Review paper on development of interactive user guideline for finite element software

Haqimah Hairum¹, Farzad Hejazi²*, Raizal S. M. Rashid³, M. Saleh Jaafar ⁴

¹Master student of Structural and Construction Engineering, Civil Engineering Department, Faculty of Engineering, Universiti Putra Malaysia
²Associate Professor, Civil Engineering Department, Faculty of Engineering, Universiti Putra Malaysia,
³Head of Civil Engineering Department, Faculty of Engineering, Universiti Putra Malaysia
⁴Dean of Faculty of Engineering, Universiti Putra Malaysia
*farzad@fhejazi.com

Abstract. Due to the rise of diverse finite element software, the needs to understand each of the software interface for optimization of the software usage is increasing among new users. Many researchers had roughly done their software explanation through a few simple methods such as clarifying the software user interface and framework to enhance user benefit with their software. They also provide conclusion for the explanation with a proper justification of an analysis or through interviewing user experiences. The three steps had been proven to be effective in term of giving recognition of software to new users with different background field. This paper will review the standard definition of a graphical user interface that many existing finite element software had completed and how it was exposed to the user. Also, how the software framework function for user to obtain a reliable data out of it and finally the validation of the enlighten framework and GUI for users ease in each model analysis.

1. Introduction
As a result of evolving technologies, there is various type of finite element software available in the market and internet, nowadays [1]. Each types of software had its own specifications and features to help ease the analysis of complex model design in any research.

Due to the rise of diverse finite element software, the needs to understand each of the software interface for optimization of the software usage is increasing among new users. This happened due to the different disciplinary background of user using software from the software designer. User such as engineer or biologist may have disadvantages when conducting finite element software due to different knowledge and skills [2].

In respect to achieve this objective, most current software provides one attribute from the software interface to aid their users. The help key, featuring a compilation of the overall software system is expected to assist user throughout their software experience. Few characteristics may be taken into considerations for any finite element software in creating the help key attribute. First is the recognizing of the software’s user interface. This is an important tool to help develop user awareness for the software in shorter period [3] and enhanced the user continuity with the software. Software/
hardware that have complex user interface usually have limited amount of user that are already expert in it and often fail to attract new user to implement the software/hardware [4,5]. Therefore, to reduce the user efforts in comprehending the interface and sparks interest, the usability rules of the software was implemented to describe the software interface to the user.

Then, a framework or process undergo by finite element software need to be grasp by the user. When user able to understand the concept of the software, they are more likely to obtain results more competent [6] thus, prevent less error in their analysis. A lot of analysis done with finite element software defer from the experimental results obtained, when compare together. This may be due to lack of understanding of the overall software framework [7], thus, causing user to input or used unsuitable operation in the software to conduct the analysis. Lastly, the rationalization of the framework proposed. Many researchers [8,9,10] had provided conclusion for their framework and GUI explanation with a proper justification of an analysis or through interviewing user experiences. The three steps had been proven to be effective in term of giving recognition of software to new users with different background field.

This review paper is an inspirational to newly produce web based software known as ARCS3D. ARCS3D program is a web based software program which users can utilize through computer or mobile internet browser and it’s a free software program with one-time registration. This program uses finite element method for its analysis in reinforced concrete structures [11]. As a newly developed program, ARCS3D is anticipating to provide its user a thorough guideline to fully utilize its features.

This paper will review the standard definition of a graphical user interface that many existing finite element software had completed and how it was exposed to the user. Also, how the software framework function for user to obtain a reliable data out of it and finally the validation of the enlighten framework and GUI for users ease in each model analysis and the best ways of validation can be proved.

2. Defining a graphical user interface as a sandbox

According to a book written by Shneiderman & Plaisant (2005), graphical user interface (GUI) of many software dated back in 2001, follows the standardization of a more common ISO 9241-11:1998 “Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability” standard, which had been revised to ISO 9241-11:2018. The standards describe one of the usability of software was to considered on the satisfaction and effectiveness of user; where a user friendly concept was introduced [13].

Gracia & Bayo (2018) and Li et al. (2017) articles are an example of the agreeable concept of user friendly interface by many software engineers that were also discussed in the previous book mentioned. Both articles talks about the GUI of their respective software application with the help of illustration for improved understanding of their explanation. The step by step explanation was kept simple and this is because of expected background of each end user is usually only in one field type only. Therefore, by doing this, helps to speed up the learning process of the software interface for user [12].

Gasparic, Janes, Ricci, Murphy, & Gurbanov (2017) quoted 12 guideline in explaining human and computer communications layout by Molich & Nielsen (1990) which support the importance of simplicity in explaining the interface. Graphical user interface explained must be spoken to the user level of understanding, in term of technical aspect and also respecting the user memory load in memorizing the software interface. Other small details of laying out user interface (UI) are to keep a consistent method in the UI explanation. All of this are methods used by many software guideline [17,18,19] to keep the understanding of user to the maximum.

A user could be familiar with normal interfaces of any software such as icons or dialogue box. However, to define these features would be hectic if it were not categorised accordingly. Johnsson & Magnusson (2017) and Puerta & Eisenstein (1999) classify these component features as concrete and abstract user interface, where concrete user interface are the solid component such as tools, buttons and boxes or any predefined values. However, abstract user interface are infinity values or feedback
such as user’s name. By following this class could aid in produce a simple explanation of the software interface to the user. This type of classification is suitable as the first step to reach a complete, simple GUI sandbox.

As of stated by Gogoi, Rajkhowa, Choudhury, & Ahmed (2011), an effective GUI explanation and the GUI itself should help the user to visualize the step by step process through the GUI available, and at the same time foresee the result of every action put in each steps. Same as Gogoi et al., (2011) and Zuo, Li, Xu, Xuan, & Na (2012) also believe to enhance the imagination; a labelled picture of each GUI available is a must in the steps. Figure 1 shows an example of labelled steps in a photograph.

As a conclusion, an efficient GUI explanation should be simple to understand by the end user of any software. The explanation should also be interactive with the aid of few images showing instructions and labelled. A lot of researchers find this as the best way to keep the complex explanation of GUI to a minimum level of simplicity.

3. Framework of finite element software
Framework or generally known as operating procedure, sketch in layers, normally is introduce to navigate a user through the specific process of each software design step [25]. Without a properly explained framework, user could interpret the software engine less efficient as compare to what the actual potential of the engine/ application could do.
An example of these possibilities was explained by Xia, Wang, Wang, & Yu (2015) in their article stating that two modules which were found to be unfeasible to work parallel together, could be solved with introduction of proper framework between the two modules. According to them, this can increase the speed of the design and at the same time reduce the repetitive action when using the modules. Figure 2 shows the basic framework produced by them. Beghini et al. (2014) also uses such technique in compiling two and more module as a unified framework and as a general idea for their article. A stand-alone framework for each software researched above is considered efficient, yet, these authors had found a new, novelty method to increase the capability of each software through a layout of formwork. The combined formwork creates a more understandable path for these authors to perform their analysis with more complexity from both software.

![Example of propose framework structure by Xia et al. (2015)](image)

However, in the case of Sofi & Romeo (2017), the framework method helps them in providing a wide range of respond from the software when their model had a missing parameters or uncertain value. They proposed a framework that helps iterate the process needed within a range of value. If the framework was not introduced, the result of the analysis could not be consistent to their predicted result from manual theory calculation.

Human capability are vast and in order to better the potential of each employee in micro, small to medium companies, Sanchez-Gordon, de Amescua, O’Connor, & Larrucea (2017) had make use of the benefit in framework when implemented in daily task of the employee. The purposed of their framework as shown in figure 3 was to enhance the quality and quantity of the product produced by the employee in micro, small to medium companies. García, Pedreira, Piattini, Cerdeira-Pena, & Penabad (2017) had also executed a “Gamification” framework in features to the individual potential.
This had shown the benefit of a framework layout could appear in many aspects. However, the desired profit would be to improve the effectiveness in using a finite element software.

Therefore, a resourceful framework could be achieved by outlining the most basic features in finite element software. This to guarantee the framework drew, covers all possibility the software has. Many framework designers had show that the basic framework was the most effective, first step to be taken. Chakravarty, Hadjesfandia\-ri, & Dargush (2017), Kim & Won (2018) and Ruiz-Rube, Dodero, & Colomo-Palacios (2015) are an example that have demonstrated their framework process starting from the basic or existing framework. After able to distinguish the fundamental area of a framework, then they started to execute their own proposal or exposing deeper on the software’s framework by increasing the layer of the basic framework. The end result of each novel framework would be complicated, yet understandable to many.

Setting deeper in these frameworks, it is noticeable that each frameworks follows a pattern that mentioned by Khoei (2002) where the flow goes from pre-processor, processor and post-processor. This three steps aid in explaining a framework in an organized manner and give a more, uncomplicated understanding to the user. Pre-processor of a software is the basic framework that contain information of the first action taken to initiate the software while post-processor is the final results resulting from the first action done during pre-processor and processor stage. However, processor is not an action control by user and therefore processor usually is a fix step that contains the work done to succeed the pre-processor stage and show the results in post-processor.

As an example from figure 3, the pre-processor is the big circle on the left side where it is the “initiation” stage where each “roles” need to learn specific “techniques” according to their “levels” and “competencies”. The authors [29] describe it as a learning stage. Then, the post-processor is the right side circle which contains the same element as the left side. However, in this stage they were expected to “change management” to facilitate their progress and results in improve and efficient work output. As the keywords for pre- and post- processor for this are “initiation” and “better work output” respectively, the processor stage is the overall work done to achieve the “better work output” results or the “change management” process. This is because, the overall learning “initiation” stage belongs to the pre- group and while better work output is not shown in the framework; we can see the overall picture of changing management of any companies is the action to create greater companies workers.

In the nutshell, framework proven to give advantages to the user in order to fully utilized software. However, explaining a framework could also be confusing if not in organized method. Thus, it is advisable to first layout the basic framework and increases the complexity based on the basic framework to ensure no single layer is left out by following the basic three steps; pre-processor, processor and post-processor.

4. Validation of the proposed GUI/framework

A proposed framework could be used for many purposes while GUI has bigger benefit for many software/hardware. However, the framework and GUI advantages must be validated to ensure that user able to take a full benefit of the layouts and theory discuss for any finite element software which originated from a theory proposed by Hevner, March, Park, & Ram (2004) that express the need to evaluate any problem solving solution produced.
Figure 3. Layers of framework for human task proposed by Sanchez-Gordon et al. (2017)

A framework proving solution by Chen & Lin (2008) and Zhu, McKenna, & Scott (2018) proposed a thorough framework and later discuss on how their framework were usable to conduct their simulation and achieved the results. Still, their validity process only for their own worked analysis and didn’t offer any other users involvement and/or with other case study. However, both articles presented a minimum of two different analyses to be compare to each other either because usage of different platform or codes. Siavvas, Chatzidimitriou, & Symeonidis (2017) also provide validation by doing their own analysis and to strengthen it, they present a comparison with other alternative case study and expert evaluation from other sources available. They stated that the comparison with expert evaluation was more reliable and this is agreeable since they had proven the validity of their own analysis with other case study and the expert evaluation was able to strengthen it.

In term of justification for user interface, Lin, Chu, Wen, Lai, & Chen, (2018) didn’t provide case study but used user involvement in their validation stage through questionnaires. This is because no other approach is more suitable than getting the user themselves to approve it. Shown in figure 4 is an example of data obtained when validating through user experiences for an interactive user interface by Gasparic et al. (2017).

Nevertheless, a framework mainly defines the theoretical value of previous research done or in this case, designs of completed finite element software [36]. Hence, some of the validation progress may not reach the desired results due to many factors such as human error or misconception of the ability of the software as stated by Xia, Wang, Wang, & Yu (2015), where a computing or human error occurred. Nonetheless, the uniformity of their result was still acceptable. Similar with Dominguez, Nigam, & Shahriari (2017), their proposed framework also shows efficiency in certain ways, but they still observed a minor error among their validation works.

A justification of a framework and graphical user interface could be in a two ways method; one is to run own analysis with the software and second is to give out questionnaire to the end user for their experiences. This step is important to observe the efficiency of any framework and GUI delivered.
5. Conclusion

In conclusion, this paper review found that:

- A thorough yet simple graphical user interface definition could enhance the experience of the user with any type finite element software and at the same time speed the understanding of user to the software.

- Framework could be utilizing to reduce the burden of finite element software, lessen the time taken to design in the software and prevent bad results from it. It also can be theoretically helpful for variety of task.

- Great quality of finite element software would have an interactive graphical user interface, understandable framework and a validation of each of the features in their modelling.

References

[1] S. P. Zotkin, N. S. Blokhina, and I. A. Zotkina, “About Development and Verification of Software for Finite Element Analysis of Beam Systems,” *Procedia Eng.*, vol. 111, pp. 902–906, Jan. 2015.

[2] B. A. Johnsson and G. Weibull, “End-User Composition of Graphical User Interfaces for PalCom Systems,” *Procedia Comput. Sci.*, vol. 94, pp. 224–231, Jan. 2016.

[3] S. Yongchareon, C. Liu, X. Zhao, J. Yu, K. Ngamakeur, and J. Xu, “Deriving user interface flow models for artifact-centric business processes,” *Comput. Ind.*, vol. 96, pp. 66–85, Apr. 2018.

[4] S. H. Han, M. H. Yun, J. Kwahk, and S. W. Hong, “Usability of consumer electronic products,” *Int. J. Ind. Ergon.*, vol. 28, no. 3–4, pp. 143–151, Sep. 2001.

[5] S.-W. Hsiao, C.-H. Lee, M.-H. Yang, and R.-Q. Chen, “User interface based on natural interaction design for seniors,” *Comput. Human Behav.*, vol. 75, pp. 147–159, Oct. 2017.

[6] H. R. faragardi, B. Lisper, K. Sandström, and T. Nolte, “A resource efficient framework to run automotive embedded software on multi-core ECUs,” *J. Syst. Softw.*, vol. 139, pp. 64–83, May 2018.

[7] F. García, O. Pedreira, M. Piattini, A. Cerdeira-Pena, and M. Penabad, “A framework for gamification in software engineering,” *J. Syst. Softw.*, vol. 132, pp. 21–40, Oct. 2017.

[8] M. Zhu, F. McKenna, and M. H. Scott, “OpenSeesPy: Python library for the OpenSees finite element framework,” *SoftwareX*, vol. 7, pp. 6–11, 2018.

[9] H.-M. Chen and Y.-C. Lin, “Web-FEM: An internet-based finite-element analysis framework with 3D graphics and parallel computing environment,” *Adv. Eng. Softw.*, vol. 39, no. 1, pp. 55–68, Jan. 2008.
[10] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” vol. 28, no. 1, pp. 75–105, Mar. 2004.

[11] F. Hejazi, M. Shahpasand, M. Mirnezhad, M. S. Jaafar, and A. A. Abang Ali, “A web-based architecture for interactive finite element program,” Comput. Civ. Build. Eng. - Proc. 2014 Int. Conf. Comput. Civ. Build. Eng., vol. 2, no. BV 2003, pp. 1393–1400, 2014.

[12] B. Shneiderman and C. Plaisant, Designing the user interface: strategies for effective human-computer interaction, vol. 215, no. 7. 2005.

[13] L. Cayola and J. A. Macías, “Systematic guidance on usability methods in user-centered software development,” Inf. Softw. Technol., vol. 97, pp. 163–175, May 2018.

[14] J. Gracia and E. Bayo, “An effective and user-friendly web application for the collaborative analysis of steel joints,” Adv. Eng. Softw., vol. 119, pp. 60–67, May 2018.

[15] B. Q. Li, M. L. Xu, X. Wang, S. H. Lu, H. L. Zhai, and J. J. Liu, “THMGUI: A user-friendly Matlab graphical user interface of Tchebichef-Hermite moment method for handling three-dimensional landscapes,” Chemom. Intell. Lab. Syst., vol. 171, pp. 93–101, Dec. 2017.

[16] M. Gasparic, A. Janes, F. Ricci, G. C. Murphy, and T. Gurbanov, “A graphical user interface for presenting integrated development environment command recommendations: Design, evaluation, and implementation,” Inf. Softw. Technol., vol. 92, pp. 236–255, Dec. 2017.

[17] R. Molich and J. Nielsen, “Improving a human-computer dialogue,” Commun. ACM, vol. 33, no. 3, pp. 338–348, Mar. 1990.

[18] “ANSYS Fluent Tutorial Guide.”

[19] “Getting Started with Abaqus: Interactive Edition.”

[20] “SymEd User’s Guide.”

[21] B. A. Johnsson and B. Magnusson, “Towards end-user development of graphical user interfaces for internet of things,” Futur. Gener. Comput. Syst., Oct. 2017.

[22] A. Puerta and J. Eisenstein, “Towards a general computational framework for model-based interface development systems,” Knowledge-Based Syst., vol. 12, no. 8, pp. 433–442, Dec. 1999.

[23] A. Gogoi, P. Rajkhowa, A. Choudhury, and G. A. Ahmed, “Development of TUSCAT: A software for light scattering studies on spherical, spheroidal and cylindrical particles,” J. Quant. Spectrosc. Radiat. Transf., vol. 112, no. 17, pp. 2713–2721, 2011.

[24] W. Zuo, W. Li, T. Xu, S. Xuan, and J. Na, “A complete development process of finite element software for body-in-white structure with semi-rigid beams in.NET framework,” Adv. Eng. Softw., vol. 45, no. 1, pp. 261–271, 2012.

[25] F. Delalondre, C. Smith, and M. S. Shephard, “Collaborative software infrastructure for adaptive multiple model simulation,” Comput. Methods Appl. Mech. Eng., vol. 199, no. 21–22, pp. 1352–1370, 2010.

[26] Z. Xia, Q. Wang, Y. Wang, and C. Yu, “A CAD/CAE incorporate software framework using a unified representation architecture,” Adv. Eng. Softw., vol. 87, pp. 68–85, 2015.

[27] L. L. Beghini, A. Pereira, R. Espinha, I. F. M. Menezes, W. Celes, and G. H. Paulino, “An object-oriented framework for finite element analysis based on a compact topological data structure,” Adv. Eng. Softw., vol. 68, pp. 40–48, Feb. 2014.

[28] A. Sofi and E. Romeo, “A unified response surface framework for the interval and stochastic finite element analysis of structures with uncertain parameters,” Probabilistic Eng. Mech., Jul. 2017.

[29] M.-L. Sanchez-Gordon, A. de Amescua, R. V. O’Connor, and X. Larrucea, “A standard-based framework to integrate software work in small settings,” Comput. Stand. Interfaces, vol. 54, pp. 162–175, Nov. 2017.

[30] S. Chakravarty, A. R. Hadjesfandiari, and G. F. Dargush, “A penalty-based finite element framework for couple stress elasticity,” Finite Elem. Anal. Des., vol. 130, pp. 65–79, Aug. 2017.

[31] J. B. Kim and E. Won, “A software framework for pipelined arithmetic algorithms in field
programmable gate arrays,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 883, pp. 83–89, Mar. 2018.

[32] I. Ruiz-Rube, J. M. Dodero, and R. Colomo-Palacios, “A framework for software process deployment and evaluation,” *Inf. Softw. Technol.*, vol. 59, pp. 205–221, Mar. 2015.

[33] A. R. Khoei, “An integrated software environment for finite element simulation of powder compaction processes,” *J. Mater. Process. Technol.*, vol. 130–131, pp. 168–174, 2002.

[34] M. G. Siavvas, K. C. Chatzidimitriou, and A. L. Symeonidis, “QATCH - An adaptive framework for software product quality assessment,” *Expert Syst. Appl.*, vol. 86, pp. 350–366, Nov. 2017.

[35] Y.-D. Lin, E. T.-H. Chu, C.-L. Wen, Y.-C. Lai, and I.-C. Chen, “Benchmarking handheld graphical user interface: Smoothness quality of experience,” *Comput. Electr. Eng.*, vol. 68, pp. 76–91, May 2018.

[36] M.-Y. Tseng, “Creating a theoretical framework: On the move structure of theoretical framework sections in research articles related to language and linguistics,” *J. English Acad. Purp.*, vol. 33, pp. 82–99, May 2018.

[37] S. Dominguez, N. Nigam, and B. Shahriari, “A combined finite element and Bayesian optimization framework for shape optimization in spectral geometry,” *Comput. Math. with Appl.*, vol. 74, no. 11, pp. 2874–2896, Dec. 2017.