Safety Science 132 (2020) 104944

Contents lists available at ScienceDirect

Safety Science

journal homepage: www.elsevier.com/locate/safety

Extended SAFPHR (Systems Analysis for Formal Pharmaceutical Human Reliability): Two approaches based on extended CREAM and a comparative analysis

Xi Zheng[a], Matthew L. Bolton[a,*], Christopher Daly[b]

[a] University at Buffalo, State University of New York, Department of Industrial and Systems Engineering, Buffalo, NY, USA
[b] University at Buffalo, State University of New York, Department of Pharmacy Practice, Buffalo, NY, USA

ARTICLE INFO

Keywords:
Human reliability analysis (HRA)
Extended cognitive reliability and error analysis method (CREAM)
Medication errors
Formal methods
Probabilistic model checking

ABSTRACT

Medication errors originating in community pharmacies can cause severe harm. To provide pharmacies with the ability to accurately predict error rates, understand why errors are occurring, and mitigate problems, we developed a new human reliability analysis (HRA) called the Systems Analysis for Formal Pharmaceutical Human Reliability (SAFPHR). Through the combination of HRA that is based on the Cognitive Reliability and Error Analysis Method (CREAM) and probabilistic model checking (an automated method for proving properties about stochastic systems), SAFPHR is able to address the limitations of previous HRAs. The previous, "basic" version of SAFPHR was based on the "basic" version of CREAM. With this, we predicted a realistic range of medication error rates for a typical US community pharmacy dispensing procedure. However, basic SAFPHR was not capable of providing point estimates except through averaging. In this research, we attempted to address this limitation by making SAFPHR compatible with the two variations of extended CREAM, enabling SAFPHR to make point predictions about pharmacy dispensing error rates. Then, to determine which of the versions of SAFPHR produce the most accurate predictions, we compare results from each approach to aggregate rates published in the community pharmacy literature. In this, arithmetic averages across basic SAFPHR's range were consistently the most accurate for the overall error rate and rates of errors originating at different stages of the dispensing procedure. We use this finding to derive recommendations from basic SAFPHR for improving the reliability of community pharmacy dispensing with the ambition of improving patient health and safety.

1. Introduction

Based on the two contemporary, comprehensive studies on community pharmacies that are primarily responsible for processing prescriptions, between 0.057% and 11% of filled prescriptions have dispensing errors (Szeinbach et al., 2007; Odukoya et al., 2015). Americans filled 3.8 billion prescriptions at retail pharmacies in 2018 (BlueShield, 2019). Thus, even with the minimum error rate of 0.057%, there are 216.6 million errors every year. Even a small error rate in prescription dispensing can have serious consequences on patients' lives and health. As such, preventing medication errors in community pharmacy has been a patient safety goal of the joint commission for many years (Parker, 2013; The Joint Commission, 2014; The Joint Commission, 2015; The Joint Commission, 2016a; The Joint Commission, 2016b).

Unfortunately, medication errors from community pharmacies are complex and not well understood. Procedures can vary from pharmacy to pharmacy; the working environments change dynamically over the course of the day; errors can occur in different stages; and, because all reporting systems are voluntary, statistics on errors are under-reported (Ashcroft et al., 2006; Allan and Barker, 1990; Wilson et al., 1998; Kaushal et al., 2001; Pape, 2001). These factors together with the reality that almost all community pharmacies are private organizations that are not required to share information about procedures, make it difficult to get comprehensive and consistent data. Some observational and experimental studies have gathered error data and identified the major causes of errors (Flynn et al., 2003; Berdot et al., 2013; Lao et al., 2016). These techniques are useful, but require significant time and effort and they are incapable of considering all of the complex system interactions that could impact reliability (Zheng et al., 2020). Thus, model-based approaches like human reliability analysis (HRA), which can predict human error rates, are appropriate for application in

* Corresponding author.
E-mail address: mbolton@buffalo.edu (M.L. Bolton).

https://doi.org/10.1016/j.ssci.2020.104944
Received 9 April 2020; Received in revised form 26 June 2020; Accepted 2 August 2020
Available online 14 October 2020

addressing medication errors coming from community pharmacies.

HRAs are techniques used to predict human error probabilities by assessing the effect sociotechnical factors have on human performance (Hollnagel, 1998a; Swain and Guttmann, 1983). While they have been used successfully in a number of safety–critical domains, they are static and do not account for how system dynamics can impact error rates. They also do not consider interactions between errors (Zheng et al., 2017).

Thus, we have created a novel, model-based HRA called the Systems Analysis for Formal Pharmaceutical Human Reliability (SAFPHR), an approach that is capable of capturing how medications move through a pharmacy, all while accounting for the types of errors, the potential contributing factors, and the dynamism that impacts these factors (Zheng et al., 2020). By combining the Cognitive Reliability and Error Analysis Method (CREAM) (a well-validated approach to HRA) with probabilistic model checking using PRISM, SAFPHR is able to address the major shortcomings of previous HRAs, while accounting for interaction between errors and dynamic system behaviors and considering all of the possible paths through a modeled system.

Our previous version of SAFPHR (henceforth basic SAFPHR; (Zheng et al., 2020)) was based on the basic version of CREAM, which enabled us to predict a valid range of medication error rates of a typical pharmacy dispensing procedure. However, basic SAFPHR could not produce precise point estimates without averaging over the large ranges necessitated by basic CREAM.

Thus, in this work, we make SAFPHR compatible with the two variations of extended CREAM to get precise predictions without the need for averaging. Then, we apply the two variations of extended SAFPHR to predict the overall error rate for the common community pharmacy dispensing procedure. We compare the predictions made with each basic and extended version of SAFPHR to published error rates. To determine which of these methods is most appropriate for use in community pharmacy, we conducted a validation study to compare the different point estimates that can be produced from all of the different versions of SAFPHR to determine which best identified where errors originate in the dispensing procedure. In the remainder of the document, we first introduce the concepts of HRAs, CREAM, probabilistic model checking, and basic SAFPHR. We then detail the steps for implementing and applying two variants of extended SAFPHR to the analysis of a typical community pharmacy dispensing procedure. After this, we provide the details of our validation study: predicting the relative error rates with different versions of SAFPHR and comparing our results against rates published in an aggregate and comprehensive community pharmacies study (Flynn et al., 2003). Finally, we discuss our results along with directions for future research.

2. Background

2.1. HRA

Even with advances in automation and autonomy, the safety of industrial systems is still dependent upon human operations. Therefore, many HRAs have been developed in order to qualitatively and quantitatively assess the human contribution to risk (De Felice et al., 2012). HRAs can generally be divided into two generations dependent on whether the methods are based on probabilistic risk assessment (PRA) or cognition respectively (Di Pasquale et al., 2013).

First-generation HRAs [i.e. Technique for Human Error Rate Prediction (THERP) (Swain and Guttmann, 1983; Swain, 1987), the Human Error Assessment and Reduction Technique (HEART) (Williams, 1986; Williams, 1988), Human Cognitive Reliability (HCR) (Hannaman et al., 1984) and the Operator Action Tree (OAT) (Wreathall, 1982)] have been heavily influenced by PRA, where human errors are modeled as if they are equipment failures. Thus, the nominal human error probability can be assigned based on the characteristics of the operator's task and then modified by performance shaping factors (PSF) such as time pressure, equipment design, and stress (Boring et al., 2006; Bell and Holroyd, 2009; Di Pasquale et al., 2013). While useful, first-generation HRAs are often criticized for failing to consider things like the impact of context, organizational factors, and errors of commission (Hollnagel, 1998a; Di Pasquale et al., 2013).

Second-generation HRAs improve on these by considering interactions between human operators, production processes, the organisation, and the environment and how they impact models of human cognition (Hollnagel, 1998a; Bye et al., 1999; Kim and Jung, 2003; Fujita and Hollnagel, 2004; Kim et al., 2006; Reer, 2008; Lee et al., 2011; Di Pasquale et al., 2013; Zheng et al., 2017). CREAM is largely considered the leading second-generation method (Bell and Holroyd, 2009). It is introduced in the next section.

2.2. CREAM

CREAM (Hollnagel, 1998a) improves on first-generation methods by grounding its approach in cognitive theory via the Contextual Control Model (COCOM) (Hollnagel, 1998b). It posits that human performance is determined more by the situation in which a task is performed than by inherent properties of the task itself (Zheng et al., 2017). In CREAM, nine sociotechnical factors called Common Performance Conditions (CPCs; Table 1) are used to describe the criteria that influence human performance (Hollnagel, 1998a). Human error probabilities are calculated based on assessments of these CPCs.

Table 1
CREAM CPCs, adapted from (Zheng et al., 2017; Hollnagel, 1998a).

CPC	Description
Organization	The roles and responsibilities of team members and the quality of additional support, communication systems, safety management, instructions, guidelines, and oversight.
Conditions	Physical working conditions such as ambient lighting, glare on screens, noise from alarms, and interruptions.
Support	Man–machine interface quality, including information on control panels, computer workstations, and operational support from decision aids.
Procedures	Availability and quality of operating and emergency procedures, familiar routines, and response heuristics.
Goals	The number of goals/tasks a person is required to pursue or attend to concurrently.
Available Time	The time available to carry out a task and how well the task execution is synchronized with process dynamics.
Time of Day	Whether the person is adjusted to the current time.
Experience	The quality of operator training and level of experience.
Collaboration	The quality of crew collaboration.

2.2.1. Basic CREAM

There are three versions of CREAM that provide different levels of quantitative analysis: one version of basic CREAM and two versions of extended CREAM (Hollnagel, 1998a). The first step in basic CREAM (which the other two build off of) is to describe the task sequences. Then, for each task, CPCs (Table 1) are assessed subjectively by a subject matter expert to determine whether the conditions associated with each CPC improve human task performance, reduce it, or are not significant (two CPCs, Goals and Time of Day, can only be not significant and reduced).[1] Moreover, to account for the dependency between CPCs, adjustments on four CPCs (Conditions, Available Time, Goals, and Collaboration) are required based on other assessed CPC values (Hollnagel, 1998a). This process is described in Fig. 1.

After adjustments, for each task, the number of CPCs assessed as improved and the number assessed as reduced are counted. Finally, these counts map to one of four COCOM cognitive control modes (Hollnagel, 1998a; Hollnagel, 1998b) (Fig. 2). The four basic control modes describe different levels of control people have over the work they are doing based on the environment (Bedford et al., 2013). In *Scrambled* control, the human chooses actions randomly with little or no thinking due to the loss of situation awareness. *Opportunistic* control occurs when a human in a situation he or she is familiar with but has no formal plan to follow. Thus, actions are chosen inefficiently. *Tactical* control encapsulates situations where a human plans and executes actions by following known rules or procedures. Finally, *Strategic* control occurs when the human has a deep knowledge of the system and can thus plan for a number of different situations. As such, actions are chosen after the human fully considers the situation. Human performance (i.e. the probability of human error) ranges from worst (higher probability) to best (lower probability) from scrambled (worst), to opportunistic, to tactical, to strategic (best) (Hollnagel, 1998a). Fig. 2 shows how each control mode maps to CREAM-specified ranges of error probability.

2.2.2. Extended CREAM

The intervals predicted by basic CREAM can be too large to be practically useful (Hollnagel, 1998a). To address this, there are two variations of extended CREAM that can produce precise point estimation of human error rates (Hollnagel, 1998a; He et al., 2008). In both versions, analysts must identify the primary cognitive function of each task as well as the task's most likely cognitive function failure (CFF; the way that the cognitive function can fail to accomplish its goal). CREAM's cognitive function and CFFs are listed in Table 2. Each CFF has an associated "nominal" probability of occurrence, its CFP (cognitive function probability). In the first approach to extended CREAM, these probabilities are modified by multiplying them by a scaling factor (*S* in Fig. 2) that is associated with the control mode, where the control mode is determined using the same process as basic CREAM. In the second (and presumably more accurate) approach, the analyst does not use the CPCs to compute a control mode. Instead, the nominal probabilities are adjusted directly based on how specific CPCs impact the associated cognitive activities (Hollnagel, 1998a). In this approach, the probability for each CFF is modified by multiplying it against weighting factors associated with each assessed CPC level for the given cognitive function (Table 3). For example, assume you are assessing a task with an "Observation" cognitive function and an "Observation Not Made" (O3) CFF. This corresponds to a CFP of 0.07 (Table 2). Then, to calculate the probability of this observation not being made, you would take the assessed values of the CPCs; adjust them using the method employed by basic CREAM (Fig. 1); look up the value for each CPC's level in the observation column from Table 3; and multiply these together with the 0.07 CFP.

Note that in the standard practice of CREAM, a single analyst with good general knowledge of the field should be able to apply it (Hollnagel, 1998a). Thus, an analyst with adequate related experiences will be in a good position to construct the event sequence and assess the various CPCs and CFFs.

CREAM has been used successfully in nuclear power applications (Hollnagel et al., 1999), radiation therapy (Castiglia et al., 2008), food manufacturing (Geng et al., 2015), oil tanker shipping (Zhou et al., 2018, 2017, 2018), gas network (Desmorat et al., 2013), and hospital pharmacies (Rantanen et al., 2012b; Rantanen et al., 2012a). Despite these successes, CREAM has the same limitations as other second-generation methods. This means that it does not account for sources of system errors beyond humans. Additionally, CREAM's predictions are static in that they do not account for how rates will change dynamically as a system evolves. In developing SAFPHR, we attempted to fix these issues by combining HRA and probabilistic model checking.

2.3. Formal methods and probabilistic model checking

Formal method is a broad area of study that is concerned with formal verification (Wing, 1990): proving properties against mathematical models of systems. Formal methods have been successfully used to evaluate erroneous human behavior in complex systems to discover specific unsafe system conditions (Bolton et al., 2013; Weyers et al., 2017; Bolton, 2017). However, non-stochastic models are used in these methods: making them inadequate for assessing human reliability.

Probabilistic model checking is a software-based formal verification technique that allows analysts to automatically prove properties about models of dynamic systems (Kwiatkowska et al., 2007). Mathematical languages describe the behavior of a system using a stochastic model (e.g., variants of Markov chains), specification properties describe desirable system properties or request the probability of a system condition occurring, and verification either proves the specification property or accurately computes the probability requested in it. Thus, probabilistic model checking accounts for all modeled system components, interactions, and dynamism when computing probabilities. This means that probabilistic model checking can address the limitations of traditional HRAs by accounting for dynamic system changes and interactions between humans and other errors in a system (Zheng, 2020).

2.4. Basic SAFPHR

To addresses the limits of previous HRAs and allow analysts to accurately predict pharmacy errors while accounting for system dynamism and non-human source of errors, we developed basic SAFPHR (Zheng et al., 2020) by combining concepts from the basic CREAM with probabilistic model checking using PRISM (the world's leading open-source probabilistic model checker; Kwiatkowska et al., 2011).

We applied basic SAFPHR to the analysis of a typical community pharmacy procedure and obtained a valid range of medication error rate predictions. In particular, we modeled a common dispensing procedure (shown later in Fig. 3) along with CPC assessments from a subject matter expert, all while accounting for the dynamism associated with different time periods (described later in Fig. 4) as well as non-human sources of error (such as prescriptions arriving at the pharmacy with validity, legality, appropriateness, and safety issues). With basic SAFPHR, we predicted that between 1.02422E-03% and 2.8856292% of prescriptions would reach patients with an error, which were in line with the published range of 0.057% to 11% (Szeinbach et al., 2007; Odukoya et al., 2015). However, because basic SAFPHR is derived from basic CREAM, the large range of predicted errors may not allow analysts to make accurate recommendations. Further, while averaging can be used to obtain point estimates from ranges, it isn't clear what approach (i.e. arithmetic vs. geometric) will yield the most accurate predictions.

[1] Note that CPCs for each task are assessed using standardized questionnaires, where the actual assessed levels ultimately translate into a CPC level of improved, not significant, or reduced (see Table 3).

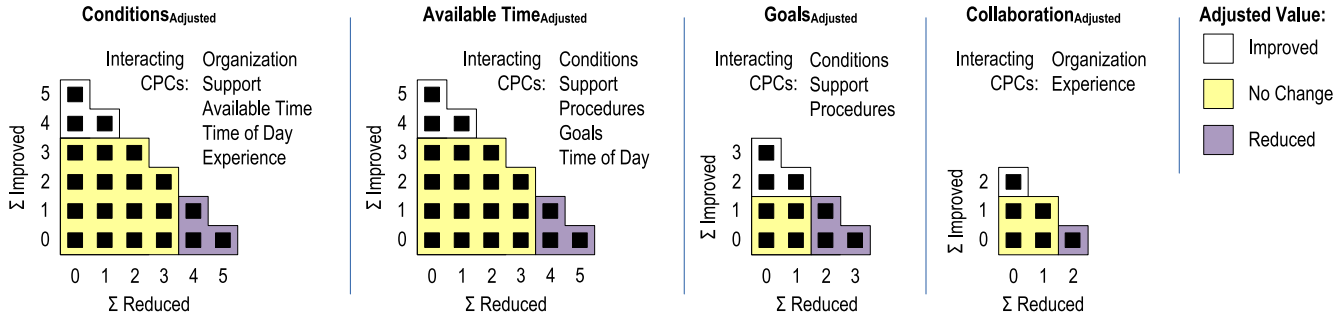

Fig. 1. Graphs illustrating the method CREAM uses to adjust CPC values to account for their dependencies (Hollnagel, 1998a). Each graph represents a CPC that is adjusted based on a list of the other CPCs on which it is dependent. The adjusted value of a CPC is calculated based on the number of the dependent CPCs that are improved (Σ Improved) and reduced (Σ Reduced). A pair of sums corresponds to a point on the graph that falls within a region. The region indicates if the adjusted value of the CPC is improved, reduced, or unchanged. Reproduced from (Zheng et al., 2020).

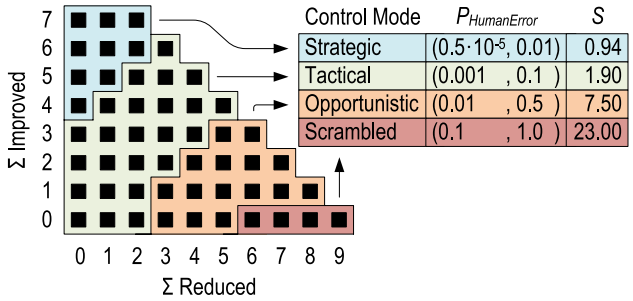

Fig. 2. The method for converting CPC values (post adjustment) into COCOM control modes in CREAM. The number of the CPCs rated as improving (Σ Improved) and reducing (Σ Reduced) human performance are mapped to control modes. These then map to ranges of human error probabilities ($P_{HumanError}$) and scaling factors (S). Adapted from (Zheng et al., 2020).

Table 2
Extended CREAM Cognitive Function Failures (CFFs) and Nominal Probabilities (CFPs) (adapted from Hollnagel, 1998a).

Function	CFF	CFP
Observation	O1: Wrong Object Observed	0.001
	O2: Wrong Identification	0.07
	O3: Observation Not Made	0.07
Interpretation	I1: Faulty Diagnosis	0.2
	I2: Decision Error	0.01
	I3: Delayed Interpretation	0.01
Planning	P1: Priority Error	0.01
	P2: Inadequate Plan	0.01
Execution	E1: Action of Wrong Type	0.003
	E2: Action of Wrong Time	0.003
	E3: Action on Wrong Object	0.0005
	E4: Action Out of Sequence	0.003
	E5: Missed Action	0.03

3. Objectives

In this work, we first attempt to address the shortcomings of basic SAFPHR (Zheng et al., 2020) by making it capable of producing point estimates. We accomplish this by extending it with the two variations of extended CREAM. The use of extended CREAM can potentially produce accurate point estimates of error rates, thus avoiding averaging across the large ranges produced by the basic methods. Based on the two variations of extended CREAM, we have developed two versions of SAFPHR: mode-effect extended SAFPHR and CPC-effect extended SAFPHR. Mode-effect extended SAFPHR uses the version of extended CREAM where the scaling factor associated with each control mode (S from Fig. 2) is used to modify nominal probabilities associated with cognitive function failures. CPC-effect extended SAFPHR uses the other approach to extended CREAM, where the scaling factor is dependent on the individual effects of specific CPCs on specific cognitive functions (Table 3). We will describe how we modified basic SAFPHR to create mode-effect extended SAFPHR and CPC-effect extended SAFPHR. We then use these new versions of SAFPHR to predict the overall error rate of the typical community pharmacy dispensing procedure and compare these predictions to those obtained with basic SAFPHR as well as rates found in the literature.

With fully development versions of both basic and extended SAFPHR, we set out to determine which approach was the most accurate and valid. Specifically, these three versions of SAFPHR can collectively produce six different methods of computing error rates: basic SAFPHR's upper bound (U), basic SAFPHR's lower bound (L), basic SAFPHR's arithmetic mean (A) of its upper and lower bounds, basic SAFPHR's geometric mean (G) of these same bounds, mode-effect extended SAFPHR's (MEE) point estimate, and CPC-effect extended SAFPHR's (CEE) point estimate. We describe how we apply each approach to SAFPHR to predict the error rates for different stages of the typical community pharmacy dispensing procedure. Finally, we compare these predictions to error rate estimates published by Flynn et al. (2003), who presents both overall error rates as well as rates of errors originating from different parts of dispensing. These comparisons allow us to assess how accurate our predictions are both in aggregate and for specific tasks. Based on the results of the validation, we use SAFPHR to make recommendations for improving community pharmacy dispensing.

4. Extended SAFPHR

The application of extended SAFPHR generally follows the same procedure outlined for basic SAFPHR (Zheng et al., 2020). The analyst first constructs the procedure (sequence of tasks) pharmacists use for achieving system goals. He or she must also identify which CPCs (Table 1) are static and which are dynamic. Static CPCs represent factors that are completely dependent on the tasks of the procedure, dynamic CPCs are variable based on other, dynamic environmental criteria. The analyst then assesses the values of the CPCs. For static CPCs, CPCs must be assessed for each task from the procedure. For dynamic CPCs, each is assessed at all possible levels of the system's dynamic elements. Because we were interested in evaluating a typical pharmacy, we used the same community pharmacy dispensing procedure constructed for (Zheng et al., 2020) (Fig. 3) and the same assessment of CPCs employed in basic SAFPHR, which were assessed by the project's subject matter expert Dr. Daly. [2]For our analysis, Goals, Available Time,

[2] A full listing of the survey and its results can found at http://fhsl.eng.buffalo.edu/SAFPHR/

Table 3
CREAM Multiplication Factors Used to Adjust Nominal Failure Probabilities (Table 2) Based on assessed CPC Levels (Table 1) (Hollnagel, 1998a).

			Cognitive Function			
CPC	Assessed Level	CPC Level	Observation	Interpretation	Planning	Execution
Organization	Very Efficient	Improved	1.0	1.0	0.8	0.8
	Efficient	Not Significant	1.0	1.0	1.0	1.0
	Inefficient	Not Significant	1.0	1.0	1.2	1.2
	Deficient	Reduced	1.0	1.0	2.0	2.0
Conditions	Advantageous	Improved	0.8	0.8	1.0	0.8
	Compatible	Not Significant	1.0	1.0	1.0	1.0
	Incompatible	Reduced	2.0	2.0	1.0	2.0
Support	Improve	Improved	0.5	1.0	1.0	0.5
	No Impact	Not Significant	1.0	1.0	1.0	1.0
	Tolerable	Not Significant	1.0	1.0	1.0	1.0
	Reduce	Reduced	5.0	1.0	1.0	5.0
Procedures	Improve	Improved	0.8	1.0	0.5	0.8
	No Impact	Not Significant	1.0	1.0	1.0	1.0
	Reduce	Reduced	2.0	1.0	5.0	2.0
Goals	Below Capacity	Not Significant	1.0	1.0	1.0	1.0
	Matching Capacity	Not Significant	1.0	1.0	1.0	1.0
	Over Capacity	Reduced	2.0	2.0	5.0	2.0
Time	Adequate	Improved	0.5	0.5	0.5	0.5
	Reduced	Not Significant	1.0	1.0	1.0	1.0
	Inadequate	Reduced	5.0	5.0	5.0	5.0
Time of Day	Adjusted	Not Significant	1.0	1.0	1.0	1.0
	Unadjusted	Reduced	1.2	1.2	1.2	1.2
Experience	Improve	Improved	0.8	0.5	0.5	0.8
	No Impact	Not Significant	1.0	1.0	1.0	1.0
	Reduce	Reduced	2.0	5.0	5.0	2.0
Collaboration	Very Efficient	Improved	0.5	0.5	0.5	0.5
	Efficient	Not Significant	1.0	1.0	1.0	1.0
	Inefficient	Not Significant	1.0	1.0	1.0	1.0
	Deficient	Reduced	2.0	2.0	2.0	5.0

Note. Value colors show how they impact the probability of error. Black indicates no change, Blue a decrease, and Red an increase.

and Time of Day are dynamic because they will vary based on when a prescription is being dispensed and the number of other tasks happening at the time. This leaves Organization, Conditions, Support, Procedures, Experience, and Collaboration as static (a deeper discussion of the difference between static and dynamic CPCs can be found in (Zheng et al., 2020)). When assessing static CPCs for each task, the procedure in Fig. 3 was presented to analysts to help them understand what they were assessing. Similarly, the distribution of prescriptions with percentages shown in Fig. 4 was used as a reference during dynamic CPC assessment. We collected the CPCs assessments by asking our subject matter expert to fill out a CREAM survey that accounted for the methodological differences introduced by SAFPHR while using specific language familiar to community pharmacists. It is important to note that the procedure evaluated, the static-dynamic CPC distribution, and CPC level assessment can all be customized to account for the specifics of an individual pharmacy (Zheng et al., 2020).

In both versions of extended SAFPHR, analysts are then required to identify the primary cognitive function of each task as well as the task's most likely CFF (Table 2). This work was also completed by the project's subject matter expert. [3].With these tasks completed, an analyst uses a systematic process to convert these assessments into a formal PRISM model and runs analyses. In what follows, we describe how this is achieved while using our modeled procedure and assessments as an example.

4.1. The formal modeling architecture

To enable analyst to translate a dispensing procedure model, CPC assessments, and CFP assessments into a formal model, we modified the formal modeling architecture from basic SAFPHR to accommodate the new extended versions. An overview of this architecture can be found in Fig. 5.

In this, the task that is executing at a given time is encapsulated by the procedure sub-model. This also controls the order in which tasks are performed based on the analyzed procedure model (e.g. Fig. 3). The environmental dynamism sub-model is used to represent dynamic factors in the environment that can impact the CPCs that are not specifically connected to human operator's procedure task. In our model, this represents the time of day (Fig. 4), which impacts the dynamic CPCs associated with time and workload (Goals, Available Time, and Time of Day). For a given procedure task, formulas map the state of the procedure and environmental dynamism submodels to the associated static and dynamic CPC values. Another formula then adjusts the CPC values in accordance with Fig. 1. An additional formula then uses principles from extended CREAM to compute a probability of error. When the procedure is indicating the performance of a non-human task (for

[3] The results of assessment on cognitive functions and cognitive function failures can found at http://fhsl.eng.buffalo.edu/SAFPHR/

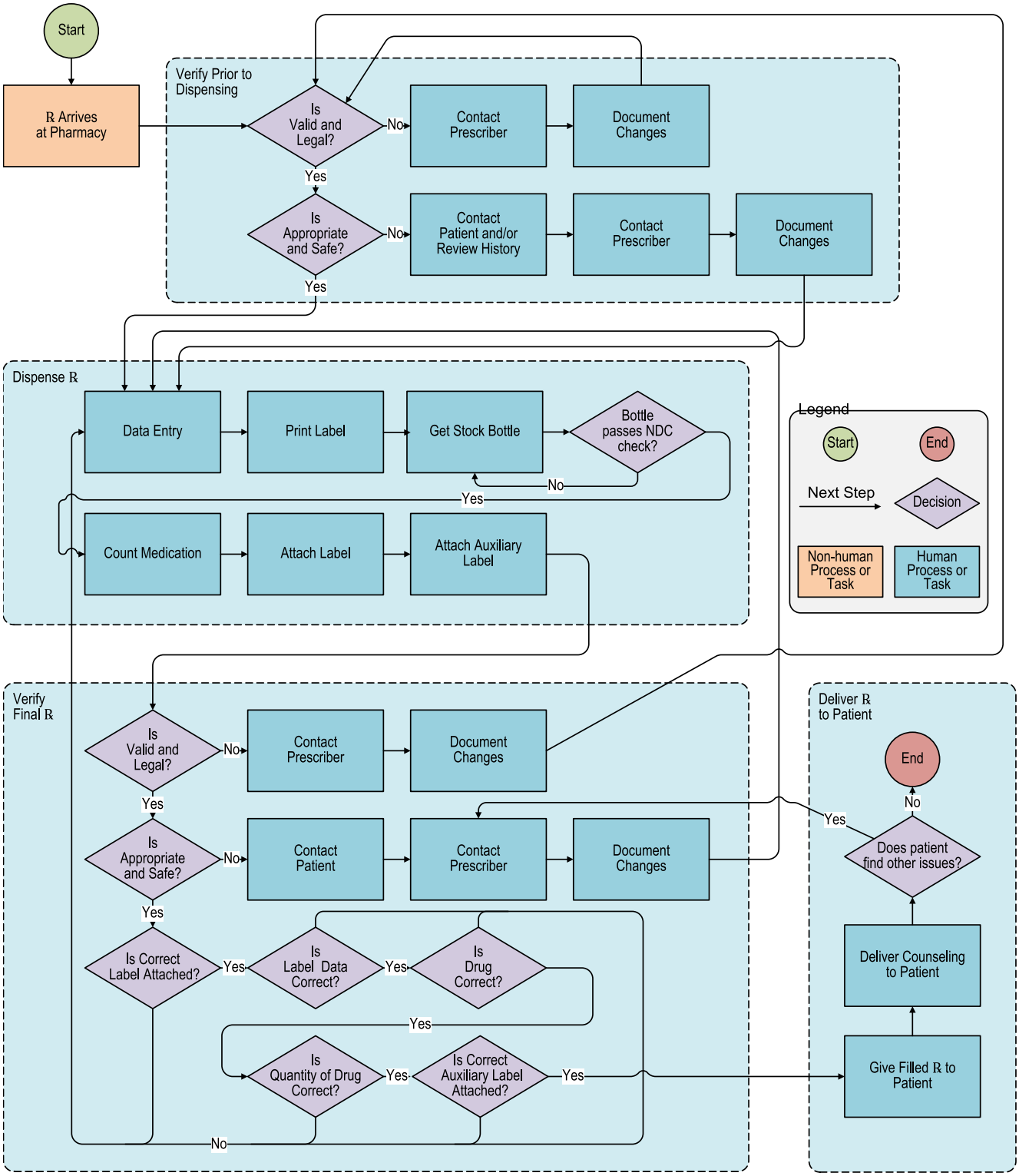

Fig. 3. Flow diagram of the community pharmacy dispensing procedures analyzed with SAFPHR. The procedure's start and end are circles. Diamonds are used for human decision tasks. Other human tasks are rectangles. Arrows point to the next step/task in the procedure. Arrows out of decisions point to the next step/task based on the answer to the decision's question. Reproduced from (Zheng et al., 2020).

example, "℞ Arrives at Pharmacy" from Fig. 3), a formula maps the step to a probability of error which is passed through "Compute Probability of Error". Ultimately, procedure compliance uses the computed error probability to determine if the given procedure step is performed "correct" or "incorrect." The procedure and environmental dynamism models can observe the state of procedure compliance so that the "correct" or "incorrect" outcomes can influence future procedure performance. For example, a pharmacist examines whether part of a prescription was filled properly in a decision task.

The major difference in the new version of the architecture (compared to basic SAFPHR; (Zheng et al., 2020)) comes from the "Map Procedure State to Nominal Probabilities" formula (highlighted in

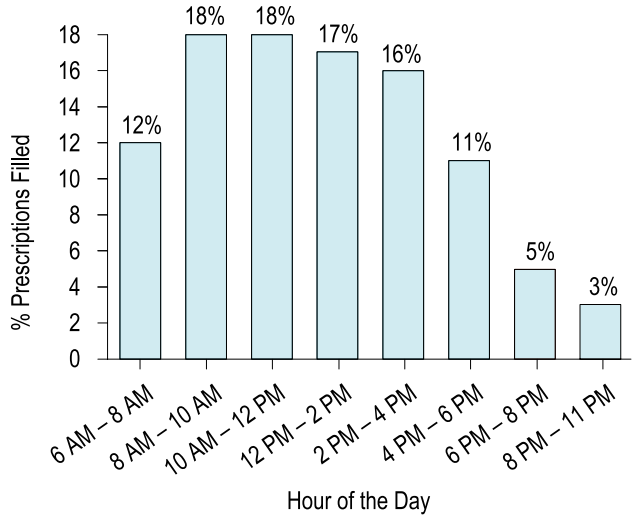

Fig. 4. Graph illustrating the distribution of prescriptions filled at a Western New York pharmacy. There are eight time zones, where all but the last encapsulates a two-hour time period. The last represents a three-hour period. The bar above each period shows the average percentage of prescriptions filled during that period. (Reproduced from (Zheng et al., 2020)). Note that this temporal distribution is calculated based on five-and-half months of real data from a typical, Western New York, community pharmacy. This data was used because it was the distribution most familiar to our subject matter expert. On average, 425 prescriptions will be dispensed daily. The average fill counts per hour from 6 AM to 10 PM are 29, 22, 34, 44, 39, 36, 37, 34, 33, 36, 27, 19, 12, 9, 4, 8, and 2, respectively.

blue). This functionally maps the state of the procedure sub-model to the nominal probabilities of occurrence (*CFP*; Table 2). Specifically, this formula uses CPC values to calculate the scaling factor for adjusting the *CFP*. The "Compute Probability of Error" formula will then multiply the *CFP* by the scaling factor associated with the control mode (S; Fig. 2; indicated by the CPCs) or by the CPCs' impacts (Table 3) for mode-effect and CPC-effect extended SAFPHR respectively.

4.2. Architectural Implementation in PRISM

Following the architecture in Fig. 5, we implement the extended SAFPHR analysis on the dispensing procedure model using PRISM's input language (Parker et al., 2017) for both the mode-effect and CPC-effect versions of extended SAFPHR. As with the basic SAFPHR model (Zheng et al., 2020), the procedure and dynamics from Figs. 3 and 4 and their associated CPC and cognitive function failure assessments were formulated as a discrete-time Markov chain (DTMC). In what follows, we describe PRISM code that was used to realize models for both versions of extended SAFPHR (see Figs. 6–12). Note that all of this code is derivative of the code originally used in the creation of basic SAFPHR (Zheng et al., 2020). Thus, the discussion below specifically describes where differentiation occurs from basic SAFPHR.

Presented code is formatted consistently. Comments are light blue and follow `//` marks. The reserved words of the PRISM modeling language are magenta. Named constants (i.e. `Improved` in Fig. 6) are green. Modules (i.e. `TimePeriod` in Fig. 7) are dark yellow. Variables (i.e. `t` in Fig. 7) are blue. Formulas like `ProbError` in Fig. 8 (which are dynamically computed in each state based on the values of the model's constants, variables, and other formulas) are orange. Red items are placeholders for code that the analyst would manually specify when completing a model. For example, `b` in Fig. 8 is the step of a task such as `Task_b`. Similarly, `DecisionCriterion` is a Boolean expression indicating when a decision task should evaluate to "yes.".

Fig. 6 lists all the constants that will be shared in later model concepts for both versions of extended SAFPHR. Constants in lines `4–25` are defined the same as in basic SAFPHR. `Improved`, `NotSignificant`, and `Reduced` from lines `4–6` are integer constants representing the three levels of CPCs. From lines `9–11`, `Incorrect`, `NotApplicable`, and `Correct` are constants for indicating if a task was performed incorrectly, not performed yet (or at all), or correctly. The `Correct` and `Incorrect` constants are also used to represent whether components of a filled prescription have errors. From lines `14–18`, `Start–End` define unique IDs for each element in the procedure model (Fig. 3). In special circumstances, such as the non-human task "℞ Arrives at Pharmacy" from Fig. 3, the element has two IDs (`RxArrives1` and `RxArrives2` in Fig. 6) because it must account for two respective factors in the modeled prescription: (1) its validity and legality and (2) its appropriateness and safety. In lines `21–22`, constants define the probabilities of prescriptions arriving with validity and legality (`P_V`) and appropriateness and safety (`P_S`) problems. The actual probabilities used here were based on the values we identified through a combination of literature review and simulation analyses (Zheng et al., 2020; Zheng, 2020). The probability that a patient will discover errors with a prescription once it has been delivered to her (Witte and Dundes (2007)) occurs on line `25`. Finally, for both versions of extended SAFPHR, we define the constant nominal values for each type of cognitive function failures from Table 2 in lines `28–34`.

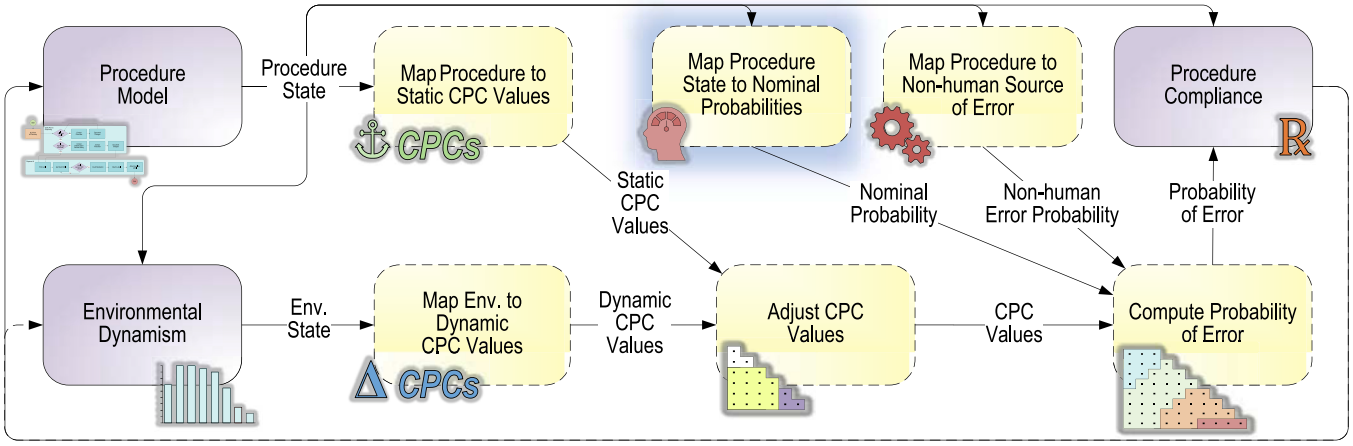

Fig. 5. Overview of extended SAFPHR's formal modeling architecture. Solidly lined shapes are sub-models. Dotted shapes are formulas (functions that compute values using variables from other formulas and sub-models). Variables shared between sub-models and formulas are represented with arrows. Dotted arrows represent shared variables that could exist in the architecture, but are not used in current analyses. Note that "Map Procedure State to Nominal probabilities," which is highlighted, represents the new contribution beyond what was reported in (Zheng et al., 2020).

```
1  dtmc // discrete-time Markov chain
2
3  // CPC level constants
4  const int Improved       = 1;
5  const int NotSignificant = 0;
6  const int Reduced        = -1;
7
8  // Task performance status
9  const int Incorrect     = -1;
10 const int NotApplicable = 0;
11 const int Correct       = 1;
12
13 //Procedure nodes
14 const int Start      = 1; // Each procedure node (start, end, task, or decision) has a
15 const int RxArrives1 = 2; // unique id with End being the total number of nodes
16 const int RxArrives2 = 3;
17 ...
18 const int End    = 32;
19
20 // Constant probabilities used by non-human sources of error
21 const double P_V = 0.1;
22 const double P_S = 0.1;
23
24 // Constant probabilitie relate to decisions not made by the pharmacist
25 const double P_PatientFindsIssues = 12/33;
26
27 //Constant nominal values of cognitive function failures
28 const double O1_WrongObjectObserved    =  0.001;  const double O2_WrongIdentification   =  0.07;
29 const double O3_ObservationNotMade     =  0.07;   const double I1_FaultyDiagnosis       =  0.2;
30 const double I2_DecisionError          =  0.01;   const double I3_DelayedInterpretation =  0.01;
31 const double P1_PriorityError          =  0.01;   const double P2_InadequatePlan        =  0.01;
32 const double E1_ActionOfWrongType      =  0.003;  const double E2_ActionAtWrongType.    =  0.003;
33 const double E3_ActionOnWrongObject    =  0.0005; const double E4_ActionOutOfSequence   =  0.003;
34 const double E5_MissedAction           =  0.03;
35 ...
```

Fig. 6. Example model code for defining general constants that will be used in later model concepts for both versions of extended SAFPHR. Those constants defined from lines 4–22 were also used in basic SAFPHR analysis (Zheng et al., 2020). Constants in lines 28–34 will be used in extended SAFPHR analysis to define the associated nominal probabilities for each CFF.

```
35 ...
36 // Environmental dynamism, eight time zones
37 module TimePeriod
38     t:[0..8] init 0;
39     [] t = 0 -> 0.12 : (t' = 1)  // 6 AM - 8 AM
40               + 0.18 : (t' = 2)  // 8 AM - 10 AM
41               + 0.18 : (t' = 3)  // 10 AM - 12 AM
42               + 0.17 : (t' = 4)  // 12 AM - 2 PM
43               + 0.16 : (t' = 5)  // 2 PM - 4 PM
44               + 0.11 : (t' = 6)  // 4 PM - 6 PM
45               + 0.05 : (t' = 7)  // 6 PM - 8 PM
46               + 0.03 : (t' = 8); // 8 PM - 11 PM
47 endmodule
48 ...
```

Fig. 7. Example model code for implementing the environmental dynamism of the model `TimePeriod` from Fig. 4 for both versions of extended SAFPHR. This `TimePeriod` module were also used in basic SAFPHR (Zheng et al., 2020).

Figs. 7 and 8 describe the two synchronously composed modules that represent the discrete-time Markov chain behavior of the model for the environmental dynamism of the model (`TimePeriod`) and the performance of the dispensing procedure (`Procedure`) respectively. Both are unchanged from basic safer.

The `TimePeriod` module appears in lines 37–47 from Fig. 7. In this, `t` represents which of the eight time periods a prescription arrives in Fig. 4. The time period `t` in the larger model' state is assigned based on the distribution of prescription arrivals (the detailed transition assignment are listed in lines 39–46). Thus, there is a 12% chance that `t` will equal 1, an 18% chance that it will be 2, and so on for all eight time periods.

The module for representing the procedure is presented in lines 50–100 of Fig. 8. This has a single variable `ProcedureStep` (line 52) to represent the element of the procedure being performed. The following variables (lines 55–68) are associated with each task of the procedure, indicating if it has been performed correctly (`Correct`), incorrectly (`Incorrect`), or not performed (`NotApplicable`; the default). This includes `RxValidLegal` and `RxAppSafe`, which account for non-human source of errors and (in our implementation) whether the prescription arrives to the pharmacy with errors. All other variables that begin with `Task_` are generic. The transition logic that follows variable definition describes how the procedure changes based on modeled system conditions. The first transition (lines 70–71) starts the performance of the procedure. This requires that module `TimePeriod` to have assigned a value for `t` that is bigger than 0. With this assignment completed (`t > 0` to be true), with a probability of 1, the procedure step will be set to `RxArrives1` (the ID of the first task in current model). The two transitions that follow (lines 73–78) show how a non-human task is represented. Because the task in question ("R Arrives at Pharmacy"; Fig. 3) can determine whether a prescription arrives with errors in two different ways, its behavior is spread over two transitions. The first determines if validity and legality issues will exist in the arriving prescription. In this, with a probability of `P_V` (a constants), `RxValidLegal` will be `Incorrect` indicating that the prescription is invalid or illegal; with a probability of `1 - P_V`, `RxValidLegal` will be `Correct` implying that the prescription will arrive without validity and legality errors. The `ProcedureStep` will move to the next transition in both situations. In the second transition, the same logic is used to determine if the arriving prescription contains appropriateness and safety issues.

Transitions from lines 59–68 (Fig. 8) describe different generic types of human task behavior. Given that these are human tasks, probability of each being performed erroneously (`ProbError`) is

```
48 ...
49 // Procedure sub-model
50 module Procedure
51     // Variable representing the current step in the procedure
52     ProcedureStep : [Start..End]         init Start;
53     ...
54     // Variable showing if the prescription is valid and legal
55     RxValidLegal  : [Incorrect..Correct] init NotApplicable;
56     // Variable showing if the prescription is appropriate and safe
57     RxAppSafe     : [Incorrect..Correct] init NotApplicable;
58     ...
59     Task_a,        : [Incorrect..Correct] init NotApplicable; // There is a unique Task_ variable
60     Task_b,        : [Incorrect..Correct] init NotApplicable; // for every task in the modeled
61     Task_c,        : [Incorrect..Correct] init NotApplicable; // procedure. a, b, ..., h represent
62     Task_d,        : [Incorrect..Correct] init NotApplicable; // the unique procedure step (with
63     ...                                                       // assumed values between Start and
64     Task_e,        : [Incorrect..Correct] init NotApplicable; // End) associated with the task
65     Task_f,        : [Incorrect..Correct] init NotApplicable;
66     ...
67     Task_g,        : [Incorrect..Correct] init NotApplicable;
68     Task_h,        : [Incorrect..Correct] init NotApplicable;
69     ...
70     [] t > 0 & (ProcedureStep = Start) ->
71         1                      :  (ProcedureStep' = RxArrives1);
72     // The first transition: in our model this represents a task with a non-human source of error
73     [] (ProcedureStep = RxArrives1) ->
74         1 - P_V  : (RxValidLegal' = Correct)   & (ProcedureStep' = RxArrives2)
75         + P_V    : (RxValidLegal' = Incorrect) & (ProcedureStep' = RxArrives2);
76     [] (ProcedureStep = RxArrives2) ->
77         1 - P_S     : (RxAppSafe' = Correct)      & (ProcedureStep' = a)
78         + P_S       : (RxAppSafe' = Incorrect)    & (ProcedureStep' = a);
79     ...
80     // Form of the transitions for a decision task:
81     // DecisionCriterion is a Boolean expression indicating if the correct decision is "yes"
82     [] (ProcedureStep = b) & DecisionCriterion ->
83         1 - ProbError          : (Task_b' = Correct)       & (ProcedureStep' = c)
84         + ProbError            : (Task_b' = Incorrect)     & (ProcedureStep' = d);
85     [] (ProcedureStep = b) & !DecisionCriterion ->
86         1 - ProbError          : (Task_b' = Correct)       & (ProcedureStep' = d)
87         + ProbError            : (Task_b' = Incorrect)     & (ProcedureStep' = c);
88     ...
89     // Form of the transitions for a human task:
90     [] (ProcedureStep = e) ->
91         1 - ProbError          : (Task_e' = Correct)       & (ProcedureStep' = f)
92         + ProbError            : (Task_e' = Incorrect)     & (ProcedureStep' = f);
93     ...
94     // The transitions consider patients discovering an error or accepting the prescription
95     [] (ProcedureStep = g) & ErrorInFilledPrescription ->
96         1 - P_PatientFindsIssues : (ProcedureStep' = End)  // No additional transitions can occur
97         + P_PatientFindsIssues   : (ProcedureStep' = h);   // once ProcedureStep = End, it means
98     [] (ProcedureStep = g) & !ErrorInFilledPrescription -> // prescription has been successfully
99         1                        : (ProcedureStep' = End); // delivered to the patient
100 endmodule
101 ...
```

Fig. 8. Example model code for implementing the procedure sub-model for both versions of extended SAFPHR. This `Procedure` module were also used in basic SAFPHR analysis (Zheng et al., 2020) to describe how the prescription will move through the procedure.

```
101 ...
102 // Dynamic CPC values determined by the time period (t)
103 formula TimeOfDay         = (t = 5 | t = 7 | t = 8) ? Reduced : NotSignificant;
104 formula SimultaneousGoals = (t = 1 | t = 2)         ? Reduced : NotSignificant;
105 formula AvailableTime     = (t = 1)                 ? Reduced :
106                             (t = 3 | t = 4 | t = 6) ? NotSignificant : Improved;
107
108 // Static CPC values at the current procedure step (i).
109 // Note: Organization_i, ... , Collaboration_i are Specific Values Determined by CPC Assessment
110 formula Organization   = ... ProcedureStep = i ? Organization_i  : ... : NotSignificant;
111 formula Conditions     = ... ProcedureStep = i ? Conditions_i    : ... : NotSignificant;
112 formula Support        = ... ProcedureStep = i ? Support_i       : ... : NotSignificant;
113 formula Procedures     = ... ProcedureStep = i ? Procedures_i    : ... : NotSignificant;
114 formula Experience     = ... ProcedureStep = i ? Experience_i    : ... : NotSignificant;
115 formula Collaboration  = ... ProcedureStep = i ? Collaboration_i : ... : NotSignificant;
116
117 // CPC adjustment: we only show the adjustment for Collaboration. Other adjustments are similar.
118 ...
119 formula CollaborationAdj = ((Organization = Improved ? 1 : 0)
120                           + (Experience = Improved ? 1 : 0)) = 2 ? Improved :
121                          ((Organization = Reduced  ? 1 : 0)
122                           + (Experience = Reduced  ? 1 : 0)) = 2 ? Reduced  : Collaboration;
123
124 // Nominal probability (CFP) at the current procedure step (i).
125 // Note that CFP_i are specific values determined by cognitive function failures assessment
126 formula CFP   = ... ProcedureStep = i ? CFP_i  : ... : 0;
127 ...
```

Fig. 9. Example model code for implementing all shared formulas that will be used for both versions of extended SAFPHR. The formula definitions from lines 103–122 were also used in basic SAFPHR analysis (Zheng et al., 2020). `CFP` is the new formula defined to get the corresponding cognitive function probabilities for each task.

```
...
// Scaling factor associated with the control mode indicated by the CPCs
const double StrategicS     = 0.94;  const double TacticalS  = 1.9;
const double OpportunisticS = 7.5;   const double ScrambledS = 23;

// Calculating the Scaling Factor
formula NumRed  = (Organization       = Reduced  ? 1 : 0) + (TimeOfDay        = Reduced  ? 1 : 0)
                + (GoalsAdj           = Reduced  ? 1 : 0) + (AvailableTimeAdj = Reduced  ? 1 : 0)
                + (ConditionsAdj      = Reduced  ? 1 : 0) + (Support          = Reduced  ? 1 : 0)
                + (Procedures         = Reduced  ? 1 : 0) + (Experience       = Reduced  ? 1 : 0)
                + (CollaborationAdj   = Reduced  ? 1 : 0);
formula NumImp  = (Organization       = Improved ? 1 : 0) + (TimeOfDay        = Improved ? 1 : 0)
                + (GoalsAdj           = Improved ? 1 : 0) + (AvailableTimeAdj = Improved ? 1 : 0)
                + (ConditionsAdj      = Improved ? 1 : 0) + (Support          = Improved ? 1 : 0)
                + (Procedures         = Improved ? 1 : 0) + (Experience       = Improved ? 1 : 0)
                + (CollaborationAdj   = Improved ? 1 : 0);

formula ModeScaling = (NumRed = 0 & NumImp >= 4) | (NumRed =  1 & NumImp >= 5)
                    | (NumRed = 2 & NumImp >= 6)                                ? StrategicS :
                      (NumRed = 0 & NumImp <  4) | (NumRed =  1 & NumImp <  5)
                    | (NumRed = 2 & NumImp <  6) | (NumRed =  3 & NumImp >= 2)
                    | (NumRed = 4 & NumImp >= 3) | (NumRed =  5 & NumImp >= 4) ? TacticalS :
                      (NumRed = 3 & NumImp <  2) | (NumRed =  4 & NumImp <  3)
                    | (NumRed = 5 & NumImp <  4) | (NumRed >= 6 & NumImp >= 1) ? OpportunisticS :
                                                                                  ScrambledS;

formula ProbError = ModeScaling*CFP > 1 ? 1: ModeScaling*CFP;
```

Fig. 10. Example model code for completing model for Mode-Effect Extended SAFPHR [the version that adjusts the nominal probability with scaling factors dictated by the control modes (*S* from Fig. 2)]. The formulas `NumRed` and `NumImp` from lines `133–142` were also used in basic SAFPHR analysis (Zheng et al., 2020) to count the number of CPCs that are improved and the number of CPCs that are reduced. Codes in lines `129–130` and in lines `144–151` are specifically used to calculate the weighting factors to adjust the CFP for mode-effect extended SAFPHR. The formula `ProbError` in line `153` shows how the probability of error will be calculated in mode-effect extended SAFPHR analysis.

computed using a formula reported in Fig. 10 or Fig. 12 for different versions of extended SAFPHR (discussed later). Lines `82–87` depict how two transitions are used to represent a decision task. The two transitions account for the two conditions that could occur during model execution, where `DecisionCriterion` (a placeholder for a task-specific Boolean expression) is true and the pharmacist should decide "Yes" or `DecisionCriterion` is false (`!DecisionCriterion`) and he or she should decide "No," with errors potentially made under both

```
...
//Weighting factors for CPCs
formula OBSOrganization  = 1; // e.g. scaling for organization under observation function
formula INTOrganization  = 1; // e.g. scaling for organization under interpretation function
formula PLANOrganization = Organization = 1 ? 0.8 : Organization = 0 ? 1 : 1.2 ;
    // Note: if oranization is deficient, PLANOrganization = 2, EXEOrganization = 2
formula EXEOrganization  = Organization = 1 ? 0.8 : Organization = 0 ? 1 : 1.2 ;

formula OBSWorkingConditions  = ConditionsAdj = 1 ? 0.8 : ConditionsAdj = -1 ? 2 : 1;
formula INTWorkingConditions  = ConditionsAdj = 1 ? 0.8 : ConditionsAdj = -1 ? 2 : 1;
formula PLANWorkingConditions = 1;
formula EXEWorkingConditions  = ConditionsAdj = 1 ? 0.8 : ConditionsAdj = -1 ? 2 : 1;

formula OBSOperationalSupport  = Support = 1 ? 0.5 : Support = -1 ? 5 : 1;
formula INTOperationalSupport  = 1;
formula PLANOperationalSupport = 1;
formula EXEOperationalSupport  = Support = 1 ? 0.5 : Support = -1 ? 5 : 1;

formula OBSProcedureAvailability  = Procedures = 1 ? 0.8 : Procedures = -1 ? 2 : 1;
formula INTProcedureAvailability  = 1;
formula PLANProcedureAvailability = Procedures = 1 ? 0.5 : Procedures = -1 ? 5 : 1;
formula EXEProcedureAvailability  = Procedures = 1 ? 0.8 : Procedures = -1 ? 2 : 1;

formula OBSSimultaneousGoals  = GoalsAdj = -1 ? 2 : 1;
formula INTSimultaneousGoals  = GoalsAdj = -1 ? 2 : 1;
formula PLANSimultaneousGoals = GoalsAdj = -1 ? 5 : 1;
formula EXESimultaneousGoals  = GoalsAdj = -1 ? 2 : 1;

formula OBSAvailableTime  = AvailableTimeAdj = 1 ? 0.5 : AvailableTimeAdj = -1 ? 5 : 1;
formula INTAvailableTime  = AvailableTimeAdj = 1 ? 0.5 :  AvailableTimeAdj = -1 ? 5 : 1;
formula PLANAvailableTime = AvailableTimeAdj = 1 ? 0.5 :  AvailableTimeAdj = -1 ? 5 : 1;
formula EXEAvailableTime  = AvailableTimeAdj = 1 ? 0.5 :  AvailableTimeAdj = -1 ? 5 : 1;

formula OBSTimeOfDay  = TimeOfDay = -1 ? 1.2 : 1;
formula INTTimeOfDay  = TimeOfDay = -1 ? 1.2 : 1;
formula PLANTimeOfDay = TimeOfDay = -1 ? 1.2 : 1;
formula EXETimeOfDay  = TimeOfDay = -1 ? 1.2 : 1;

formula OBSTrainingAndExperience  = Experience = 1 ? 0.8 : Experience = -1 ? 2 : 1;
formula INTTrainingAndExperience  = Experience = 1 ? 0.5 : Experience = -1 ? 5 : 1;
formula PLANTrainingAndExperience = Experience = 1 ? 0.5 : Experience = -1 ? 5 : 1;
formula EXETrainingAndExperience  = Experience = 1 ? 0.8 : Experience = -1 ? 2 : 1;

formula OBSCollaborationQuality  = CollaborationAdj = 1 ? 0.5 :  CollaborationAdj = -1 ? 2 : 1;
formula INTCollaborationQuality  = CollaborationAdj = 1 ? 0.5 : CollaborationAdj = -1 ? 2 : 1;
formula PLANCollaborationQuality = CollaborationAdj = 1 ? 0.5 : CollaborationAdj = -1 ? 2 : 1;
formula EXECollaborationQuality  = CollaborationAdj = 1 ? 0.5 : CollaborationAdj = -1 ? 5 : 1;
...
```

Fig. 11. Example model code using in CPC-Effect Extended SAFPHR models to implement the CPC-based scaling of probabilities from Table 3.

```
174 ...
175 // If exists 1; else 0
176 formula ObservationError    = ... ProcedureStep = i ? ObservationError_i     : ... : 0;
177 formula InterpretationError = ... ProcedureStep = i ? InterpretationError_i  : ... : 0;
178 formula PlanningError       = ... ProcedureStep = i ? PlanningError_i        : ... : 0;
179 formula ExecutionError      = ... ProcedureStep = i ? ExecutionError_i       : ... : 0;
180
181 formula OBS = ObservationError    = 0 ? 0 : OBSOrganization*OBSWorkingConditions*OBSOperationalSupport*
        OBSProcedureAvailability*OBSSimultaneousGoals*OBSAvailableTime*OBSTimeOfDay*OBSTrainingAndExperience*
        OBSCollaborationQuality;
182
183 formula INT = InterpretationError = 0 ? 0 : INTOrganization*INTWorkingConditions*INTOperationalSupport*
        INTProcedureAvailability*INTSimultaneousGoals*INTAvailableTime*INTTimeOfDay*INTTrainingAndExperience*
        INTCollaborationQuality;
184
185 formula PLAN = PlanningError      = 0 ? 0 : PLANOrganization*PLANWorkingConditions*PLANOperationalSupport*
        PLANProcedureAvailability*PLANSimultaneousGoals*PLANAvailableTime*PLANTimeOfDay*PLANTrainingAndExperience*
        PLANCollaborationQuality;
186
187 formula EXE = ExecutionError     = 0 ? 0 : EXEOrganization*EXEWorkingConditions*EXEOperationalSupport*
        EXEProcedureAvailability*EXESimultaneousGoals*EXEAvailableTime*EXETimeOfDay*EXETrainingAndExperience*
        EXECollaborationQuality;
188
189 formula CPCWeighting = max(OBS, INT, PLAN, EXE);
190
191 formula ProbError = CPCWeighting*CFP > 1 ? 1 : CPCWeighting*CFP;
```

Fig. 12. Example model code for completing models for CPC-Effect Extended SAFPHR analysis by calculating the scaling factors for CPCs and the probabilities of error. Lines 176–189 together with the content of Fig. 11 illustrate how we calculate the weighting factors used to adjust the CFP. The formula ProbError in line 191 shows how the probability of error is calculated.

conditions. Note that across these transitions, the task procedure that will be performed next will be determined by the truth of DecisionCriterion as well as whether or not the pharmacist makes an error. In contrast to a decision task, a standard human task (see lines 90–92) only had one transition, has no DecisionCriterion, and only ever proceeds to one next step.

The last set of transitions in the module (lines 95–99) represent the transitions that can complete the prescription filling process and setting ProcedureStep to End. These transitions also account for contexts in which a patient may discover an error in the filled prescription and may either give it back to the pharmacist or accept it (both conditions that can occur with or without error). For these transitions, ErrorInFilledPrescription represents a Boolean expression that indicates if the filled prescription contains any error. If the arriving prescription is invalid, illegal, inappropriate, or unsafe, and the pharmacists failed to catch and correct those issues; or if the pharmacists made any mistakes under the "Dispense Rx" sub-model, the value of ErrorInFilledPrescription will be true. It is false otherwise. Furthermore, if the prescription contains an error, there is a set probability that the patient will identify it (P_PatientFindsIssues in Fig. 8). Thus, under ProcedureStep = g and ErrorInFilledPrescription, with a probability of P_PatientFindsIssues (a constant discussed above), the patient discovers an error thus the ProcedureStep will go back to a middle step. All of the filled prescriptions will be delivered otherwise.

Fig. 9 shows all the shared formulas that will be used in both versions of extended SAFPHRx. Formulas in lines 103–122 were also used in the implementation of basic SAFPHRx, where the three formulas in lines 103–106 compute the three dynamic CPC values based on the time period t and those in lines 110–115 determine the values of the static CPCs at the current procedure step (all derived from assessments performed with the subject matter expert). The formula in lines 119–122 provides an example of how the "Collaboration" CPC is adjusted based on rules from Fig. 1. The remaining formula in line 126, was unique to the implementations of extended SAFPHRx. In this, we determine the nominal probability of an error occurring (the CFP value) at the current procedure step based on which CFF is most relevant to the current task (note that this is determined by assessments taken from the subject matter expert). If ProcedureStep = i, the value of CFP at step i will be CFP_i. Otherwise, the value of CFP will be NotSignificant. Note that i represents a placeholder for any given procedure step and CFP_i is the assessed value of the CFP for that step.

Fig. 10 illustrates the supplement steps that are required to complete the mode-effect extended SAFPHRx analysis while Figs. 11 and 12 list all the supplement code required to complete the CPC-effect extended analysis. Code for both extended SAFPHRx versions are designed to compute the required scaling factors and then calculate the probabilities of human error (ProbError) used in the transitions from Fig. 8.

For mode-effect extended SAFPHRx, the formulas for NumRed and NumImp (lines 133–142 in Fig. 10) are used to count the number of CPCs (post adjustment) rated as Reduced or Improved, respectively. The ModeScaling formula (lines 144–151) uses these values to compute the scaling factor associated with the COCOM control modes (S from Fig. 2). In this model, the different possible values of S are modeled as the constants StrategicS, TacticalS, OpportunisticS, and ScrambledS from lines 129–130. Then, the probability of error is computed by multiplying the values of ModeScaling and CFP via the formula ProbError (Fig. 10, line 153). Note that if the CPCs are assessed as being generally unfavorable, the calculated total influence of the CPCs could be large, which will lead to a probability greater than 1. Since this violates the laws of probability, we adopt the method used in CREAM (Hollnagel, 1998a), where such values become 1.

For CPC-effect extended SAFPHRx, all formulas in Fig. 11 are used to represent the rules described in Table 3 into PRISM's input language. For example, the OBSWorkingConditions (line 135) represents how the CPC "Conditions" will impact an "Observation" function-related error. The value that measures this impact will be determined based on the value of Conditions for the current activity (ProcedureStep). If Conditions is rated as Improved, the scaling factor for it adjusting "Observation" function-related failures is 0.8; if the Conditions is NotSignificant, the scaling factor will be 1; if the Conditions is Reduced, the scaling factor will be 2. All of these scaling factors will be passed through formulas OBS, INT, PLAN, and EXE (lines 181–187 from Fig. 12)) to calculate the total influence of the CPCs under Observation, Interpretation, Planning, and Execution respectively. To get these total influence factors, we also need to know the corresponding cognitive function from Table 2 based on the assessment of cognitive function failures. Formulas from lines 176–179 are used to calculate this information: where each of the four variables (ObservationError, InterpretationError, PlanningError, and ExecutionError) will be 1 if the procedure step is associated with the corresponding cognitive function and 0 otherwise. The 1 or 0 values are encompassed by ObservationError_i,

`InterpretationError_i`, `PlanningError_i`, and `ExecutionError_i` for a given step `i` in the model. For example, when `ProcedureStep = i`, if the current procedure step is identified to have an Observation-related error, `ObservationError_i` will be 1 and `InterpretationError_i`, `PlanningError_i`, and `ExecutionError_i` will be 0 because the failure of each task will only fall into one function category. Thus, formula `CPCWeighting` (line `189`) can use the MAX function to find the weighting factor that accounts for how specific CPCs impact the associated cognitive activities. Similar to mode-effect extended SAFPHR℞, multiplying `CFP` by a weighting factor `CPCWeighting` will get the probability of error (formula `ProbError` from line `191`).

4.3. Specification properties

To perform analyses on a formal model using probabilistic model checking, an analyst must have a specification property that requests the computation of a probability or proof about the model. To make predictions using the two new versions of extended SAFPHR℞ comparable with predictions from basic SAFPHR℞ and published overall error data, we check a probabilistic temporal logic property we identified in (Zheng et al., 2020) called Procedure Eventual Reliability. This allows us to use probabilistic model checking to assess the overall reliability of the pharmacy procedure (its overall dispensing error rate, which we refer to as procedure eventual reliability) with:

Procedure Eventual Reliability:

$$P = ?\left[\mathtt{F}\left(\begin{array}{l}(ProcedureStep = End)\\ \wedge\begin{pmatrix}(RxValidLegal = Incorrect)\\ \vee(RxAppSafe = Incorrect)\\ \vee_{t\in T}(t = Incorrect)\end{pmatrix}\end{array}\right)\right]. \quad (1)$$

This tells PRISM to calculate the probability ($P = ?$) that the prescription eventually ($\mathtt{F}$) is delivered to the patient ($ProcedureStep = End$) with an error that the prescription either arrived with $((RxValidLegal = Incorrect)\vee(RxAppSafe = Incorrect))$ or manifested through the misperformance of any of the human tasks from the "Dispense ℞" and "Deliver ℞ to Patient" sub-models ($t \in T$).

4.4. Methods

Using extended SAFPHR℞ with the CPC and CFF assessments from our subject matter expert, we evaluated the reliability for the same typical United States community pharmacy in Fig. 3 (based on data from a Western New York pharmacy that does not use automated dispensing equipment; see Fig. 4) as in the basic SAFPHR℞ analyses (Zheng et al., 2020).

To calculate the overall error rate of the procedure, specification asserted using the property pattern in Eq. (1) was checked using the PRISM model checker on a desktop computer with a 3.70 GHz Xeon processor and 128 GB of RAM running Linux Mint. In doing this, we used PRISM's command-line option to set the upper memory limits to 4 Gigabytes and to use the multi-terminal binary decision diagrams (MTBDDs) engine, the developer-recommended option for enabling PRISM to handle large, structured models (Parker, 2003).

4.5. Results and conclusion

Our analyses of the formal models resulted in CPC-effect extended SAFPHR℞ predicting a procedure eventual reliability error rate of 0.125069976, after 210.671 seconds of analysis time. The mode-effect extended SAFPHR℞ analysis produced an error rate of 0.060506516 in 920.884 seconds. By comparison, basic SAFPHR℞ produced a range of error rates between 1.02422E-05 and 0.028856292 (in 33.098 and 153.238 s respectively), with a geometric mean of 0.000543647 (Zheng et al., 2020) and an arithmetic mean of 0.014433267. Fig. 13 compares the computed rates along with the range of error rates reported in the literature (Szeinbach et al., 2007; Odukoya et al., 2015).

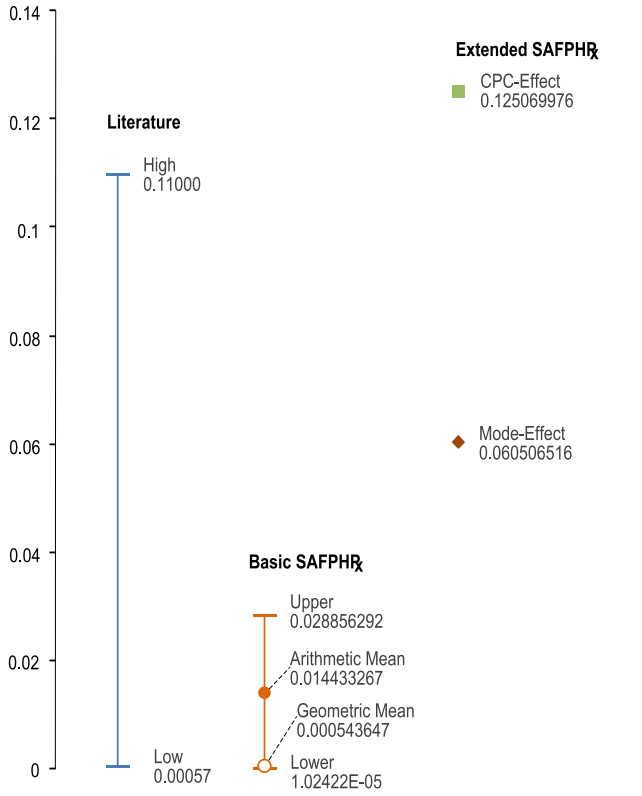

Fig. 13. A graph indicating the relationship among published error data with error rate estimates predicted by different versions of SAFPHR℞. Note that the upper and lower bounds as well as the geometric mean for basic SAFPHR℞ were originally reported in (Zheng et al., 2020). The arithmetic mean was computed as the arithmetic average of the upper and lower bound.

This showed how we were able to successfully incorporate extended CREAM concepts into SAFPHR℞ to create two versions of extended SAFPHR℞ and use these to analyze the reliability of the community pharmacy procedures. By grounding the HRA in extended CREAM (Hollnagel, 1998a) and the PRISM probabilistic model checker (Kwiatkowska et al., 2011), both versions of extended SAFPHR℞ allow us to get point estimation on reliability error rates.

However, the results produced by extended SAFPHR℞ are inconsistent with results from basic SAFPHR℞. As was shown in Fig. 13, both versions of extended SAFPHR℞ produced error rate predictions that were outside of the range predicted by basic SAFPHR℞. In fact, both were noticeably higher than basic SAFPHR℞'s upper bound. Furthermore, three of the estimates reported in Fig. 13 failed to fall into the published error rate ranges: the value produced with CPC-effect extended SAFPHR℞, the lower bound predicted by basic SAFPHR℞, and the geometric mean calculated from the basic SAFPHR℞ range. However, basic SAFPHR℞'s lower bound and geometric mean are both very close to the lowest error rate of 0.057% reported by (Szeinbach et al., 2007). The error rate 0.125069976 produced with CPC-effect extended SAFPHR is the only one clearly outside of the range seen in the literature. The inconsistent estimates among different approaches make it unclear which version of the method should be trusted. Thus, in what follows, we set out to determine which of the six different SAFPHR℞ estimation methods produces the most valid predictions.

5. Validation

Given the range of possible realistic values from the literature

Table 4
The rates of errors originating from different stages.

Dispensing Stage	Error Rate						
	L	U	G	A	CEE	MEE	By Flynn et al. (2003)
Screening	0.0000001	0.0004023	0.0000046	0.0002012	0.1203438	0.0056226	0.0002232
Order Entry	0.0000032	0.0071865	0.0001513	0.0035948	0.1224741	0.0297437	0.0035706
Get Drug	< 1E-07	0.0000807	0.0000000	0.0000403	0.1200007	0.0000888	0.0055791
Count Drug	0.0000006	0.0078885	0.0000709	0.0039446	0.1203449	0.0023049	0.0015622
Packaging and Label	< 1E-07	0.0008701	0.0000018	0.0004351	0.0001924	0.0058238	0.0002232
Inspection and Storage	0.0000070	0.0222254	0.0003947	0.0111162	0.1230375	0.0378611	0.0113814
SR	0.0225287	0.0161138	0.0219164	0.0032074	0.5838538	0.0589053	0
Log SR	27.3169485	3.9845467	13.5061329	2.8913570	8.5847381	6.2275372	0
USS	0.0001758	0.0002014	0.0001658	0.0000365	0.0682357	0.0014774	0

In the above, L, U, G, A, CEE, and MEE represent the lower model of basic SAFPHR, upper model of basic SAFPHR, geometric mean of basic SAFPHR, arithmetic mean of basic SAFPHR, CPC-effect extended SAFPHR, and mode-effect extended SAFPHR respectively. SR, Log SR, and USS represent the sum of residuals, logarithm sum of residuals, and unweighted sum of square between the current column of data with the last column of data by Flynn et al. (2003).

(Szeinbach et al., 2007; Odukoya et al., 2015), we had to find other types of data on which to make validation comparisons. Fortunately, SAFPHR supports a number of different specification properties that can be used to determine the reliability of different parts of a procedure (Zheng et al., 2020). Thus, for the validation study, we compared results from each approach to SAFPHR to rates published in an aggregate and comprehensive study by Flynn et al. (2003). The particular study was chosen for two reasons. First, the error data were obtained by observing 50 pharmacies across the United States for approximately 10 months. Pharmacies from all four U.S Census Bureau regions (Midwest, Northeast, South, and West) were involved in their observational experiments. This makes their data collection more representative and general compared to other reports from the pharmacy literature. Second, and most importantly, Flynn et al. (2003) also measured the rate of process deviations and actual errors occurring at each stage of the prescription filling process. While the actual error rates among pharmacies could be different due to the variations in working process and environment, this study is good for allowing us to compare the relative size of rates from different parts of the process. These rates are shown later in the last column of Table 4. This allowed us to compare the accuracy of overall error rate predictions and error rates from different stages of the dispensing procedures.

5.1. Modeling

As shown in Table 4, Flynn et al. (2003) measured the frequency of errors occurring at different stages of the filling process: Screening, Order Entry, Get Drug, Count Drug, Packaging and Labeling, Inspection, and Storage of Filled Prescription. To check model predictions against these numbers, we first mapped tasks from our procedure (Fig. 3)) to each of Flynn et al. (2003) stages. “Data Entry” and “Print Label” from Fig. 3 correspond to Flynn et al. (2003) Order Entry stage; “Get Stock Bottle” and “Bottle Passes NDC check?” fall into the Get Drug stage; “Count Medication” represents the Count Drug stage, “Attach Label” and “Attach Auxiliary Label” are in the Packaging and Labeling stage; “Is Correct Label Attached?”, “Is Label Data Correct?”, “Is Drug Correct”, “Is Quantity of Drug Correct?”, “Is Correct Auxiliary Label Attached?”, “Give Filled R to Patient”, and “Deliver Counseling to Patient” all correspond to inspection and storage; the remainder of Fig. 3’s tasks account for the Screening stage.

With this mapping complete, we formulated specification properties to compute the probability of errors occurring at each of the different Flynn et al. (2003) stages. Each of these properties, when checked, calculate so called eventual reliability (Zheng et al., 2020): error rates based on whether errors are still present once the prescription is delivered to a patient. All of these specifications followed the pattern:

Eventual Reliability of Stage$X$:

$$P = ?\left[\mathbb{F}\left(\begin{array}{l}(ProcedureStep = End)\\ \wedge\left(\bigvee_{task\in \mathrm{Stage}X}(task = Incorrect)\right)\end{array}\right)\right]. \tag{2}$$

This tells the model checker to compute the probability ($P = ?$) that eventually ($\mathbb{F}$) the procedure will end ($ProcedureStep = End$) with one or more of the tasks associated with the given state (Stage$X$) having been done incorrectly ($\bigvee_{task\in \mathrm{Stage}X}(task = Incorrect)$).

Specification properties of this form were formulated for all six of the stages identified by Flynn et al. (2003).[4]

5.2. Methods

While using the nominal community dispensing procedure (Fig. 3) and CPC and CFF assessments from our subject matter expert, we checked the six specifications (one for each stage) formulated using the property pattern from Eq. (2) to compute error rates comparable to those reported by Flynn et al. (2003). This was done for each of the estimation options for basic SAFPHR (upper bound, lower bound, geometric mean, arithmetic mean) as well as CPC-effect extended SAFPHR, and mode-effect extended SAFPHR.

As in the previous analyses, we constructed the models and conducted the model checking using the PRISM model checker (with the same command-line options) on the same computer workstation. Note that, for specific cases that would not converge within 10000 iterations, we increased the speed of convergence by using the “topological value iteration” method and set the “termination epsilon” to 0.001. This only occurred when checking the rate of errors originating in the packaging and labeling stage with CPC-effect extended SAFPHR.

To facilitate comparison between how well the different model predictions fit Flynn et al.’s data, we also calculated several goodness of fit measures. The first two were based on the sum of residuals in for both the base 10

$$SR = \sum_{i=1}^{n} |y_i - \hat{\pi}_i| \tag{3}$$

and the logarithmic

$$LogSR = \sum_{i=1}^{n} |\log_{10} y_i - \log_{10} \hat{\pi}_i| \tag{4}$$

scales. Note that in Eqs. (3) and (4), y_i is the actual probability and $\hat{\pi}_i$ is the predicted probability. These gave us aggregate measures of the

[4] A full listing of specification properties checked in this work can be found at http://fhsl.eng.buffalo.edu/SAFPHR/

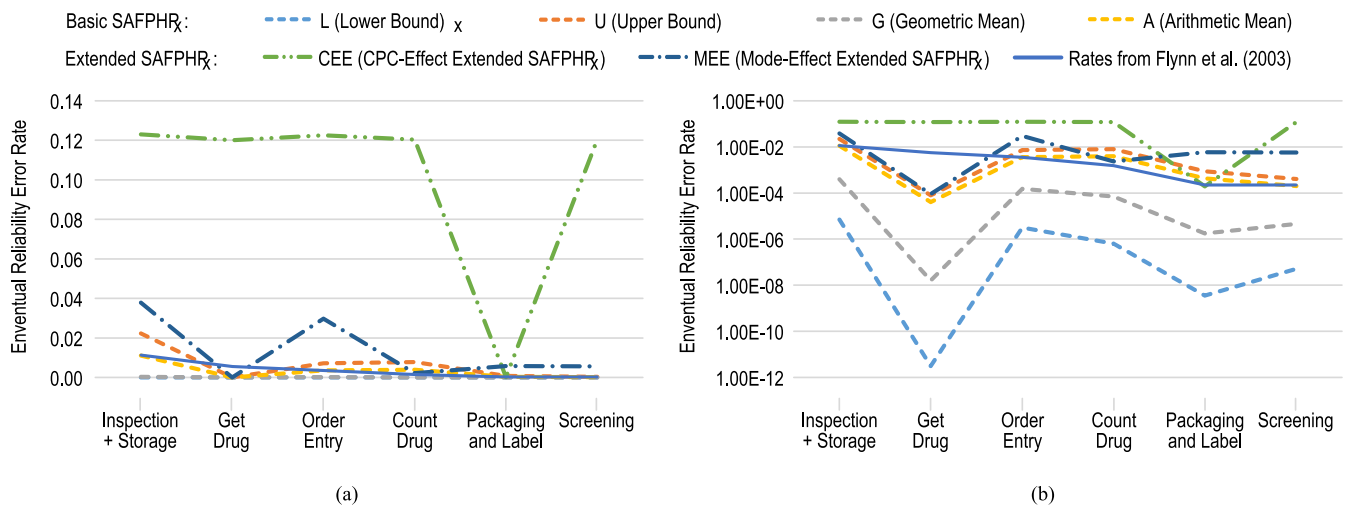

Fig. 14. The predicted SAFPHR point estimates of error rates originating from different stages of the dispensing on (a) base 10 and (b) logarithmic scales.

distance between the curves on both coordinate systems. The third statistical indicator we used was the Unweighted Sum of Squares (USS):

$$USS = \sum_{i=1}^{n} (y_i - \hat{\pi}_i)^2. \quad (5)$$

This was used because it has been shown to be an effective test for the goodness of fit for proportions (Copas, 1989; Hosmer et al., 1997; Allison, 2014). In all three of these measures, lower scores are better.

5.3. Results

The results of the verifications from each stage's eventual reliability property (Eq. (2)) are shown in Table 4. To better visualize these data, we compare predicted and actual (Flynn et al., 2003) rates on a standard base 10 scale (to illustrate the difference in prediction estimates) and on a logarithmic scale (to illustrate the difference in orders of magnitude) in Fig. 14. An examination of these results from both perspectives suggests that the worst predictions were made by CPC-effect SAFPHR (dashed green), basic SAFPHR's lower bound (light blue dashed line), and basic SAFPHR's geometric average (dotted gray line), while the best predictions were made by the arithmetic average of basic SAFPHR (the yellow dotted line).

Each of the three goodness of fit metrics ranked the six safer predictions differently (Table 4). However, all three were consistent with our observations from Fig. 14: that basic SAFPHR's arithmetic mean produced the best performance.

From the comparison of overall procedure eventual reliability error rate predictions using different versions of SAFPHR (see Fig. 15), we further found that the prediction made by the arithmetic mean for basic SAFPHR produced the closest overall error rate to that observed by Flynn et al. (2003).

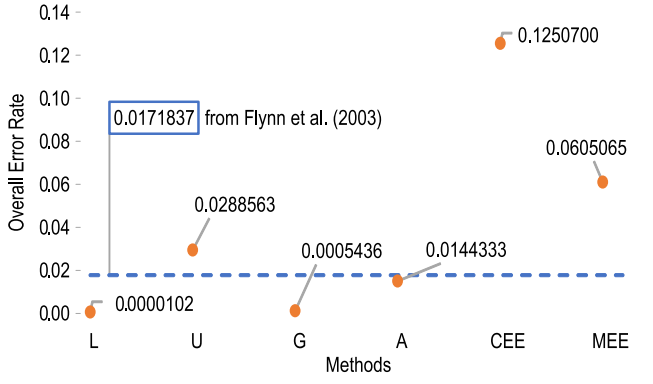

Fig. 15. The procedure eventual reliability error ratew predicted using different versions of SAFPHR. Note that L, U, G, A, CEE, and MEE represent the lower bound from basic SAFPHR, upper bound from basic SAFPHR, geometric mean from basic SAFPHR, arithmetic mean from basic SAFPHR, prediction from CPC-effect extended SAFPHR, and prediction mode-effect extended SAFPHR respectively. The blue dotted line indicates the actual error rate reported by Flynn et al. (2003).

6. Discussion And future work

In this work, we showed how we were able to incorporate extended CREAM concepts (Hollnagel, 1998a) into SAFPHR to create two versions of extended SAFPHR. This gave us the ability to find point estimates for human error rates. We used the new methods to analyze the reliability of a typical community pharmacy procedure and compared predictions to previous predictions with basic SAFPHR and the literature. Because the rates produced from the different versions of SAFPHR were inconsistent, we conducted a validation study to determine which of the different point estimates that can be obtained from SAFPHR are the most accurate. This showed that the arithmetic mean of basic SAFPHR's range was able to predict error rates best for both where they originate in a procedure and for the overall procedure.

In the following discussion, we explore the implication of these results from several perspectives. First, because the arithmetic average of basic SAFPHR ranges produced the most accurate results, we revisit the community pharmacy recommendations made in Zheng et al. (2020), which were based on the use of the geometric mean. Next, we investigate why CPC-effect extended SAFPHR appeared to perform worse than all of the other options, even though the theory would suggest it would perform the best. Finally, we explore the limitations of our method and recommend future research directions.

6.1. Recommendations for improving community pharmacy

In addition to the property for computing procedure eventual reliability (Eq. (1)), Zheng et al., 2020) introduced a number of methods for computing useful and insightful error rates from a SAFPHR model (all of which can be applied to any version of SAFPHR). Here, we revisit the recommendations made by Zheng et al. (2020) to determine what interventions have the potential to be the most effective given that this work has shown the arithmetic mean of basic SAFPHR averages are the most valid. Note that we restrict our discussion here to the analyses that provide the most insights into task-level design interventions. Additionally analyses can be found in (Zheng, 2020).

Table 5
Task General Reliability, Procedure Eventual Probability without task error contributions (New Prob.), and its Improvement over original Procedure Eventual Reliability using the Arithmetic Mean of Basic SAFPHR℞'s Range.

Task	Task General Reliability	New Prob.	Improvement
"Is Valid and Legal?"	**0.0573620**	0.0142598	0.0001735
"Contact Prescriber"	**0.0160587**	0.0144333	< 1e-07
"Document changes" (validity and legality issues)	0.0001109	0.0144295	0.0000037
"Is Appropriate and Safe?"	**0.0528028**	0.0144154	0.0000178
"Contact Patient and/or Review history" (appropriateness and safety issues)	0.0009876	0.0144333	< 1e−07
"Contact Prescriber" (appropriateness and safety issues)	0.0099080	0.0144333	< 1e−07
"Document Changes" (appropriateness and safety issues)	0.0009867	0.0144300	0.0000032
"Data Entry"	0.0090344	0.0140826	0.0003506
"Print Label"	0.0090849	0.0112709	**0.0031624**
"Get Stock Bottle"	**0.0921102**	0.0143964	0.0000369
"Bottle Passes NDC check?"	**0.0109651**	0.0143964	0.0000369
"Count Medication"	**0.0787577**	0.0105909	**0.0038424**
"Attach Label"	**0.0782910**	0.0141011	0.0003321
"Attach Auxiliary Label"	**0.0180332**	0.0143670	0.0000663
"Is Valid and Legal?" (final check, validity and legality issues)	**0.0662771**	0.0142611	0.0001721
"Contact Prescriber" (final contact, validity and legality issues)	0.0008269	0.0144333	< 1e−07
"Document Changes" (final changes, validity and legality issues)	0.0000117	0.0144300	0.0000032
"Is Appropriate and Safe?" (final check, appropriateness and safety issues)	0.0081712	0.0144131	0.0000202
"Contact Patient" (final contact, appropriateness and safety issues)	0.0001395	0.0144333	< 1e−07
"Contact Prescriber" (final contact, appropriateness and safety issues)	0.0002219	0.0144333	< 1e−07
"Document Changes" (final changes, appropriateness and safety issues)	0.0000194	0.0144325	0.0000007
"Is Correct Label Attached?"	0.0008214	0.0140826	0.0003506
"Is Label Data Correct?"	0.0007329	0.0141011	0.0003321
"Is Drug Correct"	0.0000798	0.0143959	0.0000373
"Is quantity of Drug Correct?"	0.0063675	0.0105909	**0.0038424**
"Is Correct Auxiliary Label Attached?"	0.0001140	0.0143646	0.0000686
"Give Filled ℞ to Patient"	0.0050662	0.0112709	**0.0031624**
"Deliver Counseling to Patient"	0.0032359	0.0112709	**0.0031624**

Note. Bold entries represent values from each column that are orders of magnitude larger than non-bolded entries. "Improvement" column entries are calculated based on the improvement over the original procedure eventual reliability of the corresponding "New Prob." entry.

First, Zheng et al. (2020) offered a property pattern for assessing the general reliability (the probability that a task ever is done incorrectly) of a give task (g) as

Task General Reliability:

$$P = ?\left[\mathbb{F}\begin{pmatrix}(ProcedureStep = NextStep)\\ \wedge(g = Incorrect)\end{pmatrix}\right]. \quad (6)$$

Note that here *NextStep* represents the task (or tasks) that immediately follow task g in the procedure model. Further note that this property does not account for the fact that error can be corrected due to feedback and checking within the procedure. Thus, task general reliability is best used to assess procedure efficiency rather than identify reliability-related interventions. To allow such evaluations, Zheng et al. (2020) shows that analysts can modify the system model so that a given task will never be performed incorrectly. The analyst then checks the new model for procedure eventual reliability (Eq. (1)) and compares the original value to the new one to determine how much improvement resulted from the change.

Thus, using the arithmetic mean of basic SAFPHR℞'s predicted ranges, we calculated the task general reliability of each task, the procedure eventual reliability when each task was always performed correctly (New Prob.) and its associated improvement (Improve) from the original reliability. These results are shown in Table 5.

Because of the way it is calculated, the arithmetic average rates were universally higher than their geometric mean counterparts from Zheng et al. (2020). However, the tasks that produced the highest predicted values (based on their relative orders of magnitude to the other results) were largely consistent, with a few key differences. All of the tasks that produced the highest error rates for the geometric mean doing so for the arithmetic mean. However, "Bottle Passes NDC Check?" is additionally included in this list based on the arithmetic average results (Table 5; it was not included in the geometric mean results). The relative magnitude of the predicted probability also varied slightly in the results, where pharmacists hoping to address inefficiencies in their process would prioritize interventions (based on magnitude the probability of a task general reliability failure) as follows: (1)"Get Stock Bottle," (2) "Count Medication," (3) "Attach Label," (4) "Is Valid and Legal?" (final check, validity and legality issues), (5) "Is Valid and Legal?," (6) "Is Appropriate and Safe?," (7) "Attach Auxiliary Label," (8) "Contact Prescriber," and (9) "Bottle Passes NDC Check?".

The results of the "Improvement" metric (Table 5, which uses New Prob.) is helpful for exploring the effectiveness of interventions. In fact, the highlighted tasks whose correction would most improve medication error rates are exactly the same for the arithmetic mean results as for the geometric mean results (Zheng et al., 2020). However, the internal rankings of the associated tasks (based on the amount of improvement) were different. Specifically, in the new results "Count Medication" and "Quantity of Drug Correct?" appear to have the highest effect on final reliability. This is followed by "Print Label," "Give Filled ℞ to Patient," and "Deliver Counseling to Patient," which all saw the same amount of improvement. In the original results, the order of these two groupings was reversed. Thus, according to the new results, pharmacists could focus on improving any of these factors to significantly improve pharmacy dispensing reliability, but the most significant improvements could be seen through the elimination of errors at "Count Medication" and "Quantity of Drug Correct?." This suggests that investment in automated dispensing technology could have a profound impact on community pharmacy dispensing.

6.2. Diagnosing CPC-Effect Extended SAFPHR℞ Performance

The results for the extended SAFPHR℞ analyses are somewhat surprising given that we would expect the additional information required by these analyses to improve estimates, not make them more inaccurate. This is particularly true of CPC-effect extended SAFPHR℞, where individual CPC ratings are used to refine error rates.

Thus, to determine why CPC-effect extended SAFPHR℞ yields

Table 6
Step by step simulation analyses using CPC-effected extended SAFPHR.

		CPCWeighting				ProbError			
Single Task	CFP	W_1	W_2	$W_{3,4,6}$	$W_{5,7,8}$	t = 1	t = 2	t = 3 | 4 |6	t = 5 | 7 | 8
"Is Valid and Legal?"	0.2	2.5	0.25	0.25	0.15	**0.5**	0.05	0.05	0.03
"Contact Prescriber"	0.0005	16	1.6	1.6	0.96	0.008	0.0008	0.0008	0.00048
"Document changes" (validity and legality issues)	0.03	1.28	0.128	0.256	0.1536	0.0384	0.00384	0.00768	0.004608
"Is Appropriate and Safe?"	0.2	5	0.5	0.5	0.3	**1**	**0.1**	**0.1**	0.06
"Contact Patient and/or Review history"	0.0005	0.64	0.064	0.128	0.0768	0.00032	0.000032	0.000064	0.0000384
"Contact Prescriber" (appropriateness and safety issues)	0.0005	8	0.8	0.8	0.48	0.004	0.0004	0.0004	0.00024
"Document Changes"(appropriateness and safety issues)	0.03	0.64	0.064	0.128	0.0768	0.0192	0.00192	0.00384	0.002304
"Data Entry"	0.03	1.28	0.128	0.256	0.1536	0.0384	0.00384	0.00768	0.004608
"Print Label"	0.03	0.64	0.064	0.128	0.0768	0.0192	0.00192	0.00384	0.002304
"Get Stock Bottle"	0.0005	8	0.8	0.8	0.48	0.004	0.0004	0.0004	0.00024
"Bottle Passes NDC check?"	0.2	5	0.5	0.5	0.3	**1**	**0.1**	**0.1**	0.06
"Count Medication"	0.03	10	1	1	0.6	0.03	0.003	0.003	0.0018
"Attach Label"	0.003	8	0.8	0.8	0.4767	0.024	0.0024	0.0024	0.00143
"Attach Auxillary Label"	0.03	1.6	0.16	0.32	0.192	0.048	0.0048	0.0096	0.00576
"Is Valid and Legal?" (final check)	0.2	5	0.5	1	0.6	**1**	**0.1**	**0.2**	**0.12**
"Contact Prescriber" (final contact, validity and legality issues)	0.03	0.64	0.064	0.128	0.0768	0.0192	0.00192	0.00384	0.002304
"Document Changes" (final changes, validity and legality issues)	0.003	8	0.08	0.16	0.96	0.024	0.00024	0.00048	0.00288
"Is Appropriate and Safe?" (final check)	0.2	1	0.1	0.2	0.12	**0.2**	0.02	0.04	0.024
"Contact Patient" (final contact, appropriateness and safety issues)	0.03	0.64	0.064	0.128	0.0768	0.0192	0.00192	0.00384	0.002304
"Contact Prescriber" (final contact, appropriateness and safety issues)	0.03	0.64	0.064	0.128	0.0768	0.0192	0.00192	0.00384	0.002304
"Document Changes" (final changes, appropriateness and safety issues)	0.003	0.64	0.064	0.128	0.0768	0.00192	0.000192	0.000384	0.0002304
"Is Correct Label Attached?"	0.2	1	0.1	0.2	0.12	**0.2**	0.02	0.04	0.024
"Is Label Data Correct?"	0.2	5	1	1	0.6	**1**	**0.2**	**0.2**	**0.12**
"Is Drug Correct"	0.2	1	0.1	0.2	0.12	**0.2**	0.02	0.04	0.024
"Is Quantity of Drug Correct?"	0.2	5	1	1	0.6	**1**	**0.2**	**0.2**	**0.12**
"Is Correct Auxiliary Label Attached?"	0.2	1	0.1	0.2	0.12	**0.2**	0.02	0.04	0.024
"Give Filled R to Patient"	0.0005	0.64	0.064	0.128	0.0768	0.00032	0.000032	0.000064	0.0000384
"Deliver Counseling to Patient"	0.03	0.64	0.064	0.128	0.0768	0.0192	0.00192	0.00384	0.002304

Note that entries in red represent the weighting factors that increase the probability of error. Bold entries in both blue and black are probabilities bigger than 0.1, blue bold entries are those reach to 1.

predictions that are much bigger than the other methods (and the literature data), we used PRISM to manually steps through our model in order to identify the value of model variables under specific model conditions. Specifically, to give us insights into how a single task impacted the procedure's reliability under different situations, this allowed us to observer the probability of error (`ProbError` from Fig. 12) for each task at different time periods (`t`; from Fig. 7). In doing this, we also determined what the weighting factors were for each task (`CPCWeighting`; from Fig. 12) at each period to gain insight into how the large error rates were produced.

The simulation results are reported in Table 6. The probabilities of an error occurring in each task at different time periods are reported under "ProbError". "CFP" represents the associated nominal cognitive failure probability of the cognitive function failure assessed by our subject matter expert for each task. W_1 and W_2 represent the weighting factors required to adjust the CFP under time period $t = 1$ and $t = 2$, respectively. Similarly, $W_{3,4,6}$ represent the weighting factors used to adjust the CFP under time period $t = 3$, $t = 4$, and $t = 6$. Finally, $W_{5,7,8}$ represents the weighting factors used under time periods $t = 5$, $t = 7$, and $t = 8$. Note that time periods were grouped together because they produced identical results.

These results show that there are 39 error rates predicted to be greater than 0.1 over the 224 cases. Thus, even with the feedback steps (the decision tasks in Fig. 3) that can detect and send errors back to be corrected, these high error rates can accumulate and lead to high overall error rates. Specifically, in time period `t` = 1, the error rates of "Is Appropriate and Safe?," "Bottle Passes NDC check?," "Is Valid and Legal?" (final check, validity and legality issues), "Is Label Data Correct?," and "Is Quantity of Drug Correct?" are all 1. All of these decision-related tasks will have significant impacts on procedure reliability. For example, the `ProbError` under step "Bottle Passes NDC check?" being 1 means that an error made during "Get Stock Bottle" can never be detected. The error will always be passed through the NDC check to the next step. Conversely, a correct drug bottle selection will always circulate back to "Get Stock Bottle" until an error occurs. As such, the decision-related tasks will never be able to effectively check the correctness of the factors they are designed to validate. Even worse, with these invalid "checking" steps, new errors are introduced in "Data Entry," "Count Medication," and "Attach Label" when the prescription is sent back. The results also show that probabilities of 1 are caused by large weighting factors. The weighting values responsible for this effect are inherently part of CREAM. Because they were originally derived from data collected in the nuclear power field, they may not accurately transfer to pharmacy tasks. Thus, future research should focus on determining how to calibrate these weighting values to the pharmacy domain. This could potentially improve the accuracy of extended SAFPHR.

6.3. Generalizability

The results presented here were all derived using the ratings of one subject matter expert who both a practicing pharmacist, a professor of pharmacy practice, and intimately familiar with community pharmacy procedures in multiple states. The use of a single expert is consistent with CREAM, on which SAFPHR's probabilistic estimates are based. That we achieve results that were remarkably consistent with the most comprehensive study we could find in the literature speaks to the generalizability of our findings. However, it is true that there would likely be variation in the assessments offered by different pharmacists. Unfortunately, CREAM's documentation does not provide any guidance for how to aggregate assessments across a population. Future research should investigate how population assessments can be incorporated into SAFPHR.

This said, it is important to note that SAFPHR was originally designed as a tool that individual pharmacies could use to understand and improve their reliability without the need for long, expensive,

observational studies (Zheng et al., 2020). The application of both basic and extended SAFPHR presented here was modeled after nominal community pharmacy dispensing. There can be a variety of dispensing procedures. Thus, future work should seek to apply and evaluate the performance of SAFPHR in a number of different operational environments.

6.4. Scalability

One of the biggest limitations of model checking is a combinatorial explosion (sometimes called the state explosion problem; (Katoen, 2010)). Specifically, as concurrent elements are added to a formal model, the size of the model's statespace grows exponentially (Clarke et al., 1999). This can result in models that are too big or take too long to be formally verified. In basic SAFPHR analysis, it took 33.098 and 153.238 s to get the lower and upper bound procedure eventual reliability error rates respectively (186.336 s total). This time is similar to those we observed for performing comparable extended SAFPHR analyses: 210.671 s for CPC-effect extended SAFPHR and 920.884 s for mode-effect extended SAFPHR. All of these times are very reasonable for analyzing a full pharmacy procedure. Thus, the choice of SAFPHR technique does not appear to be significantly impacted by scalability concerns.

Even if it is unlikely that scale would present a significant problem for other pharmacy environments, it is conceivable that it could limit the applicability of SAFPHR for other more complex domains. Given that our results show that the arithmetic average produced the most accurate predictions, it may be possible to use averages of control mode ranges (Fig. 2) in SAFPHR model implementations. Such an effort could avoid the need to run two verifications for every error rate produced with basic SAFPHR. This should be explored in future work. Additionally, PRISM also supports a number of methods for improving scalability. This includes different analysis engines that are more efficient for different types of models (Parker, 2003). It also has support for statistical model checking (Kwiatkowska et al., 2011), an approach that can be used to get approximate results in situations where scalability is a constraint. Future research should investigate how the different features of PRISM influence performance/accuracy trade-offs with SAFPHR.

6.5. Human reliability analysis

As with basic SAFPHR, extended SAFPHR has implications for HRAs in general. By addressing the major limitations of first- and second-generation HRAs, this work shows that HRA can be made to account for dynamic system behaviors at a level that was not previously possible. While SAFPHR has specifically been developed for use with community pharmacies, there is no reason it could not be applied to other pharmacy environments or other safety–critical domains. This should be the subject of future work.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 1918314.

References

Allan, E.A., Barker, K.N., 1990. Fundamentals of medication error research. Am. J. Health-Syst. Pharm. 47, 555–571.
Allison, P.D., 2014. Measures of fit for logistic regression, in. In: Proceedings of the SAS Global Forum 2014 Conference, pp. 1–13.
Ashcroft, D.M., Morecroft, C., Parker, D., Noyce, P.R., 2006. Likelihood of reporting adverse events in community pharmacy: An experimental study. Qual. Saf. Health Care 15, 48–52.
Bedford, T., Bayley, C., Revie, M., 2013. Screening, sensitivity, and uncertainty for the CREAM method of human reliability analysis. Reliab. Eng. Syst. Saf. 115, 100–110.
Bell, J., Holroyd, J., 2009. Review of human reliability assessment methods. Technical Report RR679. Health and Safety Executive. Derbyshire.
Berdot, S., Gillaizeau, F., Caruba, T., Prognon, P., Durieux, P., Sabatier, B., 2013. Drug administration errors in hospital inpatients: a systematic review. PloS one 8, e68856.
BlueShield, B., 2019. The value of rx integration. URL https://www.bluecrossnc.com/sites/default/files/U20694_TheValueOfRxIntegration.pdf.
Bolton, M.L., 2017. Novel developments in formal methods for human factors engineering, in: Proceedings of the human factors and ergonomics society annual meeting, SAGE Publications Sage CA: Los Angeles, CA. pp. 715–717.
Bolton, M.L., Bass, E.J., Siminiceanu, R.I., 2013. Using formal verification to evaluate human-automation interaction in safety critical systems, a review. IEEE Trans. Syst., Man Cybernet.: Syst. 43, 488–503.
Boring, R.L., Dudenhoeffer, D.D., Hallbert, B.P., Gore, B.F., 2006. Modeling human reliability analysis using MIDAS. Technical Report. Idaho National Laboratory (INL).
Bye, A., Hollnagel, E., Brendeford, T.S., 1999. Human–machine function allocation: a functional modelling approach. Reliab. Eng. Syst. Saf. 64, 291–300.
Castiglia, F., Giardina, M., Caravello, F., 2008. Fuzzy fault tree analysis in modern γ-ray industrial irradiator: Use of fuzzy version of heart and cream techniques for human error evaluation. In: 9th International Conference on Probabilistic Safety Assessment and Management, PSAM9, Kao, T. pp. 1377–1384.
Clarke, E.M., Grumberg, O., Peled, D., 1999. Model checking. MIT press.
Copas, J., 1989. Unweighted sum of squares test for proportions. J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) 38, 71–80.
De Felice, F., Petrillo, A., Carlomusto, A., Ramondo, A., 2012. Human reliability analysis: A review of the state of the art. IRACST–Int. J. Res. Manage. Technol. (IJRMT) 2.
Desmorat, G., Guarnieri, F., Besnard, D., Desideri, P., Loth, F., 2013. Pouring cream into natural gas: The introduction of common performance conditions into the safety management of gas networks. Saf. Sci. 54, 1–7.
Di Pasquale, V., Iannone, R., Miranda, S., Riemma, S., 2013. An overview of human reliability analysis techniques in manufacturing operations. In: Schiraldi, M. (Ed.), Operations management. InTech, pp. 221–240.
Di Pasquale, V., Iannone, R., Miranda, S., Riemma, S., 2013. An overview of human reliability analysis techniques in manufacturing operations. In: Operations management. IntechOpen, pp. 221–240.
Flynn, E.A., Barker, K.N., Carnahan, B.J., 2003. National observational study of prescription dispensing accuracy and safety in 50 pharmacies. J. Am. Pharmaceut. Assoc. 43, 191–200.
Fujita, Y., Hollnagel, E., 2004. Failures without errors: quantification of context in HRA. Reliab. Eng. Syst. Saf. 83, 145–151.
Geng, J., Murè, S., Baldissone, G., Camuncoli, G., Demichela, M., 2015. Human error probability estimation in ATEX-HMI area classification: From THERP to FUZZY CREAM. Chem. Eng. Trans. 43, 1243–1248.
Hannaman, G., Spurgin, A., Lukic, Y., 1984. Human cognitive reliability model for PRA analysis. Technical Report NUS-4531. Electric Power Research Institute. Palo Alto.
He, X., Wang, Y., Shen, Z., Huang, X., 2008. A simplified CREAM prospective quantification process and its application. Reliab. Eng. Syst. Saf. 93, 298–306.
Hollnagel, E., 1998a. Cognitive Reliability and Error Analysis Method (CREAM). Elsevier, Oxford.
Hollnagel, E., 1998b. Context, cognition and control. In: Waern, Y. (Ed.), Co-operative Process Management, Cognition and Information Technology. Taylor & Francis, London, pp. 27–52.
Hollnagel, E., Kaarstad, M., Lee, H.C., 1999. Error mode prediction. Ergonomics 42, 1457–1471.
Hosmer, D.W., Hosmer, T., Le Cessie, S., Lemeshow, S., 1997. A comparison of goodness-of-fit tests for the logistic regression model. Stat. Med. 16, 965–980.
Katoen, J.P., 2010. Advances in probabilistic model checking. In: International Workshop on Verification, Model Checking, and Abstract Interpretation. Springer, Berlin, pp. 25.
Kaushal, R., Bates, D.W., Landrigan, C., McKenna, K.J., Clapp, M.D., Federico, F., Goldmann, D.A., 2001. Medication errors and adverse drug events in pediatric inpatients. Jama 285, 2114–2120.
Kim, J.W., Jung, W., 2003. A taxonomy of performance influencing factors for human reliability analysis of emergency tasks. J. Loss Prev. Process Ind. 16, 479–495.
Kim, M.C., Seong, P.H., Hollnagel, E., 2006. A probabilistic approach for determining the control mode in CREAM. Reliab. Eng. Syst. Saf. 91, 191–199.
Kwiatkowska, M., Norman, G., Parker, D., 2007. Stochastic model checking. In: SFM. Springer, pp. 220–270.
Kwiatkowska, M., Norman, G., Parker, D., 2011. PRISM 4.0: Verification of probabilistic real-time systems. In: International Conference on Computer Aided Verification. Springer, pp. 585–591.
Lao, K.S., Chui, C.S., Man, K.K., Lau, W.C., Chan, E.W., Wong, I.C., 2016. Medication safety research by observational study design. Int. J. Clin. Pharm. 38, 676–684.
Lee, S.W., Kim, A.R., Ha, J.S., Seong, P.H., 2011. Development of a qualitative evaluation framework for performance shaping factors (PSFs) in advanced MCR HRA. Ann. Nucl. Energy 38, 1751–1759.
Odukoya, O.K., Schleiden, L.J., Chui, M.A., 2015. The hidden role of community pharmacy technicians in ensuring patient safety with the use of e-prescribing. Pharmacy 3, 330–343.
Pape, T.M., 2001. Searching for the final answer: factors contributing to medication administration errors. J. Continuing Educ. Nurs. 32, 152–160.
Parker, D., Norman, G., Kwiatkowska, M., 2017. PRISM Manual. Technical Report. University of Oxford.
Parker, D.A., 2003. Implementation of symbolic model checking for probabilistic systems. Ph.D. thesis. University of Birmingham.
Parker, J. (Ed.), 2013. Meeting the Joint Commission's 2013 National Patient Safety Goals. Oakbrook Terrace. URL http://www.healtheast.org/images/stories/patient-safety/2014_jc_patient_safety_goals.pdf.

Rantanen, E., Deeter, J., Burke, S., Wang, Y., 2012a. Analysis of hospital oncology department pharmacy operations. In: 2012 Symposium on Human Factors and Ergonomics in Health Care. HFES, Santa Monica, pp. 27–34.
Rantanen, E., Deeter, J., Burke, S., Wang, Y., 2012b. Human factors evaluation of pharmacy operations. In: 2012 symposium on human factors and ergonomics in health care. HFES, Santa Monica.
Reer, B., 2008. Review of advances in human reliability analysis of errors of commission part 2: EOC quantification. Reliab. Eng. Syst. Saf. 93, 1105–1122.
Swain, A., 1987. Accident Sequence Evaluation Program Human Reliability Analysis Procedure. Technical Report NUREG/CR-4772. US Nuclear Regulatory Commission. Washington, DC.
Swain, A.D., Guttmann, H.E., 1983. Handbook of human-reliability analysis with emphasis on nuclear power plant applications: Final report. Technical Report 7011944. Sandia National Labs. Albuquerque.
Szeinbach, S., Seoane-Vazquez, E., Parekh, A., Herderick, M., 2007. Dispensing errors in community pharmacy: perceived influence of sociotechnical factors. Int. J. Qual. Health Care 19, 203–209.
The Joint Commission, 2014. 2014 Hospital National Patient Safety Goals. Technical Report. The Joint Commission. URL http://www.healtheast.org/images/stories/patient-safety/2014_jc_patient_safety_goals.pdf.
The Joint Commission, 2015. 2015 Hospital National Patient Safety Goals. Technical Report. The Joint Commission. URL http://www.jointcommission.org/assets/1/6/2015_HAP_NPSG_ER.pdf.
The Joint Commission, 2016a. 2016 Hospital National Patient Safety Goals. Technical Report. The Joint Commission. URL https://www.jointcommission.org/assets/1/6/2016_NPSG_HAP_ER.pdf.
The Joint Commission, 2016b. National Patient Safety Goals Effective January 2017. Technical Report. The Joint Commission. URL https://www.jointcommission.org/assets/1/6/NPSG_Chapter_HAP_Jan2017.pdf.
Ung, S.T., 2018. Human error assessment of oil tanker grounding. Saf. Sci. 104, 16–28.
Weyers, B., Bowen, J., Dix, A., Palanque, P., 2017. The handbook of formal methods in human-computer interaction. Springer.
Williams, J., 1986. Heart–a proposed method for assessing and reducing human error. In: 9th Advances in Reliability Technology Symposium. University of Bradford.
Williams, J., 1988. A data-based method for assessing and reducing human error to improve operational performance. In: Human Factors and Power Plants, 1988, Conference Record for 1988 IEEE Fourth Conference on, IEEE. pp. 436–450.
Wilson, D.G., McArtney, R., Newcombe, R.G., McArtney, R., Gracie, J., Kirk, C., Stuart, A., 1998. Medication errors in paediatric practice: insights from a continuous quality improvement approach. Eur. J. Pediatr. 157, 769–774.
Wing, J.M., 1990. A specifier's introduction to formal methods. Computer 23 (8), 10–22 24.
Witte, D., Dundes, L., 2007. Prescription for error: process defects in a community retail pharmacy. J. Patient Saf. 3, 190–194.
Wreathall, J., 1982. Operator action trees, an approach to quantifying operator error probability during accident sequences. NUS report 4159.
Zheng, X., 2020. The Development of A Next-Generation Human Reliability Analysis: Systems Analysis for Formal Pharmaceutical Human Reliability (SAFPHR) (Doctoral dissertation). Ph.D. thesis. University at Buffalo.
Zheng, X., Bolton, M.L., Daly, C., Biltekoff, E., 2020. The development of a next-generation human reliability analysis: Systems analysis for formal pharmaceutical human reliability (SAFPHR). Reliab. Eng. Syst. Safety 202. https://doi.org/10.1016/j.ress.2020.106927.
Zheng, X., Bolton, M.L., Daly, C., Feng, L., 2017. A formal human reliability analysis of a community pharmacy dispensing procedure, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, SAGE Publications Sage CA: Los Angeles, CA. pp. 728–732.
Zhou, Q., Wong, Y.D., Loh, H.S., Yuen, K.F., 2018. A fuzzy and bayesian network cream model for human reliability analysis–the case of tanker shipping. Saf. Sci. 105, 149–157.
Zhou, Q., Wong, Y.D., Xu, H., Van Thai, V., Loh, H.S., Yuen, K.F., 2017. An enhanced cream with stakeholder-graded protocols for tanker shipping safety application. Saf. Sci. 95, 140–147.