Integrated Model for Software Usability

Sanjay Kumar Dubey
Computer Science and Engineering Department
Amity University
NOIDA, India
skdubey1@amity.edu

Anubha Gulati
Computer Science and Engineering Department
Amity University
NOIDA, India
anubha.gulati@gmail.com

Prof. (Dr.) Ajay Rana
Computer Science and Engineering Department
Amity University
NOIDA, India
ajay_rana@amity.edu

Abstract—In recent years, the usability of software systems has been recognized as an important quality factor. Many definitions and models of usability have been given so far but they are brief and informal. Most of these models also fail to cover all of the aspects of usability and are not well integrated. This paper proposes an integrated model that describes the concept of software usability and explains it by means of a detailed taxonomy.

Keywords-usability;model; taxonomy;

I. INTRODUCTION

In the last few decades the demand for usable software has increased exceptionally. It is mainly due to the change in users' perception of software systems and the increase in users' ability to distinguish software on the basis of quality. The idea of usability has been represented in various quality models over the last few decades and research shows that usability is a key component in the overall quality of a software product [26]. Usability can be understood as the degree to which software is usable by specified users with ease and comfort. ISO 9241-11 defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [12]. Usability is also defined as "the ease with which a user can learn to operate, prepare inputs for and interpret outputs of a system or component" [11].

Numerous definitions of usability have been given so far. However, these definitions are brief and informal. Neither researchers nor standards bodies have achieved consensus with regards to the concept of usability [1]. So as to overcome this limitation, this paper proposes an integrated model that describes the concept of usability based on five attributes, namely, Effectiveness, Efficiency, Satisfaction, Comprehensibility and Safety. A detailed taxonomy is also given wherein a description of each of these attributes and their sub-attributes is explained in a structured format.

II. LITERATURE SURVEY ON USABILITY MODELS

Over the past few decades, several different standards and models for quantifying and assessing usability have been proposed. In this section we review some of these models, highlighting the attributes on which usability has been considered to depend. Mc Call's model described usability as operability, training and communicativeness [15]. Boehm's model said that a software is usable if is portable and maintainable [6]. Shackel explained that a system is usable if it is effective, learnable, flexible and subjectively pleasing [7]. Bevan *et al*. considered usability based on the product, the user, ease of use and acceptability of the product [28]. In FURPS quality model the concept of usability includes aesthetics, human factors, online and context sensitive help, wizards and agents, user documentation, consistency in the user interface, and training materials [29]. IEEE Std. 1061

described usability as depending upon comprehensibility, ease of learning, and communicativeness [11]. Nielsen refers to learnability, efficiency, memorability, errors and satisfaction as usability attributes [18]. Preece *et al.* considered a classification that included safety, effectiveness, efficiency and enjoyableness [19]. Subsequently they proposed a new classification including learnability, efficiency, throughput, flexibility and attitude [20]. Dix *et al.* represented usability as learnability, flexibility and robustness [33]. ISO 9126-1 described usability as a combination of understandability, learnability, operability, attractiveness and usability compliance [14]. QUIM model describes usability as comprising of 10 factors, namely, efficiency, effectiveness, productivity, satisfaction, learnability, safety, trustfulness, accessibility, universality and usefulness [24]. Abran *et al.* extended the ISO 9241-11 definition by adding learnability and security [1]. Bass *et al.* described usability in terms of modifiability, scalability, reusability, performance and security [23]. Shneiderman *et al.* identified five usability measures, namely, time to learn, speed of performance, rate of errors by users, retention over time, and subjective satisfaction [8]. Alonso-Rios *et al.* explained usability as comprising of knowability, operability, efficiency, robustness, safety and subjective satisfaction [9]. Dubey *et al.* described usability in terms of effectiveness, efficiency, satisfaction and learnability [36]. Usability attributes given in various models are summarized in Table I.

TABLE I. USABILITY ATTRIBUTES IN VARIOUS MODELS

Model	Usability Attributes
McCall [15]	Operability, Training, Communicativeness
Boehm [6]	Portability, Maintainability
Shackel [7]	Effectiveness, Learnability, Flexibility, Subjectively Pleasing
Bevan *et al.* [28]	Type of Product, Type of User, Ease of Use, Acceptability
FURPS [29]	Aesthetics, Human Factors, Online and context sensitive help, wizards and agents, User Documentation, Consistency, Training Materials
IEEE Std. 1061 [11]	Comprehensibility, Ease of Learning, Communicativeness
Nielsen [18]	Learnability, Efficiency, Memorability, Errors, Satisfaction
Preece *et al.* [19]	Safety, Effectiveness, Efficiency, Enjoyableness
Preece *et al.* [20]	Learnability, Efficiency, Throughput, Flexibility, Attitude
Dix *et al.* [33]	Learnability, Flexibility, Robustness
ISO 9241-11 [12]	Effectiveness, Efficiency, Satisfaction
ISO 9126-1 [14]	Understandability, Learnability, Operability, Attractiveness, Usability compliance
Donyaee *et al.* [24]	Efficiency, Effectiveness, Productivity, Satisfaction, Learnability, Safety, Trustfulness, Accessibility, Universality, Usefulness
Abran *et al.* [1]	Efficiency, Effectiveness, Satisfaction, Learnability, Security
Bass *et al.* [23]	Modifiability, Scalability, Reusability, Performance, Security
Schneiderman *et al.* [8]	Time to learn, Speed of Performance, Rate of Errors by users, Retention over time, Subjective Satisfaction.
Alonso-Rios *et al.* [9]	Knowability, Operability, Efficiency, Robustness, Safety, Subjective Satisfaction
Dubey *et al.* [36]	Effectiveness, Efficiency, Satisfaction, Learnability

III. NEED FOR AN INTEGRATED MODEL

There are a number of standards and models in literature each of which describes usability in terms of a different set of attributes that are very briefly and vaguely defined. Also the models are not homogeneous i.e. when they overlap, they do so only partially, with different terms used to include the same attribute or with the same term used to describe different concepts. Therefore, they are very difficult to use and to communicate.

The lack of a consistent model leads to major problems in the evaluation of usability, as a consensus cannot be achieved on the definition of usability amongst researchers. There is very little information about how to select a set of usability factors or metrics. Hence there is a need for an integrated model that incorporates different viewpoints on usability and defines it in a uniform way. An integrated model must also be generic enough so that developers and experts can use it to measure usability for different kinds of software systems and apply it through all the phases of development.

IV. PROPOSED MODEL

This section gives an integrated model for usability in which the existing models have been unified. In this model, usability of software has been considered to depend upon five attributes, namely, Effectiveness, Efficiency, Satisfaction, Comprehensibility and Safety (Fig. 1). A detailed taxonomy of these attributes and their sub-attributes is given in a structured format.

This model integrates and uniformly presents all the factors and aspects upon which usability has been considered to depend by various researchers. This model is very generic and can be easily applied to all kinds of software systems.

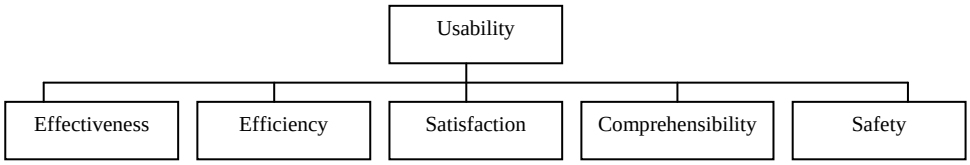

Figure 1. Usability Attributes

A. Effectiveness

It is the degree to which the software facilitates the user in accomplishing the task for which it is intended with precision and completeness while avoiding most errors in varying contexts of use [24].

1) Task Accomplishment: It is the degree to which the software allows the users to perform their tasks and achieve their goals.

- Quantity, the number of tasks that are accomplished correctly.
- Quality, the appropriateness of the task output

2) Operability: It is the degree to which the software provides the users with necessary functionalities that help them to perform tasks correctly [9].

a) Precision: It is defined as the capacity of the software to perform tasks correctly.

b) Completeness: It is defined as the capacity of the software to provide the users with all the necessary functionalities.

3) Universality: It is the extent to which the software can be used by all kinds of users with varying physical or mental characteristics or cultural backgrounds [9] [4].

a) Accessibility: It is defined as the extent to which the software can be used by persons with some type of disability such as visual, auditory, vocal etc.

b) Cultural Universality: It is defined as the extent to which users from different cultural backgrounds can use the software with ease so that the language and cultural conventions (use of symbols, numeric formats, etc.) do not create hindrances.

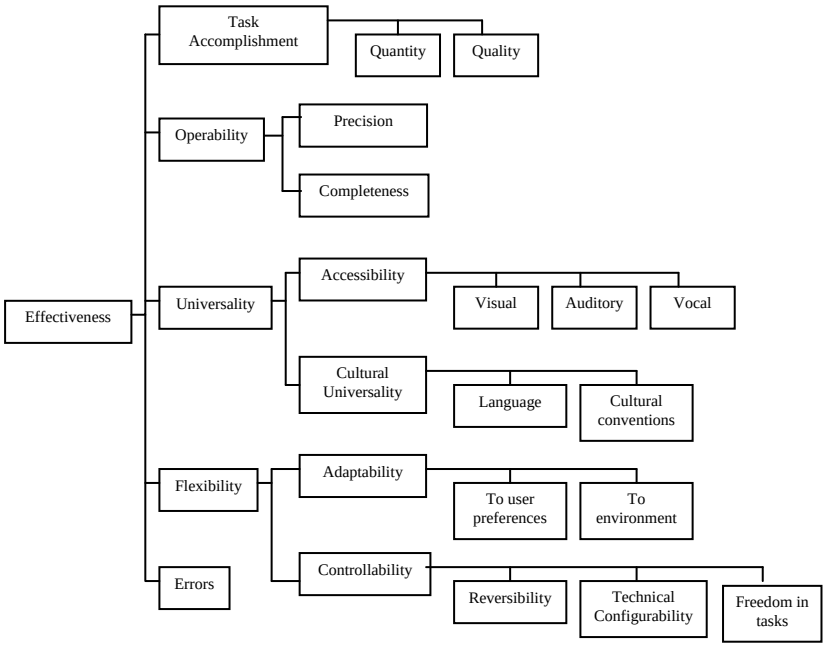

Figure 2. Effectiveness Taxonomy

4) Flexibility: It is the degree to which the software can adapt to changing user needs and preferences [7].

a) Adaptability: It is defined as the extent to which the software can be adapted to user preferences and to different types of environments [9].

b) Controllability: It is the degree to which the software allows the users to mold it according to their personal choice.

- Reversibility, includes commands for reversing actions e.g. undo
- Technical Configurability, e.g. increasing the internal memory at any given point of time.
- Freedom in tasks, one command should not be dependent on another

5) Errors: It is defined as the number of errors produced by the software for a certain number of tasks performed [18].

B. Efficiency

It is measured as the performance of the software in accurately and successfully completing a task in return for the user effort, finances and resources that are invested.

1) User Effort: It is the degree to which the software produces appropriate results in return for the physical and mental effort that a user invests.

2) Finance: It refers to the different types of expenses required for the software.

a) System Costs: It includes the cost of the equipment as well as consumables.

b) Human Resource costs: It includes the costs of the human resource.

3) Resource Utilization: It is the degree to which resources are utilized properly for successfully completing a task. It is measured as the combination of throughput and command utilization.

a) Throughput: It is the quantity of accurate results obtained after investing a certain amount of resources [20].

b) Command Utilization: It is the number of commands that are present but rarely used.

4) Performance: It is defined as the capacity of the software to use minimum possible time and memory for executing a particular task [23].

a) Execution time: It is the time spent in executing a task.

b) Memory Load: It is the amount of memory that is blocked while a task is being executed [35].

c) Decision Complexity: If more than one command are present that give similar results, decision complexity of the software increases.

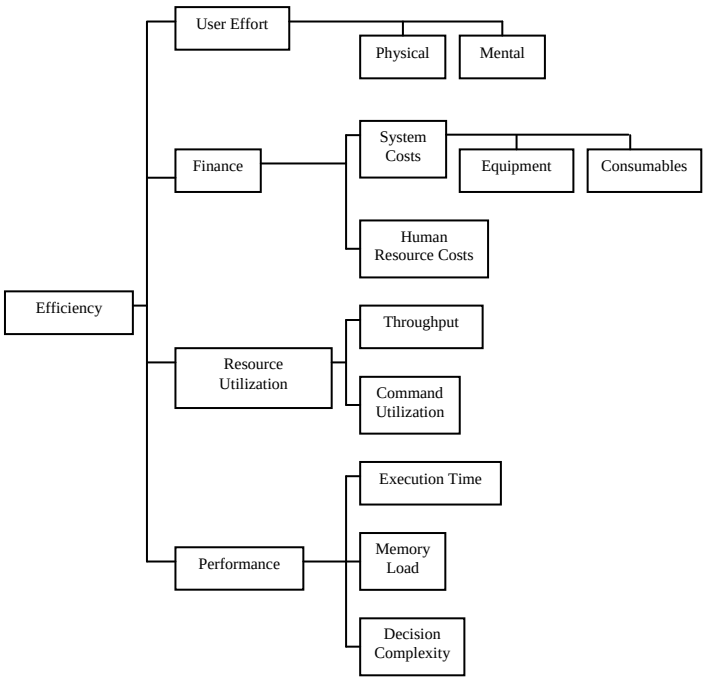

Figure 3. Efficiency Taxonomy

C. Satisfaction

It is defined as the degree to which the software is likeable, comfortable, attractive and trustworthy for the users.

1) Likeability: It is the degree to which the software is liked by the user i.e. the users' perception and opinions of the product are good [34].

2) Trustfulness: It is the faithfulness that the software offers to its users [4].

a) Stability: It is the stability of the software expected by the user.

b) Reputation: It is the reputation of the software in the industry e.g. rank

c) Intention: It is the intention of the user while buying the software i.e. if the user expects before hand that the software will wok properly only for a certain time duration.

3) Comfort: It is the degree to which the software produces a positive feeling/ attitude towards the use of the software and towards its design.

a) Use of Product: It is the response of the user after using the product.

b) Design: It includes the provision of search facilities in the software, how motivating the design is and how chaotic it is.

4) Attractiveness: It is the capacity of the system to be aesthetically pleasing to the user. It can be categorized depending on the type of sensation (visual, tactile and olfactory) [14].

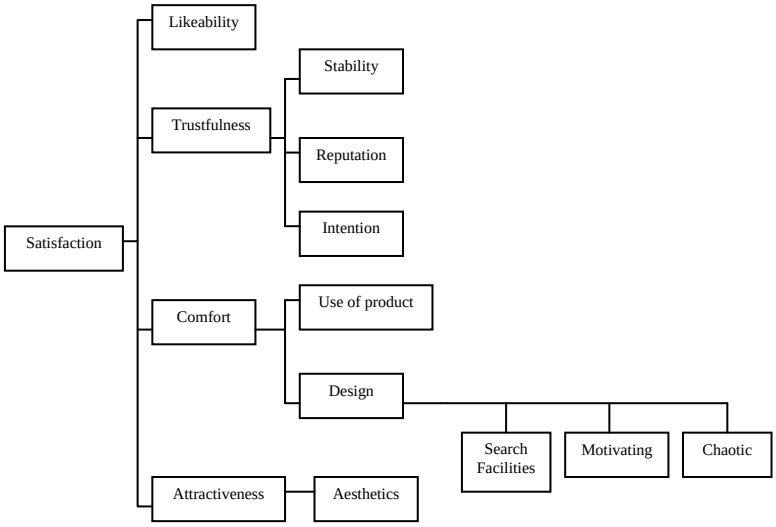

Figure 4. Satisfaction Taxonomy

D. Comprehensibility

It is defined as the degree to which the software has clarity, is easy to learn and remember and includes appropriate help/documentation.

1) Clarity: It is defined as the ease with which the system can be perceived by the mind and senses [9].

a) Clarity of Structure/Elements: It is the property of the system in terms of having its elements organized in a way that enables them to be perceived with clarity (formal) and that their meaning can be easily understood (conceptual).

b) Clarity in functionality: It refers to both the way user tasks are performed and the way system tasks are automatically executed.

2) Learnability: It refers to the degree to which the software is simple and intuitive so as it is easy to learn in minimum amount of time [14].

a) Simplicity: It can be defined as the capacity of the software to be simple i.e. having minimum complexity. E.g. the command names should signify what work they do.

b) Intuitive: It is defined as the degree to which the software can be understood without use of reason/inference.

c) Time to learn: It is the time taken by the user to learn the software.

3) Memorability: It is defined as the property of the system that enables the user to remember the elements and the functionality of the system after a period of time [18]. This attribute like clarity, is *also referred in terms of structure, elements and functioning.*

4) Helpfulness: It is defined as the type and amount of help/documentation provided by the system to help users when they cannot infer or remember how to use the system [9].

a) Suitability of documentation content: The content should be useful and adequate bearing in mind that it includes descriptions of elements and examples showing how to use them.

b) Interactivity of assistance: It is the extent to which the help provided by the software responds to the actions of the users.

c) User Guidance: It is the extent to which the software provides context sensitive help and meaningful feedback when errors occur [4].

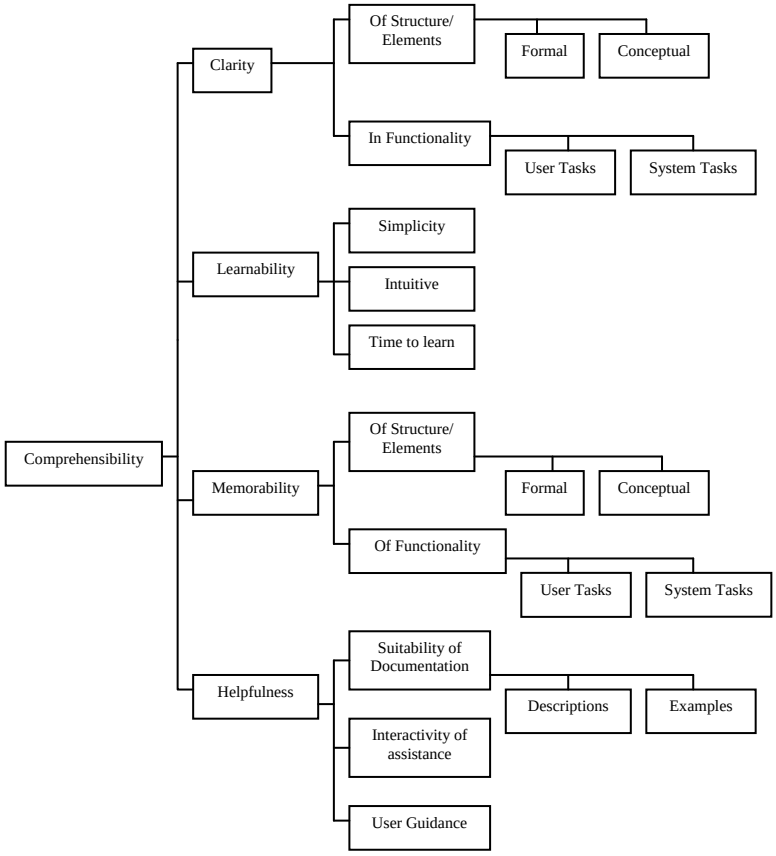

Figure 5. Comprehensibility Sub-attributes

E. Safety

It is defined as the degree to which risk/damage derived from the use of the software can be avoided.

1) User Safety: It is defined as the capacity to avoid risk and damage to the user when the system is in use. Specifying risk or damage in more detail, we distinguish between notions such as physical safety, legal safeguarding, confidentiality, and the safety of the material assets of the user [9].

2) Third Party Safety: It is defined as the capacity of avoiding risk and damage to individuals other than the user when the system is in use [9].

3) Environmental Saftey: It is defined as the capacity of the software to avoid risk/damage from changing environment.

a) Resource safety: It is the extent to which resources are affected by the change in environment.

b) Time between failures: It is the extent to which the environment affects the time between failures of the software. E.g. no air conditioning reduces the time between failures.

c) Hazard prone region: It is the extent to which the region where the software is being used is prone to hazards such as fire, flood, hurricane, etc.

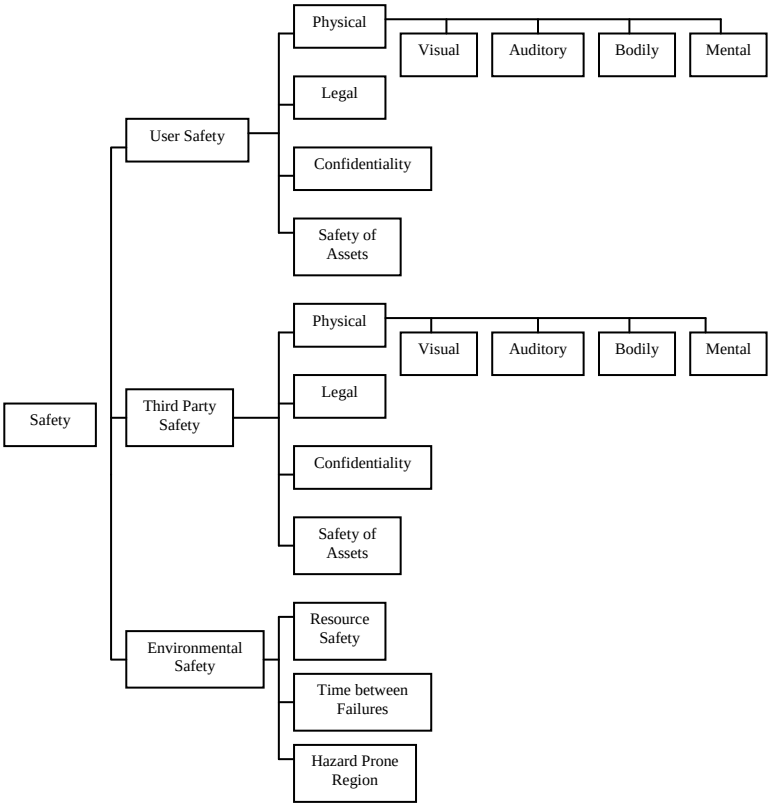

Figure 6. Safety Sub-attributes

V. COMPARISON OF PROPOSED MODEL WITH EXISTING MODELS

A comparison of the proposed model with other existing models is given in Table II. From the table it is clear that this model integrates all the already existing models and covers the concept of usability thoroughly.

TABLE II. COMPARISON OF PROPOSED MODEL WITH EXISTING MODELS

Model	**Usability Attributes**				
Proposed Model	***Effectiveness***	***Efficiency***	***Satisfaction***	***Comprehensibility***	***Safety***
McCall [15]	Operability		Communicativeness	Training	
Boehm [6]	Portability Maintainability				
Shackel [7]	Effectivenss Flexibility		Subjectively Pleasing	Learnability	
FURPS [29]	Consistency	Human factors	Aesthetics	Online and context sensitive help Wizards and agents User Documentation Training Materials	
IEEE Std. 1061 [11]			Communicativeness	Comprehensibility Ease of learning	
Nielsen [18]	Errors	Efficiency	Satisfaction	Learnability Memorability	
Preece *et al*. [19]	Effectiveness	Efficiency	Enjoyableness		Safety
Preece *et al*. [20]	Flexibility	Efficiency Throughput	Attitude	Learnability	
Dix *et al*. [33]	Flexibility Robustness			Learnability	
ISO 9241-11 [12]	Effectiveness	Efficiency	Satisfaction		
ISO 9126-1 [14]	Operability Usability Compliance		Attractiveness	Understandability Learnability	
Donyaee *et al*. [24]	Effectiveness Accessability Universality Usefulness	Efficiency Productivity	Satisfaction Trustfulness	Learnability	Safety
Abran *et al*. [1]	Effectiveness	Efficiency	Satisfaction	Learnability	Security

Model	Usability Attributes				
Proposed Model	***Effectiveness***	***Efficiency***	***Satisfaction***	***Comprehensibility***	***Safety***
Bass *et al*. [23]	Modifiability Scalability Reusability	Performance			Security
Schneiderman *et al*. [8]	Rate of errors by users	Speed of performance	Subjective Satisfaction	Time to learn Retention over time	
Alonso-Rios *et al*. [9]	Operability Robustness	Efficiency	Subjective satisfaction	Knowability	Safety
Dubey *et al*. [36]	Effectivenss	Efficiency	Satisfaction	Learnability	

VI. CONCLUSION

There are various usability models presented in literature. The existing classifications of attributes are divergent i.e. the attributes defined in these models are either vaguely defined or they overlap with each other. When they overlap, they do so only partially, with different terms used to include the same attribute or with the same term used to describe different concepts. To overcome this problem, this paper presented an integrated usability model for software systems. This model describes usability in terms of five attributes, namely, Effectiveness, Efficiency, Satisfaction, Comprehensibility and Safety. A detailed taxonomy of the five attributes is given in a structured format. Exhaustive definitions of all the attributes and their sub- attributes are also given. The taxonomy is built in such a way so as to include all the concepts, factors and attributes that affect the usability of software systems as found out by numerous researchers. As the attributes of usability are of fuzzy nature, a lot of definitions tend to overlap. Therefore, while building the taxonomy special attention was paid to avoiding redundancy as much as possible. In future the authors are going to evaluate the usability of software system in context of the model proposed in this paper.

REFERENCES

[1] A. Abran, A. Khelifi, & W. Suryn, Usability meanings and interpretations in ISO standards. Software Quality Journal, 11, 325–338, 2003

[2] A. Dix, J. Finlay, G. Abowd and R. Beale, Human-computer interaction, Prentice Hall, Hemel Hempstead, UK, 1993.

[3] A. Holzinger, Usability engineering methods for software developers. Communications of the ACM, 48(1), 71–74, January 2005

[4] A. Seffah, M. Donyaee, R. B. Kline and H. K. Padda, Usability measurement and metrics: A consolidated model. Software Quality Journal, 14, 159–178, 2006.

[5] B Shneiderman, Designing the user interface: strategies for effective human computer interaction, Addison-Wesley, USA, 1987

[6] B. Boëhm, Characteristics of software quality, Vol 1 of TRW series on software technology, North-Holland, Amsterdam, Netherlands, 1978.

[7] B. Shackel, Usability – Context, framework, definition, design and evaluation. In Human Factors for Informatics Usability, ed. Brian Shackel and Simon J. Richardson, 21–37. New York, Cambridge University Press, 1991.

[8] B. Shneiderman and C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley, Boston, MA, 2005.

[9] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey and V. Moret-Bonillo, Usability: A Critical Analysis and a Taxonomy, International Journal of Human-Computer Interaction, 26(1), 53–74, 2010.

[10] D. Wixon and C. Wilson, The Usability Engineering Framework for Product Design and Evaluation, in Handbook of Human-Computer Interaction, M.Helander(Ed.), Amsterdam, 1997.

[11] Institute of Electrical and Electronics Engineers. IEEE standard glossary of software engineering terminology, IEEE std. 610.12-1990. Los Alamitos, CA: Author, 1990

[12] International Organization for Standardization. ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability. Geneva, Switzerland: Author, 1998

[13] International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 9126-1:2001, Software engineering, product quality, Part 1: Quality model. Geneva, Switzerland: Author, 2001

[14] ISO 9126: Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for their Use. Geneva, 1991.

[15] J. A. McCall, P. K. Richards and G. F. Walters, Factors in software quality, Vols II, Rome Aid Defence Centre, Italy, 1977.

[16] J. Nielsen and H. Loranger, Prioritizing web usability. Berkeley, CA: New Riders Press, 2006.

[17] J. Nielsen and R. Molich, Teaching user interface design based on usability engineering, 1989.

[18] J. Nielsen, Usability engineering. London: Academic Press, 1993.

[19] J. Preece, D. Benyon, G. Davies, L. Keller and Y. Rogers, A guide to usability: Human factors in computing. Reading, MA: Addison-Wesley, 1993.

[20] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland and T. Carey, Human-computer interaction. Reading, MA: Addison-Wesley, 1994.

[21] J. R. Lewis, IBM Computer usability satisfaction questionnaires: psychometric evaluation and instructions of use, International Journal of Human-Computer Interaction, vol. 7, no. 1, pp. 57–78, 1995.

[22] J. Sauro and E. Kindlund, A method to standarize usability metrics into a single score. Proceedings of the SIGCHI Conference in Human Factors in Computing Systems, 401–409, 2005.

[23] L. Bass and B. E. John, Linking usability to software architecture patterns through general scenarios. Journal of Systems and Software, 66 (3), 187-197, 2003.

[24] M. Donyaee and A. Seffah , QUIM: An Integrated Model for Specifying and Measuring Quality in Use, Eighth IFIP Conference on Human Computer Interaction, Tokyo, Japan, 2001

[25] M. Maguire, Context of use within usability activities. International Journal of Human-Computer Studies, 2001.

[26] M. Porteous, J. Kirakowski and M. Corbett, SUMI User Handbook. Human Factors Research Group, University College Cork, Ireland, 1993

[27] N. Bevan & M. Macleod, Usability measurement in context. Behaviour and Information Technology, 13, 132–145, 1994

[28] N. Bevan, J. Kirakowski & J. Maissel, What is usability? Proceedings of the 4th International Conference on HCI, 651–655, 1991

[29] R. B. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
[30] R. Molich and J. Nielsen, Improving a human-computer dialogue: What designers know about traditional interface design, 1990.
[31] W. Quesenbery, Dimensions of usabilility: Opening the conversation, driving the process. Proceedings of the UPA 2003 Conference, 2003.
[32] W. Quesenbery, What does usability mean: Looking beyond “ease of use.” Proceedings of the 18th Annual Conference Society for Technical Communications, 2001.
[33] A. Dix, J. Finley, G. Abowd and R. Beale, Human-Computer Interaction, 2nd ed. Prentice-Hall, 1998.
[34] J. Rubin, Handbook of Usability Testing, New York: John Wiley.
[35] H. X. Lin, Y. Choong and G. Salvendy, A proposed index of Usability : A method for comparing the relative usability of different software systems usability evaluation methods, behaviour and information technology, Vol. 16, no 4-5, 1997.
[36] S. K. Dubey, A. Sharma and A. Rana, Usability evaluation in object oriented software systems using fuzzy logic approach, International Journal of Computer Science Issues, 2011, “in press”