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:

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.
We propose the following research agenda, which consists of three ware specifications and user requirements [40]; secondly, a suite with relevant tool support (see [27] for online versions of these where each of S\_CPS (i.e., “statement of domain knowledge” in Zave and Jackson’s modeling, automated analysis and reasoning about the context of 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])

\[ c, o : [K, R] = \{S : \text{Specification} | S \text{ controls } c \land S \text{ observes } o \land K, S + \text{R} + \text{K} \}\]

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.

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

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., identifying 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 statistics-based approach to semantics of PF differ from a logic-based approach in terms of defining, measuring, and reducing 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 extent the “DFS” (i.e., problem decomposition) is applicable to realistic software development practice?

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.

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).