Evaluating the Effectiveness of Problem Frames for Contextual Modeling of Cyber-Physical Systems: a Tool Suite with Adaptive User Interfaces

Waqas Junaid*
School of Computer Science and Engineering, Guangxi Normal University, Guilin, Guangxi 541004, P. R. China
waqasjunaid1@gmail.com

ABSTRACT

Bridging the gap between academic research and industrial application is an important issue to promote Jackson's Problem Frames approach (PF) to the software engineering community. Various attempts have been made to tackle this problem, such as defining formal semantics of PF for software development, and providing a semi-formal approach to model transformations of problem diagrams, with automated tool support. In this paper, we propose to exclusively focus on exploring and evaluating the effectiveness of Jackson's problem diagrams for modeling the context of cyber-physical systems, by developing a suite of support tools enhanced with adaptive user interfaces, and empirically and comprehensively assess its usability. This paper introduces the state of the art, corresponding research questions, research methodologies and current progress of our research.

CCS CONCEPTS

• **General and reference**; • **Requirements engineering tools and techniques**; • **Evaluation**;

KEYWORDS

Problem Frames, cyber-physical systems, context-based knowledge, adaptive user interfaces

ACM Reference Format:

Waqas Junaid. 2021. Evaluating the Effectiveness of Problem Frames for Contextual Modeling of Cyber-Physical Systems: a Tool Suite with Adaptive User Interfaces. In *Evaluation and Assessment in Software Engineering (EASE 2021), June 21–23, 2021, Trondheim, Norway*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3463274.3463344

1 INTRODUCTION

Jackson's Problem Frames approach [1] (PF) provides one of the mainstream methodologies in software requirements engineering [2-5], and recently for modeling the contexts of cyber-physical systems (CPS) [6-7], as an alternative to narrative approaches (e.g., use cases [8] or scenarios [9]), and goal-oriented approaches (e.g., i* [10] or KAOS [11]) to requirements engineering. Although many researchers have developed tool support for PF, to the best of our knowledge, there have not been enough empirical studies to evaluate the effectiveness of PF for modeling the contexts of CPS and supporting automated reasoning based on PF models (i.e., problem diagrams with associated textual descriptions).

In order to tackle the above problem, we propose a three-phase research agenda, which will be elaborated in Section 3. The long-term goal of this research approach is to bridge the gap between academic research and industrial application of PF. For example, due to the openness and dynamic changes of CPS architectures, PF is facing several difficulties and challenges in applying PF in practice, such as conceptual confusion, lack of industry-oriented guidance, and scalability problem, which is similar to goal-oriented requirements engineering [16]. Furthermore, as pointed out by Leveson in her ICSE2020 keynote speech [17], global quality requirements such as safety and security are even more difficult to handle due to the contextual complexity (e.g., degree of cohesion) inherent in modern CPS. Our focus on evaluating CPS's problem diagram model is an attempt to capture complex and heterogeneous CPS architectures and represent them as visually explicit knowledge in the form of problem diagrams, so as to make PF as widely adaptable to practice as possible.

The remainder of the paper is structured as follows. In section 2, we review and discuss related work and introduce our research objective in Section 3. Section 4 explains selected methods and proposed solutions addressing the research objectives. In Section 5, we give a short conclusion and discuss the current progress of our work.

2 RELATED WORK

For the past decade, Problem Frames have been used and extended in many directions. For instance, Choppy et al [18] have looked into the problem of deriving software architectures from Jackson's basic frames; Bleistein et al [19] and Cox et al [20] have looked at modeling business problems with PF; Hatebur et al [21] have applied Problem Frames in the context of security engineering, Strunk [22] in safety engineering and Zhang et al [23] in process control systems; while [24–26] have looked at meta-models and ontology for Problem Frames with a view to tool development. This is the rich context to which our work intends to add.

*PhD supervisor's name: Zhi Li, corresponding author, zhili@gxnu.edu.cn, School of Computer Science and Engineering, Guangxi Normal University, Guilin, Guangxi 541004, P. R, China.

EASE 2021, June 21–23, 2021, Trondheim, Norway

ACM ISBN 978-1-4503-9053-8/21/06…$15.00
https://doi.org/10.1145/3463274.3463344

Hall et al [37] have provided a set-based de-notational semantics of problem frames for software development as shown below.

$$c,\ o\ : [K,\ R] = \{S\ : Specification \mid S\ controls\ c\ \wedge S\ observes\ o\ \wedge K,\ S \vdash_{DRDL} R\}$$

Later, Hall and Rapanotti have proposed a reference model for requirements engineering [38]. Li et al [39] have attempted to provide fully formal techniques based on Hoare's CSP language, and later defined an operational semantics based on causal reasoning [31], as shown below.

$$c,\ o\ : [K,\ R] = \{S \mid S! = c\ \wedge S? = o\ \wedge K,\ S\ sat\ R\}$$
$$= \{(S_1\ \cup\ S_2\ \cup \ldots \cup\ S_n) \mid (S_1\ ||\ S_2\ \ ||\ldots||\ S_n\ )sat\ R\}$$

where each of S_1, S_2,. . .,S_n represents a process which consists of a trace of causal chain, identified by domain experts.

The above formula gives an insight into a possible solution to the semantic challenge in the original de-notational semantics, i.e., decomposing a complex problem into many sub-problems guided by a causal chain will reduce the complexity of the problem. This is the theoretical basis on which our research proposal is based.

In recent years, many researchers have been focusing their research attempts on extending and adapting theories of PF to CPS, with relevant tool support (see [27] for online versions of these tools). Chen et al have proposed an ontology-guided requirement modeling technique for PF [28] and provided a tool for guiding problem description and projection [29]; Li et al [14] have proposed a semi-formal approach to problem progression based on graph grammar and provided a suite of relevant tool support [30-31]. Chen et al [32] have extended PF with timing requirements with associated support tool TimePF [33]. Yu et al have extended PF with security requirements and developed OpenArgue [35] to support arguing that the software is secure during its evolution. However, an empirical method of evaluating the effectiveness of PF in modeling the contexts of CPS is missing from the literature.

Recently, there has been a trend in improving software engineering tools with adaptive model-driven user interfaces [15], which solves the problem of software applications being very large with hundreds of complex user interfaces, by providing a Role-Based UI Simplification (RBUIS) mechanism to increase their usability based on the context-of-use. RBUIS uses an interpreted runtime model-driven approach based on the Cedar Architecture, and is supported by the IDE - Cedar Studio. Wang et al [36] focus on the scalability issue in laying out large-scale goal models and propose a two-level layout approach to automatically lay out the models in an efficient and comprehensible manner. The second phase of our proposal shares similar motivation with that of Wang et al [36], i.e., tool usability is a key contributing factor for industrial acceptance and adoption of a method and its tool support.

3 RESEARCH OBJECTIVES

We propose the following research agenda, which consists of three phases: firstly, an operational semantics is defined to give a precise meaning of problem diagrams to guide and support the process of modeling, automated analysis and reasoning about the context of CPS (i.e., "statement of domain knowledge" in Zave and Jackson's words [40]), which is essential for bridging the gap between its software specifications and user requirements [40]; secondly, a suite of support tools with adaptive model-driven user interfaces ([15]) is proposed to embody the principles of the operational semantics defined in phase one, but tailored to support many different aspects in domain knowledge elicitation, modeling and analysis (e.g., clarifying visual representation with de-cluttering by auto-layout, reducing model complexity with problem projection and progression by automated decomposition, refining the requirements and contexts by embedded graph layering, etc); thirdly, we apply Yin's participant-observer case studies, to empirically evaluate the effectiveness of PF in a series of controlled experiment, and we further apply several Human-Computer Interaction (HCI) research methods to empirically assess the usability of the tool suite in a practical setting. With help of findings in phase 3, we test our hypothesis that both foundational work on semantics and technical tool support play different but equally important roles in supporting software development practice. Here is a list of research questions which corresponds to the three phases above.

RQ1: *How should the semantics of PF be defined so that problem complexity can be defined, measured, reduced and usefully reasoned through model transformation or machine learning?*

RQ1.1: *How does a statists-based approach to semantics of PF differ from a logic-based approach in terms of defining, measuring, and reducing problem complexity?*

Current research on PF is lacking in checking, measuring and evaluating the complexity of a problem, therefore the research gap is to be filled by giving definitions of problem complexity based on the degree of coupling of problem contexts. Answers to this question will deliver a systematic method for evaluating the global complexity of a CPS so that design efforts can be recommended for software design, implementation and maintenance. There have been several works on the logic-based approach to PF semantics, but work on the statistics-based approach is currently missing so we are using both approaches to check which one is better representation of problem complexity.

RQ1.2*: Since work on evaluating problem complexity of CPS is missing from the literature, how can we develop a strategy and UI that can allow us to "zoom in" and "zoom out" of a problem diagram so that problem complexity can be measured at any levels of modelling granularity?*

RQ2: *What difference will a tool suit based on semantics make in requirements engineering practice (including lab environment, controlled experiment or realistic development)?*

RQ2.1: *To what extend the "DFS" (i.e., problem decomposition) is applicable to realistic software development practice?*

We are using the DFS (Depth-First Search) and BFS (Breadth-First Search) for automatic problem decomposition based on causal chains identified by domain experts. The answer to this research question will involve automated decomposition of PF models into several sub-problems, which is claimed to be easier to solve than the bigger problem.

RQ2.2: *Is an adaptive user interface suitable for implementing the auto-layout algorithm?*

To answer this question, we will need to develop the tool support with various adaptive UI techniques, e.g., auto-layout, material design.

RQ3: *To what extend is Jackson PF effective for modeling complex contexts for CPS?*

The answer to this question involves usability studies, with eye tracking being used for empirical evaluation.

4 PROPOSED METHODS AND SOLUTIONS

Compared to conventional information systems, cyber-physical systems involves the modeling of dynamic and open environment whose complexity should be treated as first-class citizen in requirements analysis, design and implementation. A systematic approach to reasoning about such complexity, and further reducing and managing it remains an open problem.

This section focuses on the proposed methods and solutions for the research questions explained earlier. It consists of three phases:

Phase one contains the answer or solution to RQ1 which is mostly concerned with the semantics of PF. We are going to fill the gap asked in **RQ1.1**. The statistics-based approach to the semantics of PF is compared to the transformational approach [7] as shown in Figure 1. The current researches are lacking in measuring, checking and evaluating the complexity of PF. These gaps will be filled by our research. The main benefit of solving this problem is that researchers will have clear understanding of which approach provides better support for evaluating the complexity of PF. After solving the complexity problem, we need to "zoom in" and "zoom out" to capture the complexity of application domain for answering **RQ1.2**. It is necessary to check the complexity at each level of modeling granularity. The main advantage is to obtain in-depth knowledge of the modeling. The more details are shown, the more work is needed to reason about interactions among them. It can also help us reveal problem complexity at different dimensions.

Phase Two. There has been some work on problem decomposition in the literature, but there has not been any work on the DFS as a problem decomposition strategy for PF. We need to implement the searching algorithm to answer **RQ2.1**. We are going to improve the tool so that cause-and-effect relationships can be set by a drop-down list. We are going to improve the user experience by auto-layout algorithms, and automatic decomposition based on DFS algorithm, as shown in Figure 1

Based on achieving these research objectives we need to build tool support. The tool support is necessary for further evaluation. The tool should be built in such a way that graphical user interfaces of the software can be used easily with minimum efforts. The user interface is one of the essential elements in term of using software. So, the tool built by [7] is not quite interactive, boring and we need to add some more features to it. Although we already did some improvement to the tool, but in order to answer **RQ2.2**, as in [15] the adaptive user interface techniques such as feature set minimization and optimal layout can be used as a reference to improve the user experience. Furthermore, the Eclipse Sirius and ELK auto-layout tools can also be used to make the user interface as desired as possible.

Phase Three. We plan to adopt Yin's participant-observer case study by developing questionnaires, and use controlled experiments to evaluate usability of the tool suite.

The most important thing about the user interface is efficiency. The Yin's participants-observer will be used as a case study method for a possible solution to **RQ3**. It would be used to check the efficiency based on the experimental results. Our evaluation will check whether we have built some useful support or not. Furthermore, for checking the effectiveness of the software, eye tracking techniques will be used.

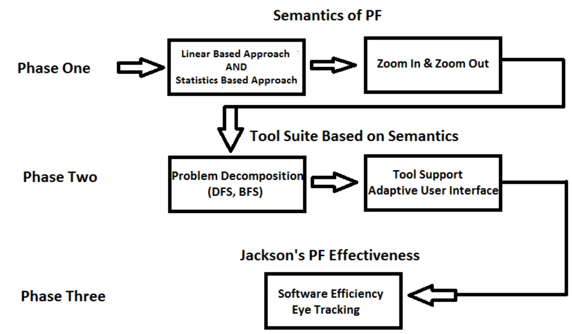

Figure 1: Work Plan.

5 PROGRESS, CONCLUSIONS AND FUTURE WORK

In this paper, we propose a rigorous approach to defining the complexity of PF modeling, based on existing work PF semantics, and improve existing PF support tools by adding more features such as adaptive user interfaces, and finally, evaluate our approach and support tools performing usability studies. We choose adaptive user interfaces techniques because we want to re-use existing work developed by many researchers, rather than following a user-centered design and developing a completely new tool suite. We believe this choice better suits the research agenda of a PhD. Both the theoretical part of defining problem complexity (RQ1), and the technical part of tool development (RQ2) are tightly coupled as the main contribution of the PhD work, while the empirical evaluation part (RQ3) may consist of controlled lab experiments (to be carried out by the PhD researcher, with students samples on eye-tracking), and usability studies by professional software developers (questionnaires will be collected after giving out tutorials and the tool suite).

In terms of progress, literature reviews on problem semantics and complexity is completed and a statistical model have been conceived, which will be submitted soon. Various existing prototypes have been studied, whose source codes will be re-factored and ported onto a new platform, with added adaptive user interfaces. Empirical evaluation is currently considered, to be designed and planned in future work.

ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (61862009), Guangxi Natural Science Foundation (2018GXNSFAA281314), Guangxi "Bagui Scholar" Teams for Innovation and Research, the Project of the Guangxi Key Lab of Multi-source Information Mining & Security (Director's grant 19-A-01-02), Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, the Innovation Projects of Guangxi Graduate Education (No. XYCSZ2017066).

REFERENCES

[1] M. Jackson. 2001. Problem Frames: Analyzing and Structuring Software Development Problems. Addison-Wesley Publishing Company.

[2] J. Hall, L. Rapanotti and M. Jackson. 2008. Problem Oriented Software Engineering: Solving the Package Router Control Problem. in *IEEE Transactions on Software Engineering*, vol. 34, no. 2, pp. 226-241, March-April 2008, doi: 10.1109/TSE.2007.70769.

[3] L. Rapanotti, J.G. Hall and K. Cox. 2004. 1st International Workshop on Advances and Applications of Problem Frames, in Proceedings. 26th International Conference on Software Engineering (ICSE2004), pp.754-755, Edinburgh, UK, 23-28 May 2004. doi: 10.1109/ICSE.2004.10010

[4] J.G. Hall, L. Rapanotti, K. Cox, and Z. Jin. 2006. 2nd international workshop on advances and applications of problem frames, in Proceedings. 27th International Conference on Software Engineering (ICSE2006), Shanghai, China, pp.1030-1031. May 20-28, 2006. doi: 10.1145/1134285.1134488

[5] T.T. Tun, J.G. Hall, L. Rapanotti and K. Cox. 3rd international workshop on advances and applications of problem frames, 30th International Conference on Software Engineering (ICSE2008), Leipzig, Germany, Companion Volume, pp.1029-1030, May 10-18, 2008, doi: 10.1145/1370175.1370231

[6] M. Jackson. 2015. System Behaviours and Problem Frames: Concepts, Concerns and the Role of Formalisms in the Development of Cyber-physical Systems. Dependable Software Systems Engineering, 79-104, 2015.

[7] Z. Li, J.G. Hall, and L. Rapanotti. 2014. On the systematic transformation of requirements to specifications. Requirements Engineering, volume 19 issue 4 pages 397-419, Springer London, 2014.

[8] Alistar Cockburn. 2000. Writing Effective Use Cases. Addison-Wesley.

[9] Ian F. Alexander and Neil Maiden, editors. 2004. Scenarios, Stories, Use Cases Through the Systems Development Life-Cycle. John Wiley and Sons, Ltd.

[10] Eric Yu. 1997. Towards modelling and reasoning support for early-phase requirements engineering. In Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97), pages 226–235, Washington D.C, USA.

[11] van Lamsweerde, R. Darimont, P. Massonet. 1995. Goal-directed elaboration of requirements for a meeting scheduler: problems and lessons learned. In: Proceedings of the 2nd international symposium on requirements engineering (RE'95), York, Canada

[12] Yuanyang Wang et, al. 2015. Xiaohong Chen, Ling Yin. TimePF: A Tool for Modeling and Verifying Timing Requirements based on Problem Frames. APRES 2015, pp.149-154.

[13] Xiaohong Chen et, al. 2010. DPTool: A Tool for guiding the problem description and the problem projection. 18th IEEE International Requirements Engineering Conference, pp. 401-402.

[14] Zhi Li et. al. 2013. On the systematic transformation of requirements to specifications. Requirement Engineering Journal (REJ). DOI: 10.1007/s00766-013-0173-8.

[15] Pierre A. Akiki et. al. 2015. Engineering Adaptive Model-Driven User Interfaces. IEEE Transition on Software Engineering (TSE). DOI: 10.1109/TSE.2016.2553035.

[16] Mavin A, Wilkinson P, Teuf S, Femmer H, Eckhardt J, Mund J (2017) Does goal-oriented requirements engineering achieve its goal? In: 2017 IEEE 25th international requirements engineering conference (RE), IEEE, pp 174–183

[17] N.G. Leveson. 2020. Everything You "Know" About Software and Safety is Probably Wrong, 2020 ACM/IEEE 42nd International Conference on Software Engineering (ICSE2020), Seoul, South Korea, July 27-19, 2020.

[18] C. Choppy, D. Hatebur and M. Heisel. 2005. Architectural patterns for problem frames. IEE Proceedings-Software, 152(4):198–208

[19] S. Bleistein, K. Cox, J. Verner. 2004. Requirements engineering for e-Business systems: integrating Jackson context diagrams with goal modelling and BPM. In: Proceedings of the 11th international Asia-Pacific software engineering conference (APSEC 2004), pp 410–417. IEEE, Busan, Korea. 30th November– 3rd December 2004

[20] K. Cox, J.G. Hall, L. Rapanotti. 2005. Editorial: a roadmap of problem frames research. Information and Software Technology 47(14):891–902

[21] D. Hatebur, M. Heisel, H. Schmidt. 2006. Security engineering using problem frames. In: Emerging trends in information and communication security, lecture notes in computer science, vol 3995/2006. Springer, New York, pp 238–253

[22] E.A. Strunk, J.C. Knight. 2006. The essential synthesis of problem frames and assurance cases. In: Proceedings of the 2006 international workshop on advances and applications of problem frames (IWAAPF '06). ACM Press, New York, pp 81–86

[23] X. Zhang, G. Auriol, C. Baron. 2010. Deriving specification for process-control systems: a problem frames approach. In: 5th International conference on systems (ICONS). IEEE Press, New York, pp 60–65

[24] D. Hatebur D, M. Heisel, H. Schmidt. 2008. A formal metamodel for problem frames. In: Model driven engineering languages and systems, lecture notes in computer science, vol 5301/2008. Springer, New York, pp 68–82

[25] X. Chen, Z. Jin, L. Yi. 2007. An ontology of problem frames for guiding problem frame specification. In: Knowledge schience, engineering and management, lecture notes in computer science, vol 4798/2007, pp 384–395

[26] L. Lavazza, A. Coen-Porisini, V. Del Bianco. 2009. Towards a meta-model for problem frames: conceptual issues and tool building support. In: 4th International conference on software engineering advances (ICSEA '09). IEEE Press, New York, pp 339–345

[27] Website: http://www.re4cps.org/home

[28] X-H. Chen, and Z. Jin. 2008. An Ontology-guided Process for Developing Problem Frame Specification: An Example. Proceedings of the 3rd International Workshop on Applications and Advances of Problem Frames, IWAAPF'08 , pp. 36-39, 2008.

[29] X-H. Chen, B. Yin, and Z. Jin. 2010. DPTool: A Tool for Guiding the Problem Description and the Problem Projection. 18th IEEE International Requirements Engineering Conference, 2010, pp. 401-402.

[30] G-Y. Liu, Z. Li, S-L. Huang, Z-F. Ouyang and Z. Liu. 2015. CARE: A Computer-Aided Requirements Engineering Tool for Problem-Oriented Software Development , International Journal of Software Engineering and Knowledge Engineering, 25(09n10):1747-1752, December 2015, doi:10.1142/S0218194015710114.

[31] Z-Y. Zhao, Z. Li, C-L. Fu, and X. Zhang. 2017. A Computer-aided Modeling and Verification Approach for Problem-oriented Software Development. Z.Chaczko, F.Pichler, F.Jiang and C.Chiu (Eds.):In Proceedings of 5th Asia-Pacific Conference on Computer Aided System Engineering(APCASE2017), pp. 21-28, Guilin, Guangxi, China, November 2017.

[32] X-H. Chen, L. Yin, Y-J. Yu, Z. Jin. 2017. Transforming Timing Requirements into CCSL Constraints to Verify Cyber-Physical Systems. ICFEM 2017: 54-70

[33] Y-Y. Wang, X-H. Chen, L. Yin. 2015. TimePF: A Tool for Modeling and Verifying Timing Requirements based on Problem Frames. APRES 2015, pp.149-154, 2015.

[34] Y. Yu, N. Yoshioka and T. Tamai. 2019. Assessing Security and Privacy Behavioural Risks for Self-Protection Systems. Engineering Adaptive Software Systems 2019: 135-147

[35] Y. Yu, T. Tun, A. Tedeschi, V. Franqueira, B.Nuseibeh. 2011. OpenArgue:Supporting argumentation to evolve secure software systems. 2011 IEEE International Requirements Engineering Conference (RE 2011): 351-352

[36] Y-D. Wang, T. Li, Q-X. Zhou and J-L. Du. 2021. Toward practical adoption of i* framework: an automatic two-level layout approach. Requirements Engineering, published online on January 28, 2021.

[37] Hall JG, Rapanotti L, Jackson MA (2005) Problem frame semantics for software development. Software and Systems Modeling 4(2):189–198.

[38] J. Hall *et al.* 2003. A reference model for requirements engineering. IEEE International Requirements Engineering Conference 2003.

[39] Z. Li, J.G. Hall and L. Rapanotti. 2011. On the construction of specifications from requirements, 14th Workshop on Requirements Engineering (WER2011), pages 431-442, Rio de Janeiro, Brazil, April 2011

[40] P. Zave and M. Jackson. 1997. Four dark corners of requirements engineering, ACM Transactions on Software Engineering and Methodology, 6(1):1-30, January 1997.