Towards a general framework for evaluating Intelligent Environments methodologies

Aditya Santokhee[a,1], Juan Carlos Augusto[b] and Lindsey Brodie[b]
[a] *R.G. on Development of Intelligent Environments, Department of Computer Science, Middlesex University, Mauritius*
[b] *R.G. on Development of Intelligent Environments, Department of Computer Science, Middlesex University, London, UK*

Abstract. Recent studies reveal that there are different methodologies for developing Intelligent Environments. Thus, it has become essential to scrutinize and evaluate the methodologies to increase our understanding of their strengths, weaknesses and features. However, these concerns have not been the target of recent research efforts. This paper presents an evaluation framework for qualitative evaluation of Intelligent Environment methodologies. It is a step towards standardization of current Intelligent Environments methodologies. The framework has been defined through studying, abstracting and unifying best practices from systems engineering. It is based on a generic life cycle model. As an initial validation, we evaluated the User Centred Intelligent Environment Development Process against the proposed framework. We note that this methodology at its current state presents some limitations which will be addressed in future works.

Keywords. Intelligent Environments methods, systems engineering, evaluation framework, smart irrigation system

1. Introduction

Engineering Intelligent Environments (IEs) is a challenging endeavour due to the dynamic nature of the operational conditions in which these systems have to function. Such conditions include diverse set of hardware devices, complex human computer interactions, unstable resource availability, utilisation of the system in situations that were not originally anticipated and the occurrence of errors that are hard to predict [2]. According to recent studies, IEs are being developed using different methodologies due to lack of any recognised standard one [3, 15, 28]. It is also reported that the methodologies are disconnected from each other and each one of them is focused on solving certain development issues [1]. It would therefore be useful to scrutinize and evaluate IE methodologies to increase our understanding of their strengths, weaknesses and features especially as this has not been a focus within recent research. However, few attempts have been made to develop evaluation frameworks for IE methodologies. Studies have investigated evaluation frameworks for related IE systems such as Pervasive Computing [4, 26], Ambient Assisted Living [25] and Multi-Agent [31]. However, the focus has been mostly on evaluation of the systems rather than on methodologies used to develop them.

[1] Corresponding Author, E-mail: a.santokhee@mdx.ac.mu

In this paper, we attempted to address this research gap by proposing an evaluation framework for IE methodologies. The framework has been defined through studying, abstracting and unifying systems engineering lifecycle models as well as considering best practices from systems engineering (SE) [13]. The rest of the paper is organized as follows: Section 2 discusses the engineering challenges for IEs, Section 3 explains the rationale for choosing systems engineering, Section 4 presents our proposed framework, Section 5 describes application of the User-Centred Intelligent Environments Development Process (UC-IEDP) to build a smart irrigation system [3], Section 6 reports on an initial evaluation of UC-IEDP against the evaluation framework and Section 7 highlights the study limitations. The paper is concluded in Section 8 with some directions for current and future work respectively.

2. Background

IEs emphasize developing systems that incorporate both a smart environment and ambient intelligence and are based on the ubiquitous availability of services that will enhance occupants' experiences [2]. Accordingly, every IE should be able to satisfy certain basic design goals [2]. IE projects are inherently complex; both from a technical and management perspective. Moreover, according to a landmark study carried by [12], the development of intelligent systems poses a number of challenges as a result of increasing complexity. While these systems are often multidisciplinary, they require involvement of diverse stakeholders during the entire systems development process. Therefore, it is imperative to have proper communication and cooperation beyond the limits of individual specialist disciplines. This infers that ensuring a standard system understanding represents a key challenge. Moreover, according to [22], these systems require compliance to numerous quality attributes such as robustness, availability, extensibility, safety, security, timeliness and efficiency. This further reinforces careful management of systems requirements during the systems development process and beyond.

IEs also consist of numerous components or sub-systems interfacing with each other. Only complete control over the interfaces enables successful system integration. The increased interactions and interfaces between the individual components further enhances the design complexity of the overarching system to deliver the emergent effect that each individual component cannot provide on its own. To this end, human machine interfaces are also of particular concerns. In [24], the author asserts that testing alone is not sufficient for these systems due to their complexity level and the consequences of their failure. As a result, integration of suitable verification and validation methods is critical for truly dependable systems. As such, system architecting and engineering tools must be adopted to manage the level of complexity in these types of systems.

3. Why Systems Engineering?

According to the International Council on SE [16], "*Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems ... Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.*" This implies that SE addresses both the system to be developed and the associated project [12]. It is particularly concerned with guiding the systems engineering part in a stepwise way with the goal of delivering a quality system that meets user needs. Traditionally, SE has been extensively used to manage the engineering of highly complex systems such as those used in aerospace, defense and security over the last decades [6, 9, 10, 21, 30]. Often, these projects were carried out by government agencies or large organisations and had strict targets for time, cost and quality [5, 11]. There is also plenty of evidence in the literature regarding the suitability of SE for effectively managing the engineering of complex projects, reducing risks and creating successful systems [5, 7, 16, 18]. In a comprehensive study by [17], it was established that common systems engineering activities contribute towards overall success and technical quality.

However, SE has evolved considerably over the past two decades [29]. This has spawned a number of standards and system life cycle process models such as linear, vee, spiral, waterfall, agile and model based systems engineering. As a result, there is no single model that is accepted worldwide that fits every possible situation [7, 29]. Fortunately, all life cycle models subdivide the system life into a set of basic steps that separate major decision milestones. One such model is the generic life cycle model proposed by [17]. It is representative of the majority of systems that are developed including those containing significant software functionality at component level and is based on the convergence on three major sources: the Department of Defence Acquisition Model (DoD 5000.2), the International model ISO/IEC 15288 and the National Society of Professional Engineers (NSPE) model. Hence, in this study we adopted this generic lifecycle model as basis for our evaluation framework.

4. The proposed evaluation framework

The evaluation framework consists of three main stages as shown on Figure 1. The first two stages cover the developmental part of the life cycle while the third the post development period. The figure also shows the main inputs and outputs of each of the stages: those above the blocks refer to specifications and documentation whereas those below represent the evolution from concept to operational system.

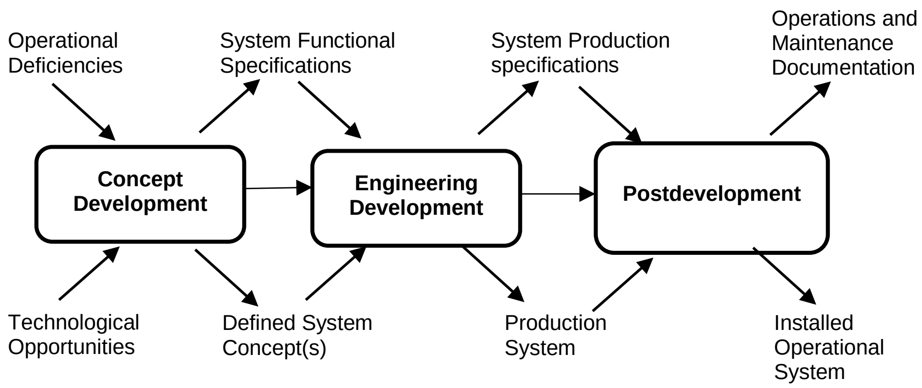

Figure 1: Principle stages in the lifecycle [13]

Concept development stage is the initial stage in the lifecycle. In [17], the authors attest that the decisions made during this stage determine the outcome of the project. The aim of this phase is to formulate and define a system concept that will best satisfy valid user needs. Therefore, the main objectives of the concept development stage are [17]:

- To establish that there are valid needs for a new system that is technically and economically feasible.
- To produce and explore a spectrum of ideas and potential system concepts and formulate a set of system quality requirements. Use of modelling and simulation techniques is highly recommended to inform the decision.
- To select the most relevant system concept, define its functional and quality characteristics, and develop a detailed plan for the subsequent stages of engineering, production, and operational deployment of the system. Decisions are made largely on analyses, simulations and functional designs.

The first activity during the engineering development stage is to identify and minimize development risks. In particular, any new technology that may be required is validated to avoid or mitigate risk of failure during later stages. Detailed system design specifications and test plans for the selected concept are then produced taking into consideration operational, quality, cost and schedule requirements. Following this, the system's functional design is implemented into hardware and software components using a variety of tools and methods. Component testing may take place concurrently. Finally, the components are integrated to produce a production prototype which is evaluated in a realistic operational environment to ensure that it performs as expected and user needs have been met satisfactorily. This is generally followed by formal design reviews. In addition, the end users get the opportunity to provide crucial feedback to the system developers and any implementation deviations are rectified. The postdevelopment stage is mostly concerned with providing support and maintaining effective configuration management strategies after the system has been released for production. Critical bugs or unanticipated system failures are carefully addressed and monitored. Provisions are also made for training users to use the system.

5. Application of the UC-IEDP methodology

The UC-IEDP methodology was used to build a low cost smart irrigation system for small-scale planters in Mauritius. It is a relatively new methodology which has been specifically designed to meet the needs of the IE community and to guide the development of IEs [3]. It consists of three core stages represented as iterative loops: initial scoping, main development and IE installation. It is also guided by an ethical framework which ensures that ethical issues are properly managed. The aim of the project was to provide a more efficient water irrigation system whereby wastage is avoided as much as possible. Existing solutions are too expensive. The project started in October 2018 and is expected to complete by end of April 2019. The system consists of humidity and temperature sensors, a microcontroller, and a solenoid valve as an actuator. Users can also monitor and control the system through a custom-built Android application that connects via Wi-Fi to the microcontroller. A focus group consisting of 3 planters as primary users participated in the development process. They are also the principal stakeholders in the project.

Stakeholders were involved through a variety of activities such as face-to-face interviews, questionnaires, workshops and project pilots. Successive prototypes were generated following feedback from the focus group. A timeline of the focus group engagement activities is given in Table 1. It is to be noted that stakeholders participated regularly in the development process.

Table 1. Timeline of focus group engagement activities

ID	Type of Involvement	Week Number
IT1	Interview	1
QS1	Questionnaire	2-3
WS1	Workshop	4
MT1	Meeting	7
MT2	Meeting	10
WS2	Workshop	14
IT3	Interview	16
QS3	Questionnaire	19
WS3	Workshop	20
QS4	Questionnaire	20-23
WS4	Workshop	24

Another salient feature of UC-IEDP is its flexibility: it may be applied in an iterative manner or using more conventional waterfall based approach. In this project, an iterative approach was favored to allow changes as feedback was received from the stakeholders or requirements changed. This resulted in development of several prototypes as described in the following sections.

5.1. First prototype

Stage 1

Initial scoping: The developer interviewed the planters to know their expectations. Discussions also focused on identifying any specific ethical needs. They were then given a questionnaire to fill in. These activities (IT1, QS1) were instrumental towards defining the services the system should provide based on the problems identified in the questionnaires by the planters. It was quite evident from initial observations that they were moderately at ease with using mobile app technologies. A workshop session (WS1) between the two parties was then held during which the set of services to be implemented was refined and finalized. The developer also provided details of devices that would be required and a consensus was reached based on constraints such as available funds, availability of devices on the local market and time required to develop the system. Following the workshop, the developer then generated an initial list of functional and non-functional requirements. A use case model of the system was also prepared along with low-fidelity mock-ups of the user interfaces for the application.

Stage 2

Main development: The first design was discussed during a meeting (MT1) in week 8. Discussions also focused on clarifying the contexts under which the system will operate. For example, the focus group shed more light on the approximate water required by specific plants for optimal growth and the specific conditions under which this should be automated. Feedback was also collected on the UI mock-ups. Following this meeting, the developer then finalized the list of requirements for the system, and proceeded with more detailed designs using UML artefacts such as class diagrams, activity diagrams and state chart diagrams. The test cases for validating the functional requirements were also defined and a circuit diagram was designed using the Fritzing tool. Android material design was used to refine the initial mock-ups into high fidelity prototype based on

Nielsen usability heuristics. A second meeting (MT2) with stakeholders was then organized during which the prototype was demonstrated and the design document signed off. Consequently, the developer then implemented the system based on the final design document. The mobile application was developed using Android studio. Arduino IDE was used to code the sensing and actuating part as well as linking both the hardware and the custom-built software. The system was tested using a white box testing strategy and uncovered bugs were resolved duly.

Stage 3
IE installation: A second workshop (WK2) was then scheduled in week 14 whereby the implemented system was demonstrated and explained to the focus group. The users then had the opportunity of trying the system. Questions were asked to get their opinions and recommendations to improve the current model. Feedback was generally positive. However, the users suggested having an interface which would display the current humidity and temperature values as well as a report displaying trends of water consumption over a period of time.

5.2. Second prototype

Stage 1
Initial Scoping: Taking cognizance of the recommendations requested by the users, the developer proceeded with a new set of requirements. Then, during an interview session (IT3) in week 16, the users reported on their experience of using the system. They also unanimously requested the ability to use finger print authentication to login into the application. Mockups of the water consumption report was shown and feedback received.

Stage 2
Main Development: Based on feedback received, the developer then updated the requirements specification document and consequently implemented the new functionalities. Fingerprint authentication was added, a system's overview display was added for staying up to date at a glance and new report was implemented.

Stage 3
IE installation: After installation of the improved system, the primary users were allowed to use the system for 3 weeks. They were also given a questionnaire with a set of tasks to perform and to rate the system's efficiency and usability. A workshop (WK4) was scheduled in week 24 to discuss the results of the evaluations with the users. Work is currently ongoing to improve the system based on latest feedback received. We also plan to evaluate the system with a larger group of users in the near future.

6. Evaluating UC-IEDP against the proposed evaluation framework

An initial evaluation of UC-IEDP was then carried out with the evaluation framework described in Section 4. It is clear that there is a strong focus on stakeholder involvement during the entire development process. UC-IEDP is also informed by an ethical framework to embed ethics during development so as to guard against violations of ethical principles. However, its focus is more on the technical side and to a lesser extent on project management. It does not explicitly define project planning and control activities such as work breakdown structure, statement of work and risk management. On the other hand, the evaluation framework addresses both technical and project management aspects equally. Project planning would lead to better coordination of efforts, especially if the project is large, and identification of critical activities. Risk management is particularly important for IEs as it would allow to mitigate potential risks with unproven technologies.

UC-IEDP is flexible enough to accommodate an iterative, linear or incremental system development model. In the project described in Section 5, an iterative approach was adopted. This allowed accepting changes in requirements as the system evolved. The stakeholders were also kept in the loop and their regular feedbacks were particularly instrumental towards informing the development of the irrigation system. A downside to this approach is unavoidable delays if stakeholders are not available. On the other hand, the evaluation framework follows a waterfall like approach. Compared to UC-IEDP, end users experience the prototype system quite late during the engineering stage.

In both approaches, several potential concepts are considered before one is chosen principally by the stakeholders. However, the evaluation framework recommends carrying out modeling and simulations to analyse the behavior of the potential systems to better inform the decision process [20]. This is particularly important for IEs whereby system developers can gain an understanding of how the systems will behave even before they exist. The selected option is then subject to more elaborate designs. Additionally, the evaluation framework advises on defining metrics such as measures of effectiveness which, according to [23], leads to establishment of more meaningful and verifiable system requirements. It also helps to define better testing criteria for the system. However, UC-IEDP methodology does not particularly enforce such requirements and is not explicit on which quality attributes to measure and assess against once the system has been developed. [27] identified that quality attributes have a big impact on the usability of Ambient Assisted Living Systems and this is definitely an area to improve for IEs.

According to Lehman's law of continuing change, any system will undergo continual change [19]. Usage of the system will entail additional functionality even though the system has fulfilled all auditing requirements. Systems are also subject to periodic upgrades or bugs may crop up post production. However, UC-IEDP does not offer any guidance how configuration management should be carried out.

7. Limitations

A known limitation of the presented study is that UC-IEDP methodology was applied to a small project. This was important to get first-hand experience of the development process. Based on lessons learnt, we were able to identify some of the strengths, weaknesses and features of UC-IEDP. Moreover, the impact of the adoption of the evaluation framework was not measured systematically. This is another limitation which we are currently addressing.

8. Conclusion

This study proposes an evaluation framework to scrutinize and evaluate IE methodologies to increase our understanding of their strengths, weaknesses and features.

As an initial validation of the framework, we have evaluated the UC-IEDP methodology by applying it towards development of a smart irrigation system. UC-IEDP in its current format has certain limitations. In particular, it focuses mostly on the technical aspect of project development. We also recommend UC-IEDP methodology to provide guidance to define suitable metrics and quality attributes for IEs. These should be identified and defined more explicitly. They should be monitored and measured during systems development for a more realistic evaluation of IEs. A proper configuration management strategy should also be put in place within the UC-IEDP methodology. We plan to consult researchers who applied UC-IEDP to bigger projects such as Poseidon and validate our initial analyses [3]. Improving UC-IEDP based on the findings from these two projects is in the pipeline. As future work, we plan to improve application of the proposed evaluation framework by developing a software tool and apply the improved UC-IEDP to a new IE project.

References

[1] U. Alegre, J.C Augusto, T. Clark, Engineering context-aware systems and applications: A survey, *Journal of Systems and Software*, Volume 117, 2016, Pages 55-83, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2016.02.010.

[2] J. Augusto, V. Callaghan, A. Kameas, D. Cook, I. Satoh, Intelligent Environments: a manifesto. *Human - centric Computing and Information Sciences*, 3:12, 2013. Springer.

[3] J. Augusto, D. Kramer, U. Alegre, A. Covaci and A. Santokhee, Co-creation of Smart Technology with (and for) People with Special Needs, *Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*, 2016, pp. 39-46

[4] B. Abdulrazak & Y. Malik, Review of Challenges, Requirements, and Approaches of Pervasive Computing System Evaluation, *IETE Technical Review*, 2012, 29:6, 506-522, DOI: 10.4103/0256-4602.105007

[5] B. Blanchard, *Systems Engineering Management,* Third Edition. John Wiley & Sons, 2004.

[6] B. Boehm, Some future trends and implications for systems and software engineering processes. *Systems Engineering*, 9 (1): 1–19, 2006

[7] BKCASE Editorial Board, *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 1.9.1 R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. 2017 Accessed 10th March 2019. www.sebokwiki.org.

[8] R. S. Bruff, *Systems Engineering Best Practices as Measured for Successful Outcomes in Selected United States Defense Industry Aerospace Programs*, Walden University, Dissertation 2008.

[9] J. P. Elm, D. Goldenson, K. El Emam, N. Donatelli, and A. Neisa, *A Survey of Systems Engineering Effectiveness - Initial Results*, Software Engineering Institute, Pittsburgh, PA, USA, Dissertation 2008.

[10] J. P. Elm and D. R. Goldenson, *A Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey*, Software Engineering Institute, Pittsburgh, Report 2012.

[11] H. Eisner, *Essentials of Project and Systems Engineering Management,* Second Edition. Wiley, 2002, Chapter 1.

[12] J. Gausemeier, R. Dumitrescu, D. Steffen, A. Czaja, O. Wiederkehr, C. Tschirner, *Systems Engineering in industrial practice*, Paderborn, 2015

[13] S. Hesari, H. Mashayekhi, R. Raman, Towards a General Framework for evaluating Software Development Methodologies. *Proc. IEEE 34th Annual Computer Software and Applications Conference*, 2010

[14] E. Honour, Systems engineering and complexity. *INSIGHT-INCOSE Journal*, 11 (1): 20–21, 2008

[15] M. Hornos, Application of Software Engineering techniques to improve the reliability of Intelligent Environments. *Journal of Reliable Intelligent Environments*. 3. 1-3. 10.1007/s40860-017-0043-0, 2017

[16] International Council on Systems Engineering, *A Guide for System Life Cycle Processes and Activities*, Version 3.2, July 2010.

[17] A. Kossiakoff, W. Sweet, S.J. Seymour, & S.M. Biemer, *Systems Engineering: Principles and Practice*. Second Edition ed. New Jersey: Wiley, 2011

[18] J. K. K. Kludze, Jr., *Engineering of Complex Systems: The Impact of Systems Engineering at NASA*, George Washington University, Dissertation 2003.

[19] M. M. Lehman, Programs, life cycles, and laws of software evolution, in *Proceedings of the IEEE*, vol. 68, no. 9, pp. 1060-1076, Sept. 1980. doi: 10.1109/PROC.1980.11805

[20] A. Murad, J.C Augusto, and D. Windridge, A survey of user-centred approaches for smart home transfer learning and new user home automation adaptation. *Applied Artificial Intelligence*. ISSN 1087-6545. 2019 (Accepted/In press), https://doi.org/10.1080/08839514.2019.1603784

[21] NASA: NASA – Systems Engineering Handbook, NASA/SP-2007-6105 Rev1, 200

[22] E. Ras, M. Becker, J. Koch, Engineering tele-health solutions in the ambient assisted living lab. *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*; Niagara Falls, ON, Canada. 21–23 May 2007; pp. 804–809.

[23] N.B Reilly, *Successful Systems Engineering for Engineers and Managers* , Chapman and Hall, 1993, ISBN 0-442-01586-0

[24] J. Rushby, Formal methods and critical systems in the real world. *Formal Methods for Trustworthy Computer Systems (FM89)*, pp.121-125, 1989

[25] D. Salvi, C. Montalvá, J. Arredondo, M. Prazak-Aram, B. Mayer, Christopher, A framework for evaluating Ambient Assisted Living technologies and the experience of the universAAL project. *Journal of Ambient Intelligence and Smart Environments*. 7. 329-352. 10.3233/AIS-150317, 2015

[26] Y. Malik, M. Soliman, B. Abdulrazak, Towards an Evaluation Framework for Pervasive Computing System. 8, 2011

[27] M. Mukhtiar et al, Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. *Sensors (Basel, Switzerland)* vol. 14,3 4312-41. 4 Mar. 2014, doi:10.3390/s140304312

[28] A**.** Santokhee, J.C Augusto, and C. Evans, Engineering intelligent environments: preliminary findings of a systematic review. In: 2nd Workshop on Citizen Centric Smart Cities Solutions (CCSCS'18), *The 14th International Conference on Intelligent Environments*, 25-28 June 2018, Rome, Italy.

[29] S.A Sheard, Evolution of systems engineering scholarship from 2000 to 2015, with particular emphasis on software, *Systems Engineering*, **21**, 3, (152-171), 2018.

[30] R. Stevens, P. Brook, K. Jackson, and S. Arnold, *Systems Engineering, Coping with Complexity*. Prentice Hall, 1988.

[31] Q. Tran, G. Low, M. Williams, A Feature Analysis Framework for Evaluating Multi-agent System Development Methodologies*. Lecture Notes in Computer Science*. 10.1007/978-3-540-39592-8_87. 2004