowel marks added above or below the letter in order to refer specified meaning of word. Although diacritics are not used commonly, they are used in legal documents and religious books.

c) Numeral characters: Middle Eastern countries use Hindi numbers which are written from right to left and North African countries use Arabic numbers which are written from left to right. Herein, the necessity of the support of two systems rose.

d) Neutral characters: Neutral characters in Arabic script are same to those used in Latin scripts. However, Arabic is written from right to left, in contrast neutral Latin characters are written from left to right. This causes inverting of the neutral character direction and hence its meaning may change in some cases.

#### II. Other issues:

- a) Text direction: Arabic is written and from right to left, in contrast Latin texts are written and from left to right. When a mix text of Arabic and Latin is written on the same line, the words get mixed and text loses its meaning.
- b) Arabic fonts and Calligraphic styles: The changing graphic form of Arabic letters according to their position in word creates a challenge in supporting more Arabic fonts.
- c) Date format: Arabic countries use western and Islamic calendars; both of they must be supported.
- d) Styles and colors: Arabic different cultures and traditions shall be considered in selecting styles and colors.
- e) No abbreviations: Arabic language does not use abbreviations or acronyms, this characteristic of Arabic should be considered in translating into Arabic.
- f) Terminology: correct terminology shall be used properly.

g) Lack of standards and models which supports Arabic, Arabic supported and Arabized software.

Second, Non-Technical challenges of software industry in Arab world:

- 1) Social challenges.
- 2) Economic challenges.
- 3) Absence of qualified staff.
- 4) Political challenges.

# Arability model

There are many standards and approaches to control software quality such as ISO/IEC 25000 series. Although these standards are useful to measure and quantify quality characteristics of software, they do many generalizations. Moreover, they do not include techniques to control the quality of localization and internationalization for specific language or community which are considered very challenging for some languages like Arabic, Chinese...etc. One of the suggested models to deal with challenges related to Arabic language is Arability model. This model highlights the quality of Arabic, Arabic supported and Arabized software. Though this model does not include how to do the processes of creating Arabic or Arabized software, it is supposed to make approaches with well-analytical mathematical methods and techniques to measure the quality of Arabic and Arabized software in all phases of software development life cycle (SDLC). Furthermore, Arability model shall list the specifications and requirements which make the software more Arable (having the quality of Arabic character).

# A. Arability term

Arability is compound noun of the first two letters of the word "Arabic" and the suffix "ability". Ability suffix means "the quality of being able to be or having to be" or "the fact of having the quality mentioned. [13]"Hence, Arability is the quality of something of being able to be Arabic or having the quality of an Arabic character.

Arability term can be confused with term "Arabization". However, the two terms cannot be used interchangeably. Arabization is the noun form of the verb "Arabize" which is defined in oxford dictionary as "give (someone or something)

an Arab or Arabic character. [14]".In contrast, "Arability" discusses the quality of something or someone of being able to be Arabic or having the quality of Arab or Arabic character. The fact is Arabization is sub term in Arability model as discussed later.

To be more specific, Arability of software is the quality of software of being able to adopt an Arabic character or having the quality of an Arabic character. The Arable software is the software which has the quality of being able to adopt an Arabic character or having the quality of an Arabic character.

# B. Scenarios and stages of Arability

There are three scenarios that Arability model must standardize their stages:

- 1. Creating Arabic-able software from scratch.
- 2. Creating software that supports multi languages including Arabic.
- 3. Giving an exist software an Arab or Arabic character. In other words, Arabization scenario.

Whether the first scenario, second or the third is applied, the standardization must include the stages of software development which are:

- 1. Requirement analysis: which are classified to functional and nonfunctional requirements which meet the needs of Arabs and Arab communities? (ISO standards are helpful in this case [15, 16]).
- 2. Design and implementation.
- 3. Evaluation: including testing the internal quality, external quality and quality in use of the software. This means we have to answer the question: Does the produced software match the needs of the Arabic communities? (ISO standards are also helpful in this case[15, 17]).

In few words; Arability as a suggested model containing standards to control quality of Arabic, Arabic supported and Arabized software. It shall also give solutions for technical and non-technical for the all stages of Arabic, Arabic supported and Arabized software life cycle

### C. Arability model points of view:

Arability model controls the quality of Arabic, Arabic supported and Arabized software from three points of view:

- I. The process: Arability model shall state the requirements, measures and evaluation of each process through SDLC in order to control the quality of processes.
- II. Specialists: Arability model shall state the qualifications needed for Arability specialists.
- III. Standardization: Arability shall state standards to control the quality of produced software.

It is clear that the more quality of process, the more complex functions are implied therefore, more qualified specialists are needed to do this complex process. It is also clear that the quality of processes and qualifications of specialists control the quality of produced software. It can be summarized as following:

Well-qualified specialists  $\rightarrow$  High quality of processes  $\rightarrow$  high quality of software

# D. Arability factors:

Points of views of Arability can be treated as factors which control the quality of produced software. Quality of process and qualified staff can play a great role in the quality enhancement of Arabized and Arabic software. Unless these factors are controlled as prescribed in related standards, the quality of software itself will definitely fall down. Stating the requirement of software, designing and implementation or Arabization and evaluation of software are three stages, if they are done as per standards, the quality of process for software will be controlled and enhanced. In contrast, the quality of the second factor is controlled if the qualification requirements of staff, training and development process and evaluation of staff are done according to related standards. Figure (2) lists the stages which are controlled by Arability factors.

Figure (2) Arability factors.



#### The phases of Arabization scenario:

Arabization is the third scenario of Arability model where foreign software is being Arabized. Arabization scenario consists of three phases shown in table (2):

Arabization scenario phases						
Pre- Arabi	zation phase					
Pre selection stage	Selection stage	Arabization phase	Post-Arabization phase			

#### Table (2) The phases of Arabization scenario

First, Pre-Arabization phase: it is similar to the requirements gathering phase in SDLC. Functional and non-functional Requirements of desired software in specified context of use shall be written in this phase. One point to consider here is that the software is not designed from scratch in Arabization scenario; however, software is re-formed in way that meets the needs of Arabs and Arab communities. Therefore, the requirements of Arabized software in specified context of use are the criteria which decision of selecting the desired software shall be based on. Hence, Pre-Arabization phase consists of two stages:

- a) Pre-selection stage: the requirements of desired software shall be stated in this stage. These requirements can be categorized into two groups; first one is the requirements related to software in specified context of use. For instance, library management system shall do library functions based on prespecified criteria. In other words, first group represents functional and nonfunctional requirements of desired software. In contrast, Second group represents the Arability requirements. Arability requirements can be listed as below:
  - 1. Proper Arabic-compatible character encoding shall be used.
  - 2. Proper text direction algorithm shall be used.
  - 3. Arabic fonts and calligraphic shall be supported.

- 4. Arabic date format shall be supported (Islamic and western calendars)
- 5. Styles and colors which are appropriate to Arabs and Arabic communities shall be used.
- 6. Abbreviations shall not be translated literally but their long form shall be stated.
- 7. Terms and idioms shall be translated and confirmed by specialists.
- b) Selection stage: After stating the requirements of pre-selected software, software which meets the pre-specified requirements at different percentages shall be nominated to be Arabized in the next phase. Then, the final selection shall be made, this will answer the question; what is the most appropriate software in comparison to other suggested software in meeting the pre-specified requirements? Comparative studies where two or more suggested software are put together into comparison to find their matching degree with requirements can also be very helpful.

Second, Arabization phase: it is core of Arabization process. In this phase, software isn't only translated into Arabic, but also its back-end and front-end technologies are enabled to function and support Arabic language and meet Arab users and Arabic communities' needs. Arability requirements shall be fulfilled in this phase. Moreover, all quality characteristics of Arabized software shall be tested and qualified during Arabization process. Therefore, the quality of Arabization shall be controlled and enhanced.

Third, Post-Arabization phase: after Arabization phase is completed, the quality of produced software shall be evaluated. The evaluation process requires evaluators, standards and mathematical approaches to quantify the software quality attributes. Evaluators can be users of produced software, acquirers, and developers of software or independent evaluators. Evaluation shall trace all drawbacks of the previous two phases of Arabization. Hence, the quality of Arabized software can be enhanced based on evaluation results. Furthermore, evaluation shall ensure that produced Arabized software the requirements which are written in pre-selection stage of first

phase of Arabization scenario. The quantified quality of software shall be called "Arability degree" which can be calculated in various ways depending on Evaluator's decision, a suggested simple way to calculate Arability degree of Arabized software is to find the difference between the quantified quality measures of software and its Arabized version. The less difference, the Arability degree increases and the software is more arable. Quantified quality measures can be calculated using equations and methods which are stated in ISO/IEC 25023[12].

# Case study:

For purpose of illustration, Arabization phases for selecting, Arabizationand evaluating of an integrated library system (ILS) is chosen as a case study. In pre-Arabization phase PMB ILS is selected for Arabization.PMB is free open source integrated library system including; cataloging, circulation, acquisition, SDI, reporting, OPAC...etc. Table (3) shows the summary for the implementation of Arability model in this case study.

	Arabization scenario phases to select, Arabize and evaluate an integrated library system						
Pre- Arabization phase		Arabization phase	Post-Arabization phase				
	Pre-selection stage	Selection stage					
•	The requirements by ministry of education in Jordan for desired integrated library system. Arability requirements.	PMB integrated library system is selected based on previous related researches.	Arability requirements shall be fulfilled in this phase	Evaluation techniques to quantify quality and to calculate Arability degree are needed.			

Table (3): Arabization scenario phases to select, Arabize and evaluate an integrated library system

First, Pre-Arabization phase:

#### **Pre-selection stage:**

The requirements by ministry of education in Jordan for desired integrated library system are listed below:

- 1. Multi languages shall be supported.
- 2. Arabic characters shall be supported.
- 3. Documentation shall be available.
- 4. Multi-language documentations shall be supported.
- 5. Updates on regular basis shall be available.
- 6. Credibility and trust.
- 7. Sustainability.
- 8. Compatibility with web browsers.
- 9. The system shall be supported by different operating system such as Linux, Windows and Mac.
- 10. Quantity of records which database of system accepts.
- 11. Cloud based system.
- 12. Attractiveness.
- 13. Compliance.
- 14.System is General Public License (GPL) compatible.
- 15. The programming languages of system are GPL compatible.
- 16. The database of system is GPL compatible.
- 17. Authorization.
- 18. Backup module.

- 19. Data migration including import and export.
- 20. Maintenance module.
- 21. Desktop application.
- 22. The system shall support scanning technology (vertical coding-barcode).
- 23. The system shall support different types of information sources.
- 24. ISBN and ISSN.
- 25. Search engine in cataloging unit.
- 26. Files attachment.
- 27. References.
- 28. Copy cataloging.
- 29. Acquisition module.
- 30. Circulation module.
- 31.Serial control module.
- 32. SDI module.
- 33. Inventory module.
- 34. Reports module.
- 35. Marc compatible.
- 36. ISBD standard shall be supported.
- 37. The system shall support XML.
- 38.Dublin core shall be supported.
- 39. Z39.50 protocol shall be supported.
- 40. LDAP protocol shall be supported.
- 41. OAI protocol shall be supported.

- 42. Field authority.
- 43. Synonyms files shall be available.
- 44. Stop-words file shall be available.
- 45. Thesaurus module shall be available.
- 46. Union Catalog and Cooperative Cataloging shall be available.
- 47. Interlibrary Loan (ILL) shall be supported.
- 48. OPAC shall support simple and advanced search.
- 49. OPAC shall support Historical Search.
- 50. OPAC shall support Hierarchical Search.
- 51. OPAC shall support External Search.
- 52. Federated Search Engine shall be supported in OPAC.
- 53. Portal unit shall be available.
- 54. Search for full text in abstract and notes section.
- 55. Ability to customize.
- 56. RFID shall be supported.
- 57. E-document management shall be available.
- 58. Cards Generating shall be available.
- 59. The system shall be able to be operated via USB (Portable).
- 60. OPAC Management Module shall be available.

Note: the second requirement can be treated as one of the requirements of Arability model.

Arability requirements are stated in Arability scenario sections.

# Selection stage:

"Degree of Suitability of PMB & SLIMS Open Source Systems For School Libraries in Light of The Requirements of The Ministry of Education in Jordan" is a thesis done by Othman in order to find the more suitable open source software for school libraries in Jordan between PMB and SLIMS based on the requirements of ministry of education in Jordan which we mentioned in the previous section. The results of this thesis showed that the quantified degree of suitability for SLIMS is 55 [18]. In contrast, the degree of suitability for PMB is 200. Obviously PMB is more suitable than SLIMS for school libraries in Jordan according to this study. "A comparative study of two open source Integrated Library Systems (ILS): PhpMyBiblio and NewGenLib" is another study made by researcher Fares Ali and Professor Fawaz Zaghoul [19]. This study showed that PMB is more suitable for school libraries. In contrast, NewGenLib is more suitable for college libraries.

**Second,** Arabization phase: In this phase, Arability requirements are fulfilled as following:

- 1. UTF8 is used as character encoding.
- 2. CSS direction property (rtl) is used to solve text direction challenge.
- 3. Styles and colors are customized to match the Arabic community personality.
- 4. The long form of abbreviations is translated.
- 5. Terms are translated by the specialists from ArLISC team.

UTF8 stands for Universal Coded Character, UTF8 uses 8-bit code units, and it is also capable to encode all possible characters in Unicode which is "a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems [8]". Moreover, UTF8 is the most used character encoding for World Wide World. World Wide Web Consortium (W3C) recommends using UTF8 as default encoding character in XML and HTML. Furthermore, UTF8 supports all Arabic letters of different shapes, with and without Tashkil. It also supports Arabic and Hindi numbers. In addition, UTF8 supports neutral characters in Arabic.

CSS stands for Cascading Style Sheets; it is used to change the style of web pages and cloud-based software. CSS Direction property is used to change the text direction. In our case, the desired text direction is from right to left. Therefore, direction property is set rtl (right to left). Since PMB targets libraries, the terms and idioms must be related to library field. The researcher and librarian Fares Othman tries his best to find the most appropriate Arabic equivalents for terms and idioms mentioned in PMB French version. For instance, "circulation" is a term related to libraries and information science which including the activities of managing loans and return of books, it also includes the managements of borrowers. However, circulation word can have many meanings such as as blood circulation, flow, popularity, spreading...etc. As we can see, terms and idioms are challenging for specialists through Arabization process. Finally, the Arabization process of PMB was not only translating from French to Arabic, but the Arabic character was also added to this software by using Arabic-compatible character encoding (UTF8), rtl property to cope with Arabic text direction (from right to left) and terms of PMB get special treatment by specialists.

**Third,** post-Arabization: Arabized PMB hasn't been evaluated yet, new methods are needed to measure and quantify its quality with respect to international software quality standards (such ISO/IEC 25000 Series) and the quality of its original version. Different software characteristics must be tested and evaluated to ensure the quality of PMB Arabization.

#### **Conclusion and recommendations:**

In conclusion, a new framework to control the quality of Arabic and Arabized software is modeled considering the challenges facing Arabic software industry and the possible scenarios and points of views. It is a genuine well-designed and integrated model and without such model, the quality of Arabic and Arabized software will get lower. If Arabic or Arabized software are continued to be written without any standard or model to control their quality, there will not only be a shortage of Arabic or Arabized software of good quality, but Arab countries may not be able to keep pace with the acceleration of software advancement in various aspects of use. Finally developers, researchers or even users are invited to participate in forming different levels of Arability model. Mathematical approaches which quantify Arability degree in all scenarios for all quality

characteristics of software are highly needed. With cooperation, we all together can take Arabic and Arabized software industry to better place.

#### References

- [1] World Wide Web Consortium, "Localization vs. Internationalization" [Online]. Available: http://www.w3.org/International/questions/qa-i18n.
- [2] J. Bentahar and R. Mrabet, "Internationalization of a Distance Exam Web Environment," 21st International Unicode Conference, Dublin, Ireland.
- [3] Globalization Partners International, "Arabic Localization Services Arabization." [Online]. Available: http://www.globalizationpartners.com/resources/arabic-translationquick- facts/arabic-localization.aspx. [Accessed: 27-May-2015].
- [4] G. Engström, "Internationalization and Localization Problems in the Chinese and Arabic Scripts," Uppsala University, Sweden , 2008.
- [5] E. Galdo and J. Nielsen, International User Interface.: John Wiley & Sons, Inc., 1996.
- [6] S. Abufardeh and K. Magel, "SOFTWARE LOCALIZATION: THE CHALLENGING ASPECTS OF ARABIC TO THE LOCALIZATION PROCESS (ARABIZATION," FEBRUARY 2008
- [7] International Organization For Standardization, 'Information technology -- 8-bit single-byte coded graphic character sets -- Part 6: Latin/Arabic alphabet', 8859-6, 1999.
- [8] Wikipedia, the free encyclopedia, "UTF-8"[Online]. Available:https://en.wikipedia.org/wiki/UTF-8.
- [9] Unicode Consortium, "UNICODE BIDIRECTIONAL ALGORITHM" [Online]. Available: http://www.unicode.org/reports/tr9/.
- [10] International Organization For Standardization, 'System and software quality models', 25010, 2015.
- [11] ISO/IEC 25000, "Software Quality Model" [Online]. Available: http://iso25000.com/.
- [12] International Organization For Standardization, 'Measurement of system and software product quality', 25023, 2015.
- [13] A. Hornby and S. Wehmeier, *Oxford advanced learner's dictionary*, 8th ed. Oxford: Oxford University Press, 2011.
- [14] A.Stevenson, Oxford Dictionary of English, Third edit. NewYork: Oxford University Press, 2010.

- [15] G. Perspectives and E. M. May, "System Quality Requirement and Evaluation," Glob. Perspect. Eng. Manag., vol. 2, no. 2, pp. 52–59, 2013.
- [16] International Organization For Standardization, Software product Quality Requirements and Evaluation (SQuaRE) -- Quality requirements', 25030, 2007.
- [17] International Organization For Standardization, 'Systems and software Quality Requirements and Evaluation (SQuaRE) Evaluation process', 25040, 2011
- [18] F. Othman, 'Degree of Suitability of PMB & SLIMS Open Source Systems For School Libraries in Light of The Requirements of The Ministry of Education in Jordan', M.A. thesis, University of Jordan, 2014
- [19] F. Othman and F. Al-Zaghoul, 'A comparative study of two open source Integrated Library Systems (ILS): PhpMyBiblio and NewGenLib', in *Libraries and Information Centers in a Changing Digital Environment*, 2013, pp. 579-606.