International Journal of Industrial Ergonomics xxx (2016) 1–14

Contents lists available at ScienceDirect

International Journal of Industrial Ergonomics

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

Properties for formally assessing the performance level of human-human collaborative procedures with miscommunications and erroneous human behavior☆

Dan Pan [a], Matthew L. Bolton [b, *]

[a] Department of Industrial Engineering, Tsinghua University, Beijing 100084, PR China
[b] Department of Industrial and Systems Engineering, University at Buffalo, State University of New York, Amherst, NY 14260, USA

ARTICLE INFO

Article history:
Received 6 June 2015
Received in revised form
22 January 2016
Accepted 4 April 2016
Available online xxx

Keywords:
Human-human collaboration
Human communication
Task analysis
Model checking
Human error

ABSTRACT

Human-human interaction and collaboration is crucial to teamwork, where team members work together to perform tasks and share information to ensure mutual understanding. Human-human collaborative procedures are developed to ensure that relevant information is correctly heard and actions are correctly executed. Such procedures should be designed to be robust to miscommunications and other erroneous human behaviors. However, such procedures can be complex and thus fail in ways not anticipated by designers. To address this, previous efforts have used formal proof analyses to assess the robustness of collaborative procedures to miscommunications. However, these analyses only indicate strict success or failure: outcomes that fail to capture the degrees of success of collaborative procedures. Further, none of these analyses considered the interaction between miscommunications and other erroneous human behaviors. In this paper, we create specification properties to evaluate the level of success of a collaborative procedure formally. We demonstrate the use of these properties to formally evaluate realistic collaborative procedures from a nuclear power plant with and without both generated miscommunications and erroneous human behavior. We discuss the results of this evaluation and outline area of future research.

Relevance to industry: The method, performance levels, and associated specification properties allow analysts to compare the robustness of different collaborative procedures to miscommunication and attentional slips. The power of our approach is demonstrated with the nuclear power plant application. It can be easily adapted for use with collaborative procedures from other domains.

1. Introduction

With the development of industrial technology, safety-critical systems in nuclear power plants (NPP), the chemical process industry, and air transportation have become more complex. As such, their safe operation depends not only on the individual skills and knowledge or human operators, but also effective and efficient team communication and collaboration. In these sensitive systems, failures can be associated with the erroneous behavior of individual human operators (Reason, 1990) as well as human-human collaboration. For example, it is reported that in Germany, communication errors are responsible for about 10% of the workplace incidents resulting from human error (Sträter, 2003).

Of particular interest to this work is the main control room (MCR) of nuclear power plants (NPPs), where communications and collaboration among operators are essential factors for understanding how and how well MCR operators deal with abnormal or emergency situations. In particular, the performance of MCR crew under abnormal/emergency situations in NPPs is strongly affected not only by operators' cognitive processes, but also by communication and collaboration among operators. Communication error has been considered as one of the main causes of accidents and incidents in NPPs. Hirotsu et al. (2001) reported that in Japanese

☆ Note that *this paper is an extended version of a conference paper* (Pan and Bolton, 2015) *that was presented at HCI International* 2015. This article constitutes a significant contribution beyond this original manuscript. Specifically, it includes erroneous human behavior in the analyses; presents verification results (statespace size and verification times) for each analysis; and evaluates a new collaborative procedure (procedure 2) so that it can be compared to the original procedures (procedure 1). It also features an extended discussion.

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

http://dx.doi.org/10.1016/j.ergon.2016.04.001
0169-8141/© 2016 Elsevier B.V. All rights reserved.

NPPs, 25% of human error incidents were due to communication failure. Sträter (2003) investigated 232 operational events involving human error in German NPPs and found that roughly 10% of them which involve human errors were mainly caused by communication problems. Similar results have been observed in ground transportation (Murphy, 2001), medicine (Wilson et al., 1995), and aviation (Connell, 1996).

From these investigations and analyses, we can conclude that maintaining reliable communication and human behavior is essential to secure the safety of large, complex systems. If team members could perform and collaborate better, the safety of many systems would be improved. So far, standard human-human collaborative procedures and communication protocols have been used to ensure effective and efficient collaboration in many safety-critical systems. For example, operation crews in MCR of nuclear power plants use communication protocols to diagnose problems and execute emergency operations (Kim et al., 2010). However, there is concurrency between human operators and parts of procedures. The concurrency creates complexity and thus potentially induces unanticipated interactions between operators. Further, humans are fallible. They can perform protocols erroneously by incorrectly performing their parts of the procedure or by miscommunicating information to team member. Therefore, it can be difficult to evaluate the safety of human-human collaborative procedures using conventional analyses methods, like experimentation and simulation that can miss unexpected conditions and interactions.

Formal methods offer proof-based analysis techniques capable of considering all possible interactions. While formal methods have been used to evaluate machine communication protocols, the existing approaches (Bochmann and Sunshine, 1980; Sidhu and Leung, 1989) are ill-suited for use with human-human collaborative procedures for several reasons. First, humans behave in different ways from machines. Humans follow tasks as opposed to machine code and human-human communication must be contextualized as part of a task (Traum and Dillenbourg, 1996). Second, humans are more flexible than machines and are thus fallible in different ways. Third, human collaborative procedures are inherently less fragile than machine communication protocols because of the looser dynamics of human-human communication. As such, the outcome of human-human collaboration may represent degrees of success beyond a simple binary one (correct or incorrect). For example, if two persons are attempting to collaboratively diagnose a problem, it is problematic if they end up with only one reaching the correct conclusion. However, this is better than if both reach the same incorrect conclusion because the incorrect conclusion has a better chance of being identified and corrected as humans continue to collaborate.

Procedures for both collaborative and non-collaborative situations have been assessed formally to determine if they are safe, even with generated erroneous behavior and/or miscommunications (Bolton et al., 2013; Bolton, 2015). However, these analyses are still limited in that they have not considered the way erroneous human behaviors (e.g., attention error) and miscommunications can interact. Further, much like machine communication protocols, they only consider the binary success of human-human collaboration. This is constraining because it does not give analysts the tools they need to fully evaluate the robustness of such procedures. Therefore, an approach is needed to account for both individual erroneous behavior and miscommunication between individuals while giving analysts metrics for assessing the degrees of a procedure's success in different conditions.

In this paper, the approach in Bolton (2015) is extended to allow an analyst to model human collaborative procedures in the context of a task analytic modeling formalism and use model checking to evaluate the degrees of a procedure's success even with erroneous human behaviors. Before presenting the method, we cover the background material necessary for understanding it. We then present the method and describe how it was realized. In doing this, we use an NPP diagnosis case study to frame our analyses and illustrate how our approach can be applied to the evaluation of realistic safety-critical, human-human collaborative procedures. Finally, our results and future works are discussed.

2. Background

2.1. Formal method

Formal methods are tools and techniques for proving that a system will always perform as intended (Clarke and Wing, 1996). Model checking is an automated means of performing formal verification, checking whether a system model adheres to specifications (Clarke et al., 1999). A system model is a representation of a system's behavior in a mathematical formalism such as a finite state machine. A specification is a formal description of a desirable property about the system, usually in a temporal logic. Model checking works by exhaustively searching a model's statespace for violations of the specification. The result of this is documented in a verification report which contains either a confirmation if the model adheres to the specification or a counterexample if it does not. A counterexample lists the incremental model states that resulted in the specification being violated. This can be used by analysts to address the discovered failure.

There are a variety of temporal and modal logics that have been used to express specifications. The most common one, and the one used in the presented work, is linear temporal logic (LTL) (Emerson, 1990). LTL allows one to reason about the relationship between different states and/or variables over ordinal time and assert properties about all of the paths through a model. It does this using model variables; basic Boolean logic operators including $\wedge$, $\vee$, $\neg$, $\Rightarrow$, and $\Leftrightarrow$; and temporal operators (Table 1).

While formal methods have traditionally been used in the analysis of computer hardware and software systems, a growing body of work has been investigating how to use them to evaluate human factors issues (Bolton et al., 2013). However, when it comes to issues of human-human communication and coordination, there has been very little work. The Concur Task Trees formalism (Paternò et al., 1997) has been extended to allow for the modeling of human-human coordination and communication, where communications could have different modalities (synchronous or asynchronous, point-to-point, or broadcast), and used to formally evaluate pilot and air traffic control radio communications during runway operations using different shared task representations (Paternò et al., 1998). Although this method is useful, it did not easily distinguish between separate and shared operator tasks, nor did it account for potential miscommunications and operator perceptual or cognitive errors. Both limitations were addressed by the Enhanced Operator Function Model with Communications (EOFMC).

Table 1
Temporal operators of LTL for specification.

Name	Operator	Interpretation
Global	$G\,\Phi$	Φ will always be true.
Next	$X\,\Phi$	Φ will be true in all next states.
Future	$F\,\Phi$	Φ will eventually be true.
Until	$\Phi\,U\,\Psi$	Φ will be true until Ψ is true.

Please cite this article in press as: Pan, D., Bolton, M.L., Properties for formally assessing the performance level of human-human collaborative procedures with miscommunications and erroneous human behavior, International Journal of Industrial Ergonomics (2016), http://dx.doi.org/10.1016/j.ergon.2016.04.001

2.2. Enhanced operator function model with communication

Enhanced Operator Function Model (EOFM) (Bolton et al., 2011) was extended to EOFMC (Bass et al., 2011) to support the modeling of human-human communication and coordination as shared task structures between human operators. Specifically, EOFMC represents groups of human operators engaging in shared activities with an input/output system. Inputs represent the human interface, environment, and/or mission goal concepts. Outputs are human actions. The operators' task models (local variables) describe how human actions are produced (i.e., task behavior, and inner group coordination and communication) and how the internal state of the human changes (i.e., perceptual or cognitive processing).

Each task in an EOFMC is a goal directed activity that decomposes into other goal directed activities and, ultimately, atomic actions. Tasks can either belong to one human operator, or they can be shared between human operators. A shared task is explicitly associated with two or more human operators, making it clear which human operators perform each part of a task.

Activities can have preconditions, repeat conditions, and completion conditions (collectively referred to as strategic knowledge). These are represented by Boolean expressions written in terms of input, output, and local variables as well as constants. They specify what must be true before an activity can execute (precondition), when it can execute again (repeat condition), and what is true when it has completed execution (completion condition).

An activity's decomposition has an operator that specifies how many sub-activities or actions can execute and what the temporal relationship is between them. In the presented work, the following decomposition operators are involved:

`sync` – all actions must be performed synchronously (at the exact same time);
`xor` – exactly one action must be performed;
`and_seq` – all of the actions must be performed, one at a time, in any order;
`ord` – all of the actions must be performed, one at a time, in the order listed; and.
`com` – a communication action is performed (this is discussed subsequently).

Actions occur at the bottom of EOFMC task hierarchies. Actions are either an assignment to an output variable (indicating an action has been performed) or a local variable (representing a perceptual, cognitive, or communication action). Meanwhile, decomposition can specify how many sub-activities or actions can execute and what the temporal relationship is between them. Shared activities can explicitly include human-human communication using the `com` decomposition. In such decompositions, communicated information from one human operator can be received by other human operators (modeled as an update to a local variable). By exploiting the shared activity and communication action feature of EOFMC, human-human communication protocols can be modeled as shared task activities.

EOFMC has formal semantics that specify how an instantiated EOFMC model executes. Each activity or action has one of three execution states: Ready (waiting to execute), Executing, and Done. An activity or action transitions between states based on the state of itself, its parent activity (if it has one), the other actions in the given decomposition, the children that decompose from it, and its strategic knowledge.

These semantics are the basis for the EOFMC translator (Bolton et al., 2011; Bolton, 2015) that allows EOFMC models to be automatically incorporated into the input language of the Symbolic Analysis Laboratories (SAL) family of model checkers (De Moura et al., 2004). The basic principle behind this translation is that a finite state machine is created representing the behavior of the human operator team. Specifically, the module is input variables representing the EOFMC's inputs, outputs as variables representing the human operator actions, and local variables representing the internal state of the model. These local variables can be explicitly defined in the EOFMC XML markup (where assignments to these variables represent perceptual or cognitive actions) as well as variables representing every activity's and action's execution state. Transitions in this model are created that describe how the execution state of activities and actions change based on logical conditions asserted using model input, output, and local variables. Model outputs (human actions) will also change to produce actions when given actions' execution states are Executing. See Bolton et al. (2011) and Bolton (2015) for more details.

2.2.1. Miscommunication generation

Bass et al. (2011) used EOFMC to model and evaluate communication protocols used to convey clearances between air traffic control and pilots. Bolton (2015) extended the EOFMC infrastructure to enable the automatic generation of miscommunications in EOFMC models. In miscommunication generation, any given communication action can execute normatively, have the source of the communication convey the wrong information, have one or more of the communication recipients receive the wrong information, or both. In all analyses, the analyst is able to control the maximum number of miscommunications that can occur (`cMax`). The net effect of this is that analysts can evaluate how robust a protocol is for all possible ways that `cMax` or fewer miscommunications can occur. Bolton (2015) used this to evaluate the robustness of different protocols air traffic control could use to communicate clearances to pilots. A limitation of all of these EOFMC studies is that they only considered specifications that would indicate whether or not the evaluated protocols always accomplished their goals, where perfect performance was required for the specification to prove true. For example, in Bolton (2015), formal verifications would only return a confirmation if, at the end of a given protocol, the entered clearance matched what was intended by the air traffic controller. While useful, such analyses do not give analysts nuanced insights into the performance of the protocol or the criticality of the failure.

2.2.2. Erroneous behavior generation

EOFMC can generate erroneous human behavior related to how human operators perform the non-communication portions of his or her task (Bolton and Bass, 2013). Specifically, the formal semantics of EOFM can be extended to allow the human operator to make simulated failures of attention, where they don't properly evaluate the strategic knowledge in EOFMC models, and perform slips (Reason, 1990). This enables activities to be erroneously omitted, repeated, or committed. A maximum (`sMax`) is used to control the number of erroneous behaviors considered in a given evaluation. This generation technique has been implemented as a feature in the EOFMC to SAL translator in which optional transitions are used to allow communication actions to be performed with incorrect outcomes (see Bolton, 2015 for more details). However, despite the presence of this feature, no analyses have examined how erroneous behaviors can interact with miscommunications in formal analyses that use EOFMC.

3. Objectives

There is a real need for an approach that will allow analysts to evaluate the degree to which a human-human collaborative procedure succeeds, and to do so with and without miscommunication and human operator erroneous behavior. This paper describes an extension of the approach from Bolton et al. (2013) that addresses this need. Specifically, we introduce novel specification criteria capable of allow analysts to diagnostically evaluate the performance of a human-human collaborative procedure, where each specification asserts that the procedure must perform at a different level of success; that is, assert an outcome that falls along an ordinal continuum of desirable outcomes. By formally verifying the specifications, the analyst will be able to determine what level of performance can be guaranteed with a given collaborative procedure with a given maximum number of miscommunications and generated human operator slips. Because human-human collaborative procedures can vary drastically from one application to another, there is no clear way to develop generic diagnostic specifications for all procedures. Thus, we contextualize our work in terms of a specific application.

In the following sections, a NPP application is used to demonstrate how our method works. Firstly, the background of this application, a Steam Generator Tube Rupture (SGTR) scenario, is described. A procedure for diagnosing an SGTR with two operators and a human-human communication protocol are then introduced. We next use EOFMC to model the SGTR diagnosis procedure and translate it into SAL. Different versions of the SAL file are created, each allowing for different maximum numbers of miscommunications and attention errors. Then, we identify six performance levels associated with SGTR diagnosis procedure and formulate them as specifications that are formally verified with model checking. We present these results along with an interpretation of their meaning. Based on these results, we updated our procedure (and its associated models) in an attempt to achieve a higher level of performance. Results from this modification are reported. Finally, we discuss the results and outline areas of future research.

4. Application

To illustrate how this approach can be used with safety-critical human-human collaborative procedures, a series of instantiated EOFMCs were created for an SGTR diagnose situation in main control room of an NPP.

In pressurized water reactors (PWRs), the steam generator tubing system is undergoing a variety of degradation processes which can lead to tube cracking, wall thinning, and potential leakage or rupture. This can cause an SGTR accident, where a leak in the reactor coolant system (RCS) results in coolant flowing into the steam generator (SG). If the safety systems are unavailable, or operators take incorrect or late actions, the pressure will increase rapidly and water or vapor with radioactive substances may be released into the environment. Even more seriously, the loss of reactor coolant may cause core damage. Once core damage occurs, and if the containment is bypassed, serious radioactivity release will happen (MacDonald et al., 1996). While very dangerous, SGTR accidents happen frequently (MacDonald et al., 1996). Thus operators must be prepared to handle them appropriately.

In this study, we are considering a 900MWe pressurized water reactor NPP where an alarm indicates that safety injection has lasted over 5 min. Two operators (operator 1 and operator 2) in main control need to collaborate to diagnose whether a given alarm signifies an SGTR accident. For safety purposes, human operators are expected to strictly follow the SGTR diagnosis procedure and associated human-human communication protocol. When an alarm sounds, indicating safety injection has lasted over 5 min, operators need to collaboratively diagnose the situation using the procedure in Fig. 1 (Dong, 2010).

As shown in Fig. 1, at step T0, the operators must observe the CVI, APG, and VVP radioactivity and judge whether they are more than 100 times their normal values. If not, the operators should conclude that it is not an SGTR accident and proceed to other diagnostic procedures (not discussed here) (Guangdong Nuclear Power Training Center, 2005). If true, the operators should proceed to step T1.

At T1, the operators should observe the liquid level and feedwater flow rates of all three SG and judge whether (a) the liquid level difference between any two SG is big, namely more than 10%, (b) the feedwater flow difference between any two SG is big, more than 0.2 E5 kg/h, and (c) one or more of the CVI, APG, and/or VVP radioactivity parameters are higher than normal. If (a), (b) and (c) are true, the operators should conclude that an SGTR accident has occurred and that the emergency operation procedure for an SGTR accident (a T2 procedure) should be performed. If not, the operators should conclude that something other than an SGTR accident has occurred and perform other diagnostic procedures (see Dong, 2010).

During the collaborative diagnostic process, two operators have to obey a communication protocol for confirming the iterative conclusions (reached through the diagnosis of the liquid levels, feedwater flow rates, and radioactivity levels) and final conclusion (whether or not to perform at T2 procedure) that are reached. In this protocol, operator 1 (Op1) takes the lead and is responsible for confirming conclusions with operator 2 (Op2). It proceeds as follows:

1. Op1 comes to a conclusion about the system.
2. Op1 communicates his[1] conclusion to Op2.
3. Op2 checks the system to see if he agrees with Op1's conclusion.
4. Op2 states whether he agrees or disagrees with Op1.
5. If Op1 hears a confirmation (“agree”), then he proceeds to a different diagnostic activity. If not, Op1 must re-evaluate his original conclusion.

5. Levels of performance

To assess the degree of success of this procedure for different maximum numbers of miscommunications and attention errors, we needed to identify different outcomes indicative of different levels of performance. To accomplish this, we observed that the goal of the procedure was to ensure that the operators achieved an accurate consensus about the system and what to do in response to the alarm. Within the model, this was indicated by the final and intermediate conclusions reached between the two operators. Thus, we identified the different ways that agreement could manifest after the performance of the procedure based on the final conclusions reached by each and, if they were correct, if the intermediary conclusions were consistent. We considered the safety implications of each of these outcomes and ordered them based on their desirability going from *A* (most desirable) to *F* (least desirable) (Table 2).

In the most desirable outcome (*A*), both Op1 and Op2 reach the correct final conclusion and the same intermediate conclusions. This ensures that they not only both agree on what to do, but they share the same understanding of the system state. In the second most desirable outcome (*B*), they both reach the same final conclusion, but have different intermediate conclusions. This is a slightly less desirable outcome than *A* because the difference in intermediary conclusions represents a disagreement in the

[1] Note that in the Chinese NPP used as the basis for this work, all operators are male.

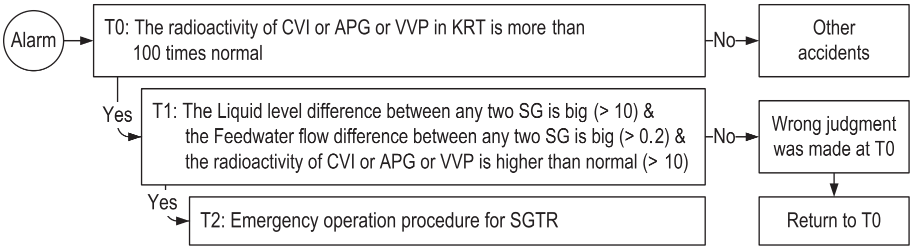
Alarm
T0: The radioactivity of CVI or APG or VVP in KRT is more than 100 times normal
No
Other accidents
Yes
T1: The Liquid level difference between any two SG is big (> 10) & the Feedwater flow difference between any two SG is big (> 0.2) & the radioactivity of CVI or APG or VVP is higher than normal (> 10)
No
Wrong judgment was made at T0
Yes
T2: Emergency operation procedure for SGTR
Return to T0

Fig. 1. SGTR diagnosis procedure.

Table 2
Diagnosis outcomes.

Outcome	Description
A	Op1 and Op2 reach the correct final conclusion and the same correct intermediary conclusions
B	Op1 and Op2 reach the correct final conclusion but differ on the intermediary conclusions, one of which may be wrong
C	Op1 has the correct final conclusion and Op2 does not
D	Op2 has the correct final conclusion and Op1 does not
E	Op1 and Op2 have different wrong final conclusions
F	Op1 and Op2 have the same wrong final conclusion

situational understanding between the operators that could potentially lead to confusion in later processes. Any situation where wrong final conclusions are reached (*C*–*F*) is undesirable. However, it is more desirable for Op1 to reach the correct final conclusion (*C*) since he is in charge of leading the response. This is slightly better than outcome *D*, where Op1 has reached the wrong final conclusion but Op2 the right one. This is still more desirable than latter outcomes because Op2 having the right final conclusion will increase the chances that the discrepancy will be noticed and that corrective action will be taken. In situations where both Op1 and Op2 reach the wrong final conclusions (*E* and *F*), it is more desirable for Op1 and Op2 to reach different conclusions as this could allow them to potentially discover their disagreement as activities proceed. Finally, a situation where Op1 and Op2 both reach the same wrong final conclusion is clearly the worst outcome, because they are more likely to proceed based on their wrong conclusion without noticing any disagreement. Note that these levels are consistent with the outcomes Jones and Endsley (2002) identified for team situation awareness.

6. Analysis of collaborative procedure 1

6.1. Modeling

The SGTR diagnosis procedure and communication protocol was instantiated as an EOFMC (visualized in Figs. 2–5). This model has two human operators: Op1 and Op2. The model had one task for Op1, five for Op2, and one shared task between them. The human operators had the same input variables representing the information from the environment they had access to (see Table 3). Generally, Op1 is responsible for working through the SGTR diagnosis procedure (Figs. 2 and 3). In this, when an alarm sounds, Op1 attempts to diagnose the procedure by first dismissing previous conclusions he may have made about the system (aResetConclusions). Then, he must determine if radioactivity is exceedingly high (aOp1CheckT0; Fig. 3a). If it is not, he concludes that something else is wrong with the system. If it is, he must check (under aOp1CheckT1; Fig. 2) the liquid levels (aOp1LiquidLevelsAreDifferent; Fig. 3b), the feedwater flow rates (aOp1FeedWaterFlowLevelsAreDifferent; Fig. 3c), and the radioactivity (aOp1RadioactivtyTooHigh; Fig. 3d) in any order. If all of these are consistent with an SGTR accident, he should conclude (aOp1FormConclusion; Fig. 2) that the T2 procedure needs to be performed. However, if at any point one of the checks fails, he should conclude that another procedure will need to be performed.

During this process, whenever Op1 reaches an intermediate or final conclusion (when lOp1UncheckedConclusion is set to a value not equal to NoConclusion; Figs. 2 and 3), he remembers what he has learned by setting the appropriate local variable to true or false and assigning lOp1UncheckedConclusion the new conclusion. He must then confirm that conclusion with Op2. Note that all of the tasks in Fig. 3 as well as aOp1Conclusion (Fig. 2) can only complete if Op1 has heard that Op2 agrees with him (the activities must repeat if Op2 disagrees) and lOp1UncheckedConclusion has been reset to NoConclusion. The process by which lOp1UncheckedConclusion is reset is handled by the task for the collaborative procedure (discussed in the next paragraph). Once Op1 has reached a final conclusion (aOp1FormConclusion; Fig. 2) and used the collaborative procedure to check the conclusion with Op2 (aOp1Conclusion), one of two things can happen. If Op2 agreed with Op1, the activity completes. Otherwise, the entire task (Fig. 2) repeats (starts over from the beginning).

This collaborative procedure is used by the two operators to determine if they reach the same conclusions. It executes in parallel to the tasks of Op1 and Op2. In the EOFMC, it is represented as a shared task (Fig. 4). In this, when Op1 has an unchecked conclusion (lOp1UncheckedConclusion ≠ NoConclusion), he must communicate that conclusion to Op2. Once Op2 has heard the conclusion reached by Op1, the task waits for Op2 to determine if he agrees with Op1 by keeping task execution from proceeding until the precondition on aCommunicateAgreeOrDisagree (Fig. 4) is satisfied (the tasks Op2 uses to accomplish this are discussed next). Once this occurs, if Op2 agrees with Op1's conclusion he will communicate back an “Agree”, otherwise he will communicate a “Disagree”. Note that at the beginning and end of the collaborative procedure, variables are reset to ensure proper coordination between the different tasks (Figs. 2–4) in the model.

Once Op2 has heard a conclusion from Op1, he is responsible for determining whether he agrees or disagrees with it using one of the five tasks in Fig. 5. The conclusion heard from Op1 (lOp2ConclusionFromOp1) determines which task executes using the preconditions on the tasks topmost activities. Op2 has separate tasks for confirming or contradicting each of the conclusions (final or otherwise) that Op1 has reached using the same criteria as Op1.

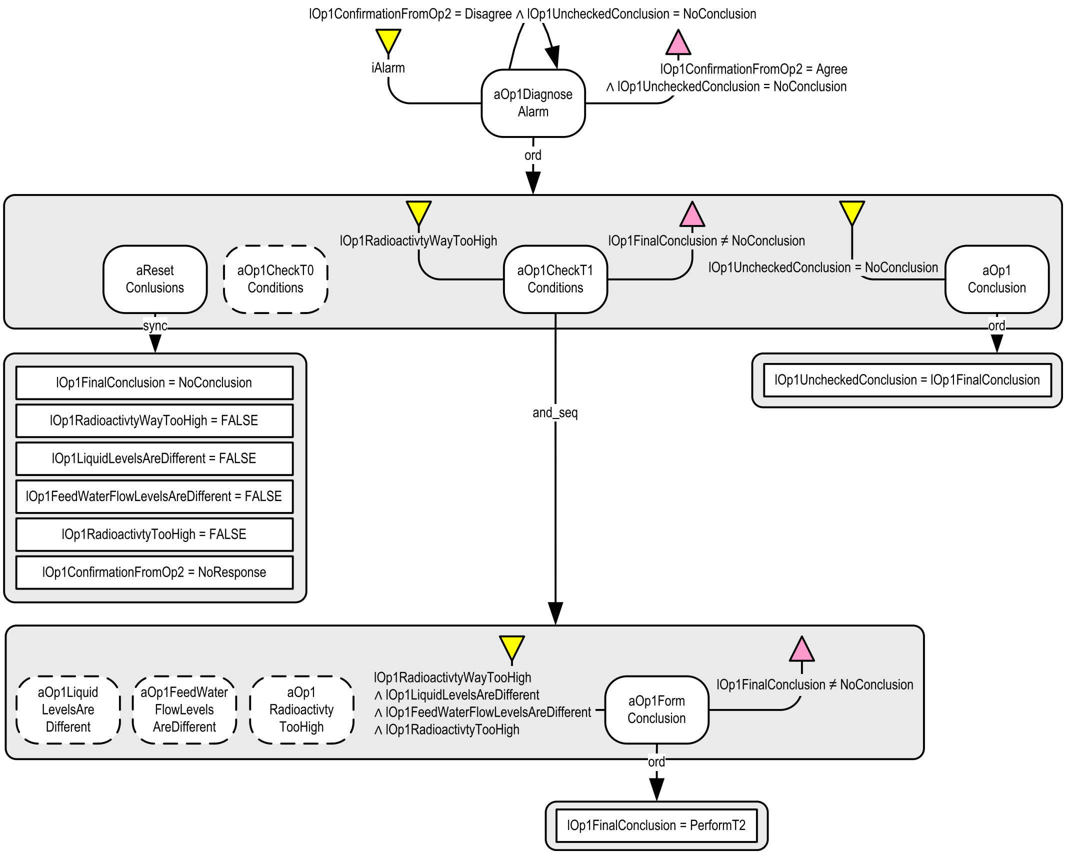
lOp1ConfirmationFromOp2 = Disagree ∧ lOp1UncheckedConclusion = NoConclusion
iAlarm
aOp1Diagnose Alarm
lOp1ConfirmationFromOp2 = Agree ∧ lOp1UncheckedConclusion = NoConclusion
ord
lOp1RadioactivtyWayTooHigh
lOp1FinalConclusion ≠ NoConclusion
aReset Conlusions
aOp1CheckT0 Conditions
aOp1CheckT1 Conditions
lOp1UncheckedConclusion = NoConclusion
aOp1 Conclusion
sync
ord
lOp1FinalConclusion = NoConclusion
lOp1RadioactivtyWayTooHigh = FALSE
lOp1LiquidLevelsAreDifferent = FALSE
lOp1FeedWaterFlowLevelsAreDifferent = FALSE
lOp1RadioactivtyTooHigh = FALSE
lOp1ConfirmationFromOp2 = NoResponse
lOp1UncheckedConclusion = lOp1FinalConclusion
and_seq
aOp1Liquid LevelsAre Different
aOp1FeedWater FlowLevels AreDifferent
aOp1 Radioactivty TooHigh
lOp1RadioactivtyWayTooHigh ∧ lOp1LiquidLevelsAreDifferent ∧ lOp1FeedWaterFlowLevelsAreDifferent ∧ lOp1RadioactivtyTooHigh
aOp1Form Conclusion
lOp1FinalConclusion ≠ NoConclusion
ord
lOp1FinalConclusion = PerformT2

Fig. 2. Visualization of the instantiated EOFMC collaborative procedure 1 (using the EOFMC visual notation; see Bolton and Bass, 2010) representing the task performed by Op1. Activities are rounded rectangles. Actions, in this case local actions (variable assignments) representing human operator mental operations, are rectangles. Conditions are connected to the activity they modify: a precondition is represented by a yellow, downward pointing triangle connected to the left side of the activity; a completion condition is presented as a magenta, upward pointing triangle connected to the right of the activity; and a repeat condition is conveyed as a recursive arrow attached to the top of the activity. A decomposition is presented as an arrow, labeled with the decomposition operator (see Section 2.2), extending below an activity that points to a large rounded rectangle containing the decomposed activities or actions. Activities with dotted lines are defined elsewhere (see Fig. 3). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The complete, instantiated EOFMC task model was converted into the input language of SAL using the EOFMC java-based translator (Bolton et al., 2011; Bolton, 2015). The SAL version of the model was then modified to create different versions for analyses. Specifically, in each version of the model, the maximum number of communication errors (`cMax`) was set from 0 to 4 in increments of 1 and the maximum number of attention errors/slips (`sMax`) was set from 0 to 2 in increments of 1.

6.2. Specification

To assert different levels of performance, we developed specification properties asserting that at least a given performance outcome (Table 2) was achieved by comparing the operator local variables representing the different conclusions the operators came to over the course of the procedure (see Table 3). This resulted in six specification properties (Table 4). Each was designed so that, if it verified true, its corresponding level of performance was guaranteed.

6.3. Formal verification and results

Formal verifications were performed using SAL's Symbolic Model Checker (SAL-SMC). For each system model with different values of `cMax` and `sMax`, all six of the specifications (Table 4) were checked starting with I and working towards VI. At any point in this process, if a specification verified to true, verification on that model was stopped. The specification that verified to true indicated the performance level guaranteed by that model. If at any point a model performed at the worst level of performance (VI), no additional analyses were conducted with larger values of `cMax` and `sMax` because such increments would only produce models that performed at the lowest level.

Analysis results, along with the number visited states and verification times for the reported performance level, are shown in Table 5. These results show that collaborative procedure 1 achieves different performance levels in different conditions. For no miscommunications and no slips of attention, the model performed at level I. Thus, when there are no problems, the protocol (as modeled) will always achieve the top level of performance. When there are no attention errors, for all other values of `cMax`, the model performed at level III (guaranteeing at least an outcome of *C*; Table 2). Given that the models consistently performed at level III as the maximum number of miscommunications increased beyond 0, it is very likely that this perform level would continue to be observed if `cMax` was further increased. This is a positive result for the procedure because it indicates that the lead operator will always reach the correct conclusion. Since the lead operator is responsible for executing interventions based on the conclusion he reaches, this means that the procedure will likely be successful even with miscommunications. However, performance degraded when slips were included in the analysis.

Please cite this article in press as: Pan, D., Bolton, M.L., Properties for formally assessing the performance level of human-human collaborative procedures with miscommunications and erroneous human behavior, International Journal of Industrial Ergonomics (2016), http://dx.doi.org/10.1016/j.ergon.2016.04.001

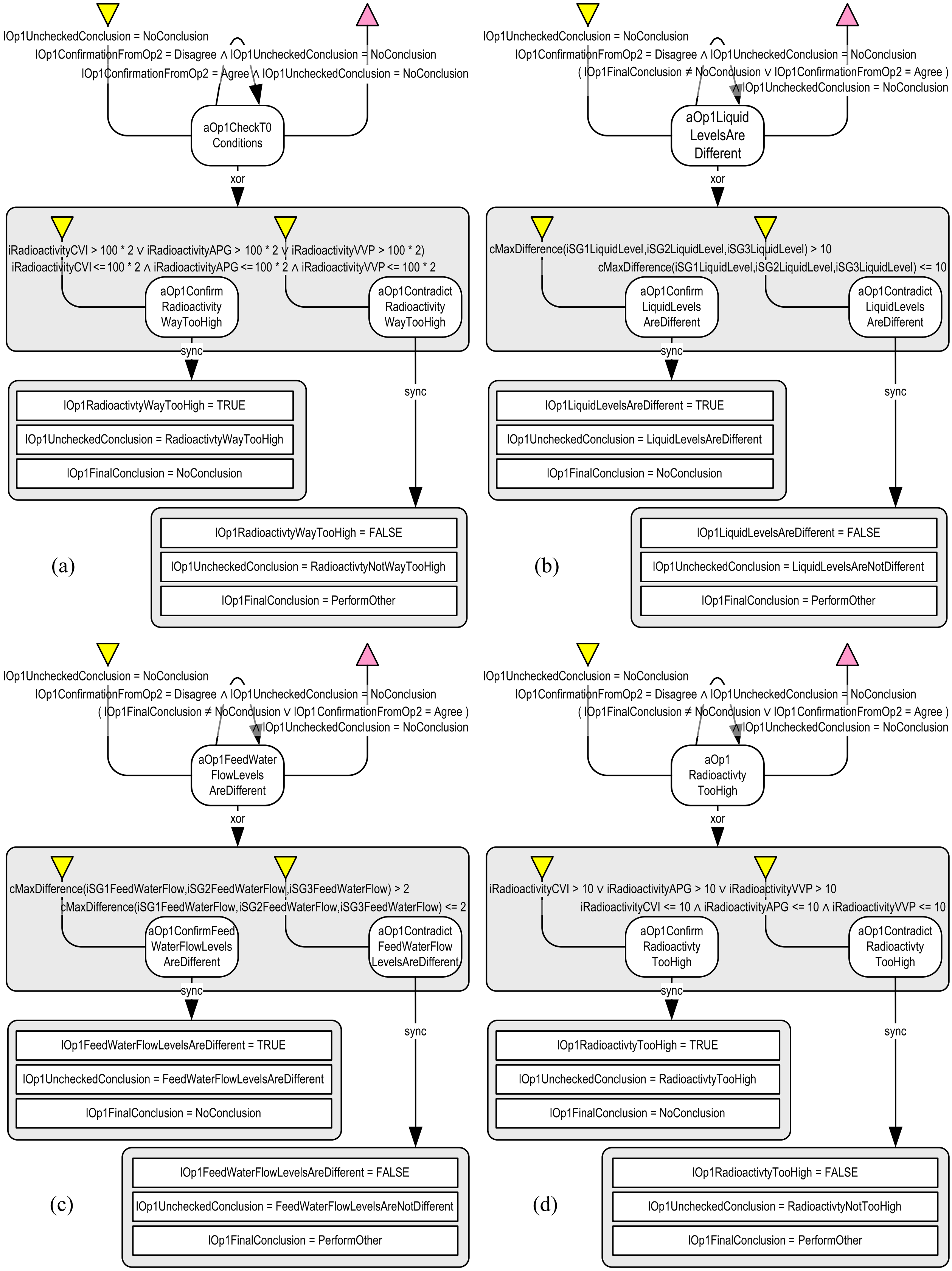
lOp1UncheckedConclusion = NoConclusion
lOp1ConfirmationFromOp2 = Disagree ∧ lOp1UncheckedConclusion = NoConclusion
lOp1ConfirmationFromOp2 = Agree ∧ lOp1UncheckedConclusion = NoConclusion
aOp1CheckT0 Conditions
xor
iRadioactivityCVI > 100 * 2 ∨ iRadioactivityAPG > 100 * 2 ∨ iRadioactivityVVP > 100 * 2)
iRadioactivityCVI <= 100 * 2 ∧ iRadioactivityAPG <= 100 * 2 ∧ iRadioactivityVVP <= 100 * 2
aOp1Confirm Radioactivity WayTooHigh
aOp1Contradict Radioactivity WayTooHigh
sync
sync
lOp1RadioactvtyWayTooHigh = TRUE
lOp1UncheckedConclusion = RadioactvtyWayTooHigh
lOp1FinalConclusion = NoConclusion
lOp1RadioactvtyWayTooHigh = FALSE
lOp1UncheckedConclusion = RadioactvtyNotWayTooHigh
lOp1FinalConclusion = PerformOther
(a)
lOp1UncheckedConclusion = NoConclusion
lOp1ConfirmationFromOp2 = Disagree ∧ lOp1UncheckedConclusion = NoConclusion
(lOp1FinalConclusion ≠ NoConclusion ∨ lOp1ConfirmationFromOp2 = Agree)
∧ lOp1UncheckedConclusion = NoConclusion
aOp1Liquid LevelsAre Different
xor
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) > 10
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) <= 10
aOp1Confirm LiquidLevels AreDifferent
aOp1Contradict LiquidLevels AreDifferent
sync
sync
lOp1LiquidLevelsAreDifferent = TRUE
lOp1UncheckedConclusion = LiquidLevelsAreDifferent
lOp1FinalConclusion = NoConclusion
lOp1LiquidLevelsAreDifferent = FALSE
lOp1UncheckedConclusion = LiquidLevelsAreNotDifferent
lOp1FinalConclusion = PerformOther
(b)
lOp1UncheckedConclusion = NoConclusion
lOp1ConfirmationFromOp2 = Disagree ∧ lOp1UncheckedConclusion = NoConclusion
(lOp1FinalConclusion ≠ NoConclusion ∨ lOp1ConfirmationFromOp2 = Agree)
∧ lOp1UncheckedConclusion = NoConclusion
aOp1FeedWater FlowLevels AreDifferent
xor
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) > 2
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) <= 2
aOp1ConfirmFeed WaterFlowLevels AreDifferent
aOp1Contradict FeedWaterFlow LevelsAreDifferent
sync
sync
lOp1FeedWaterFlowLevelsAreDifferent = TRUE
lOp1UncheckedConclusion = FeedWaterFlowLevelsAreDifferent
lOp1FinalConclusion = NoConclusion
lOp1FeedWaterFlowLevelsAreDifferent = FALSE
lOp1UncheckedConclusion = FeedWaterFlowLevelsAreNotDifferent
lOp1FinalConclusion = PerformOther
(c)
lOp1UncheckedConclusion = NoConclusion
lOp1ConfirmationFromOp2 = Disagree ∧ lOp1UncheckedConclusion = NoConclusion
(lOp1FinalConclusion ≠ NoConclusion ∨ lOp1ConfirmationFromOp2 = Agree)
∧ lOp1UncheckedConclusion = NoConclusion
aOp1 Radioactivty TooHigh
xor
iRadioactivityCVI > 10 ∨ iRadioactivityAPG > 10 ∨ iRadioactivityVVP > 10
iRadioactivityCVI <= 10 ∧ iRadioactivityAPG <= 10 ∧ iRadioactivityVVP <= 10
aOp1Confirm Radioactivty TooHigh
aOp1Contradict Radioactivty TooHigh
sync
sync
lOp1RadioactvtyTooHigh = TRUE
lOp1UncheckedConclusion = RadioactvtyTooHigh
lOp1FinalConclusion = NoConclusion
lOp1RadioactvtyTooHigh = FALSE
lOp1UncheckedConclusion = RadioactvtyNotTooHigh
lOp1FinalConclusion = PerformOther
(d)

Fig. 3. EOFMC visualizations of the EOFMC activities from Fig. 2. Each activity is concerned with checking different criteria from the SGTR procedure (see Fig. 1): (a) checking the T0 condition that radioactivity is higher than 100 times normal, (b) checking if liquid levels are different, (c) checking feedwater flow levels, and (d) checking if radioactivity is higher than normal.

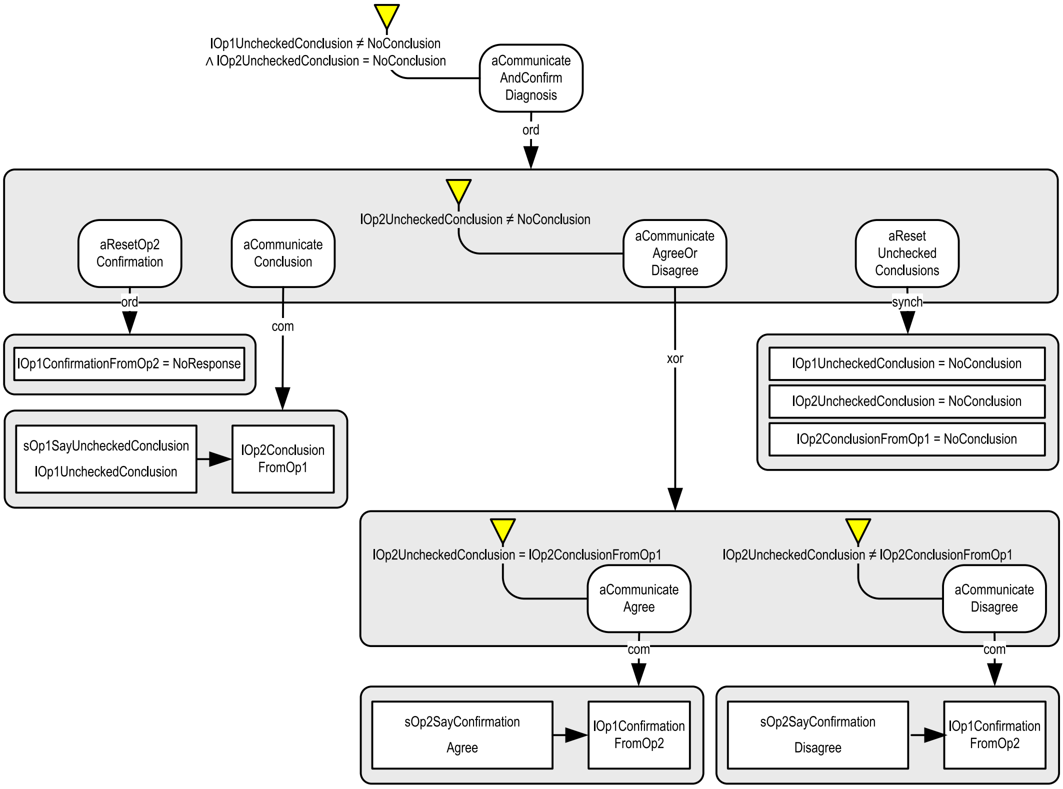
lOp1UncheckedConclusion ≠ NoConclusion
∧ lOp2UncheckedConclusion = NoConclusion
aCommunicate AndConfirm Diagnosis
ord
lOp2UncheckedConclusion ≠ NoConclusion
aResetOp2 Confirmation
aCommunicate Conclusion
aCommunicate AgreeOr Disagree
aReset Unchecked Conclusions
ord
com
xor
synch
lOp1ConfirmationFromOp2 = NoResponse
sOp1SayUncheckedConclusion
lOp1UncheckedConclusion
lOp2Conclusion FromOp1
lOp1UncheckedConclusion = NoConclusion
lOp2UncheckedConclusion = NoConclusion
lOp2ConclusionFromOp1 = NoConclusion
lOp2UncheckedConclusion = lOp2ConclusionFromOp1
lOp2UncheckedConclusion ≠ lOp2ConclusionFromOp1
aCommunicate Agree
aCommunicate Disagree
com
com
sOp2SayConfirmation
Agree
lOp1Confirmation FromOp2
sOp2SayConfirmation
Disagree
lOp1Confirmation FromOp2

Fig. 4. Visualization of the EOFMC representing the shared task between Op1 and Op2 for communicating conclusions, confirmations, or disagreements. Note that the rectangles in a communication decomposition (one labeled with com) describe how communication actions occur. For example, under aCommunicateConclusion, the communication action sOp1SayUncheckedConclusion is performed by Op1. In this the value of lOp1UncheckedConclusion is communicated to Op2 who remembers it (stores it in the lOp2-ConclusionFromOp1 variable).

When one attention error occurs and there are no miscommunications, the model performs at level III. For larger numbers of miscommunications and/or slips, model performance decreased to level VI, the worst possible level of performance. This indicates that Op1 and Op2 have the same wrong final conclusion and thus would likely not properly respond to the alarm. This result indicates that collaborative procedure 1 is robust to miscommunication, but not to more than one slip or combination of slip and miscommunication.

To understand why only performance levels III and VI were achieved above, counterexamples from the model checking analyses were evaluated using the EOFMC counterexample visualizer (Bolton and Bass, 2010). For example, for the condition where cMax = 1 and sMax = 0, we analyzed the counterexample that was returned when checking for level II performance. This revealed an execution sequence where everything proceeded as it should until Op1 attempted to tell Op2 that he concluded that a different procedure from T2 (PerformOther) should be performed. However, a miscommunication occurred and Op2 heard Op1 say that feedwater flow levels were not different. Since this was true in the scenario in the counterexample, Op2 gave Op1 a confirmation. Thus Op1 finished the diagnostic procedure having come to the right conclusion, but without Op2 reaching that conclusion. Counterexamples showing level II performance failures for sMax = 0 and higher numbers of cMax were similar.

For the level III performance observed when cMax = 0 and sMax = 1, the counterexample produced when verifying for level II performance was evaluated. This showed that when it came time for Op1 to communicate the correct final conclusion to Op2, he omitted this activity (a slip) and ended the procedure. Thus, Op1 finished with the correct conclusion (in this case to not perform T2) and Op2 did not.

In the case when cMax = 1 and sMax = 1, the counterexample from the level V verification was analyzed. This should that the failure occurred as follows. First, Op1 mistakenly concluded that radioactivity was not way too high (a slip). Then, when Op1 communicated this to Op2, a miscommunication occurred and Op2 heard that feedwater flow levels were different. Since this was true, Op2 communicated a confirmation. This caused Op1 to incorrectly conclude not to perform T2. He then communicated his conclusion correctly to Op2. However, because Op2 had not checked all of the information necessary for concluding that T2 needed to be performed, he agreed with Op1's final conclusion.

7. Analysis of collaborative procedure 2

The information contained in the counterexamples illustrated a number of potential limitations of the original collaborative procedure. One such limitation comes from the fact that Op2 uses a confirmation or disagreement statement to indicate if they have reached the same intermediate or final conclusion as Op1. In other areas, such as air transportation, readbacks have shown themselves to be more robust to miscommunications (Farris and Barshi, 2013). Thus, in this section, we modify the original collaborative procedure to incorporate readbacks so that we can assess how this modification might improve the performance of the collaborative procedure.

This new procedure proceeds as follows:

1. Op1 comes to a conclusion about the system based on SGTR diagnosis procedure.
2. Op1 communicates his conclusion to Op2.

Please cite this article in press as: Pan, D., Bolton, M.L., Properties for formally assessing the performance level of human-human collaborative procedures with miscommunications and erroneous human behavior, International Journal of Industrial Ergonomics (2016), http://dx.doi.org/10.1016/j.ergon.2016.04.001

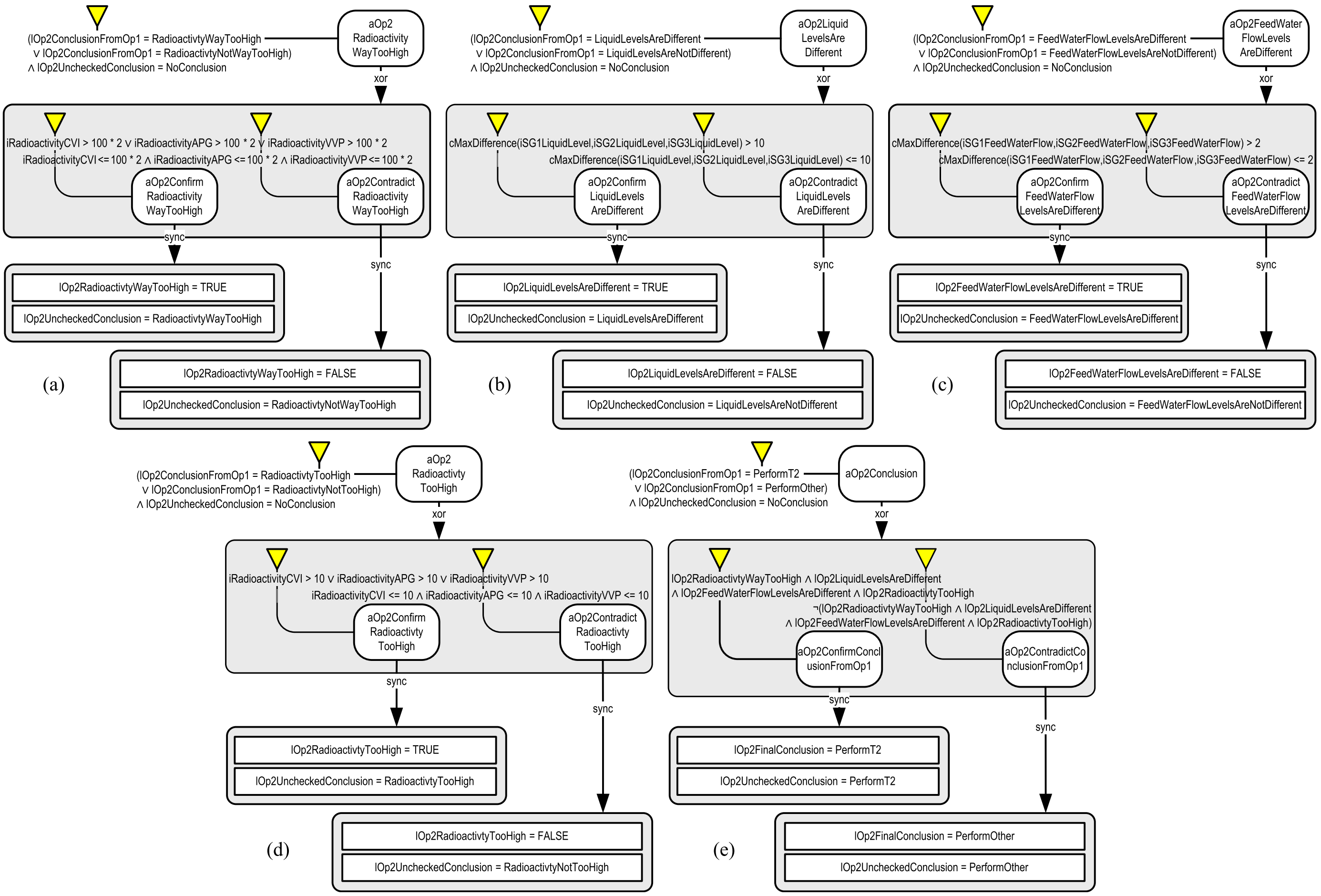
(lOp2ConclusionFromOp1 = RadioactivtyWayTooHigh
∨ lOp2ConclusionFromOp1 = RadioactivtyNotWayTooHigh)
∧ lOp2UncheckedConclusion = NoConclusion
aOp2 Radioactivity WayTooHigh
xor
iRadioactivityCVI > 100 * 2 ∨ iRadioactivityAPG > 100 * 2 ∨ iRadioactivityVVP > 100 * 2
iRadioactivityCVI <= 100 * 2 ∧ iRadioactivityAPG <= 100 * 2 ∧ iRadioactivityVVP <= 100 * 2
aOp2Confirm Radioactivity WayTooHigh
aOp2Contradict Radioactivity WayTooHigh
sync
sync
lOp2RadioactivtyWayTooHigh = TRUE
lOp2UncheckedConclusion = RadioactivtyWayTooHigh
lOp2RadioactivtyWayTooHigh = FALSE
lOp2UncheckedConclusion = RadioactivtyNotWayTooHigh
(a)
(lOp2ConclusionFromOp1 = LiquidLevelsAreDifferent
∨ lOp2ConclusionFromOp1 = LiquidLevelsAreNotDifferent)
∧ lOp2UncheckedConclusion = NoConclusion
aOp2Liquid LevelsAre Different
xor
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) > 10
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) <= 10
aOp2Confirm LiquidLevels AreDifferent
aOp2Contradict LiquidLevels AreDifferent
sync
sync
lOp2LiquidLevelsAreDifferent = TRUE
lOp2UncheckedConclusion = LiquidLevelsAreDifferent
lOp2LiquidLevelsAreDifferent = FALSE
lOp2UncheckedConclusion = LiquidLevelsAreNotDifferent
(b)
(lOp2ConclusionFromOp1 = FeedWaterFlowLevelsAreDifferent
∨ lOp2ConclusionFromOp1 = FeedWaterFlowLevelsAreNotDifferent)
∧ lOp2UncheckedConclusion = NoConclusion
aOp2FeedWater FlowLevels AreDifferent
xor
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) > 2
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) <= 2
aOp2Confirm FeedWaterFlow LevelsAreDifferent
aOp2Contradict FeedWaterFlow LevelsAreDifferent
sync
sync
lOp2FeedWaterFlowLevelsAreDifferent = TRUE
lOp2UncheckedConclusion = FeedWaterFlowLevelsAreDifferent
lOp2FeedWaterFlowLevelsAreDifferent = FALSE
lOp2UncheckedConclusion = FeedWaterFlowLevelsAreNotDifferent
(c)
(lOp2ConclusionFromOp1 = RadioactivtyTooHigh
∨ lOp2ConclusionFromOp1 = RadioactivtyNotTooHigh)
∧ lOp2UncheckedConclusion = NoConclusion
aOp2 Radioactivty TooHigh
xor
iRadioactivityCVI > 10 ∨ iRadioactivityAPG > 10 ∨ iRadioactivityVVP > 10
iRadioactivityCVI <= 10 ∧ iRadioactivityAPG <= 10 ∧ iRadioactivityVVP <= 10
aOp2Confirm Radioactivty TooHigh
aOp2Contradict Radioactivty TooHigh
sync
sync
lOp2RadioactivtyTooHigh = TRUE
lOp2UncheckedConclusion = RadioactivtyTooHigh
lOp2RadioactivtyTooHigh = FALSE
lOp2UncheckedConclusion = RadioactivtyNotTooHigh
(d)
(lOp2ConclusionFromOp1 = PerformT2
∨ lOp2ConclusionFromOp1 = PerformOther)
∧ lOp2UncheckedConclusion = NoConclusion
aOp2Conclusion
xor
lOp2RadioactivtyWayTooHigh ∧ lOp2LiquidLevelsAreDifferent
∧ lOp2FeedWaterFlowLevelsAreDifferent ∧ lOp2RadioactivtyTooHigh
¬(lOp2RadioactivtyWayTooHigh ∧ lOp2LiquidLevelsAreDifferent
∧ lOp2FeedWaterFlowLevelsAreDifferent ∧ lOp2RadioactivtyTooHigh)
aOp2ConfirmConcl usionFromOp1
aOp2ContradictCo nclusionFromOp1
sync
sync
lOp2FinalConclusion = PerformT2
lOp2UncheckedConclusion = PerformT2
lOp2FinalConclusion = PerformOther
lOp2UncheckedConclusion = PerformOther
(e)

Fig. 5. Visualization of the EOFMC tasks Op2 uses to determine whether he agrees with Op1. There is a separate task for each condition that Op2 would need to check: (a) radioactivity, (b) liquid levels, (c) feedwater flow levels, (d) if radioactivity is too high, and (e) the final conclusion.

Table 3
Variables used in the application EOFMC.

Variable class	Variable name	Description
Shared input variables	`iAlarm`	Boolean variable that is true if an alarm is sounding and false otherwise.
	`iRadioactivityCVI`	Variables indicating the levels of the radioactivity at three different locations in the power plant.
	`iRadioactivityAPG`	
	`iRadioactivityVVP`	
	`iSG1LiquidLevel`	Variables representing the liquid levels of the three SG.
	`iSG2LiquidLevel`	
	`iSG3LiquidLevel`	
	`iSG1FeedWaterFlow`	Variables representing the feedwater flow levels at the three SG.
	`iSG2FeedWaterFlow`	
	`iSG3FeedWaterFlow`	
Operator local variables	`lOp1RadioactivtyWayTooHigh`	Boolean variables representing Op1's and Op2's respective determinations about whether radioactivity is too high when checking T0. Initially set to false.
	`lOp2RadioactivtyWayTooHigh`	
	`lOp1LiquidLevelsAreDifferent`	Boolean variables representing Op1's and Op2's respective determinations about whether liquid levels are different. Initially set to false.
	`lOp2LiquidLevelsAreDifferent`	
	`lOp1FeedWaterFlowLevelsAreDifferent`	Boolean variables representing Op1's and Op2's respective determinations about whether feedwater flow levels are different. Initially set to false.
	`lOp2FeedWaterFlowLevelsAreDifferent`	
	`lOp1RadioactivtyTooHigh`	Boolean variables representing Op1's and Op2's respective determinations about whether radioactivity is too high when checking T1. Initially set to false.
	`lOp2RadioactivtyTooHigh`	
	`lOp1FinalConclusion`	The final conclusions reached by Op1 and Op2 respectively. Initially set to NoConlcusion
	`lOp2FinalConclusion`	
Operator local variables used for communication and coordination between models	`lOp1UncheckedConclusion`	Any intermediate or final conclusion reached by Op1 that has not been checked with Op2.
	`lOp2ConclusionFromOp1`	The conclusion Op2 hears when communicating with Op1.
	`lOp2UncheckedConclusion`	Any intermediate or final conclusion reached by Op2 when checking a conclusion communicated by Op1.
	`lOp1ConfirmationFromOp2`	Agreement or Disagreement heard by Op1 from Op2 about a conclusion Op1 communicated to Op2.

Table 4
Specifications of different performance levels.

Performance level	Specification	Guaranteed diagnosis outcomes
I	$G\,(A)$	A
II	$G\,(A \vee B)$	B
III	$G\,(A \vee B \vee C)$	C
IV	$G\,(A \vee B \vee C \vee D)$	D
V	$G\,(A \vee B \vee C \vee D \vee E)$	E
VI	$G\,(A \vee B \vee C \vee D \vee E \vee F)$	F

Note. A − F are diagnostic outcomes (Table 1) expressed logically using model variables from Table 3.

3. Op2 checks the system and comes to a conclusion about the system.
4. Op2 states his conclusion to Op1.
5. Op1 hears Op2's conclusion. If the conclusion Op1 hears is the same as Op1's original conclusion, then he proceeds to a different diagnostic activity. If not, Op1 must re-evaluate his original conclusion.

7.1. Modeling and specification

Collaborative procedure 2 was instantiated as a modification of collaborative procedure 1. These changes are shown in Figs. 6 and 7, representing the new Op1 procedure originally from Figs. 2 and 3 (note that the local variable lOp1HasNewUncheckedConclusion was added to Op1 to facilitate coordination between the tasks in the model). Fig. 8, representing the new shared task originally from Fig. 4. The instantiated EOFMC for Op2 was the same as in collaborative procedure 1 (Fig. 5). In this, when Op1 has an unchecked conclusion, he must communicate that conclusion to Op2. When Op2 hears an unchecked conclusion from Op1, Op2 should check the system parameters, come to his own unchecked conclusion and state it directly to Op1. If Op1 agrees with the conclusion (because it matches his original one), he will proceed to a different activity. If not, he will reevaluate his original conclusion and repeat the process. As with collaborative procedure 1, variables are reset at the beginning and end of the shared task.

This new instantiated EOFMC was translated into SAL (Bolton et al., 2011). As with the first procedure, the SAL version of the model was then modified to create different versions with values of cMax from 0 to 4 values of sMax from 0 to 2.

The same specifications (Table 4) were used in the verification of this new procedure as was done with procedure 1.

7.2. Formal verification and results

Formal verifications for procedure 2 were conducted using the same procedures used for procedure 1. Results are reported in Table 6. Comparing the results obtained for procedure 1: when there were no slips and one miscommunication, the performance level was improved from III to II. However, no other improvements were observed.

To understand how the performance level was improved to level II when cMax = 1 and sMax = 0, we analyzed the counterexample obtained from verifying for level I performance. In this counterexample, everything proceeded normatively until Op1 told Op2 that he concluded that a different procedure from T2 (PerformOther) should be performed. However, a miscommunication occurred and Op2 heard Op1 say that feedwater flow levels were different. Since this is true in the scenario, Op2 told Op1 that feedwater flow levels were different. Op1 heard Op2 say this correctly. However, because this did not confirm with his original conclusion, he rechecked his original conclusion and communicated it back to Op2. This time Op2 repeated Op1's final conclusion back to him. Thus, while Op1 and Op2 both reached the same final conclusion, Op1 never checked feedwater flow levels as part of his procedure. This resulted in Op1 and Op2 having slightly different situational understandings about the system.

When the number of miscommunications increased to 2 (cMax = 2 and sMax = 0), level III performance was observed. Thus we analyzed the corresponding counterexample showing level II performance failure. In this, everything proceeded normatively until Op1 told Op2 that he had concluded not to perform T2 (to PerformOther). However, a miscommunication occurred and Op2 heard Op1 say that liquid levels were not different. Then, Op2 repeated back this same, miscommunicated conclusion. However, a second miscommunication occurred, where Op1 heard that Op2 agreed with his conclusion to not perform T2. Thus, the procedure finished with Op1 reaching the correct conclusion and Op2 not reaching any conclusion.

Additional counterexample analyses are not presented here because of their similarity to the counterexamples discussed in this and the previous section.

8. Discussion and future work

The presented work constitutes a significant contribution in that it gives analysts the ability to better assess the robustness of human-human collaborative procedures using formal verification. Specifically, by allowing analyst to assess the level of performance guaranteed by a procedure for varying numbers of miscommunications and attentional slips, analysts can gain additional insights into how well it will perform.

The application described in this study is illustrative of the power of our approach. Specifically, if the presented procedures were formally evaluated in the traditional way, just at level I, it would be considered a failure for all maximum numbers of miscommunications and slips greater than 0. By verifying our novel specifications, it is now clear that, although it does not provide

Table 5
Formal verification Results for Collaborative procedure 1.

	sMax								
cMax	0			1			2		
	Level	# States	Time (s)	Level	# States	Time (s)	Level	# States	Time (s)
0	I	8,750,901,376	6.27	III	344,117,662,824	67.13	VI	5,625,942,614,224	655.89
1	III	406,460,756,960	33.74	VI	20,043,870,775,760	1143.24			
2	III	4,117,867,927,982	139.82						
3	III	20,553,148,929,006	700.18						
4	III	58,367,109,537,562	3367.32						

Note. # States indicates the number of states that were visited during a given verification run.

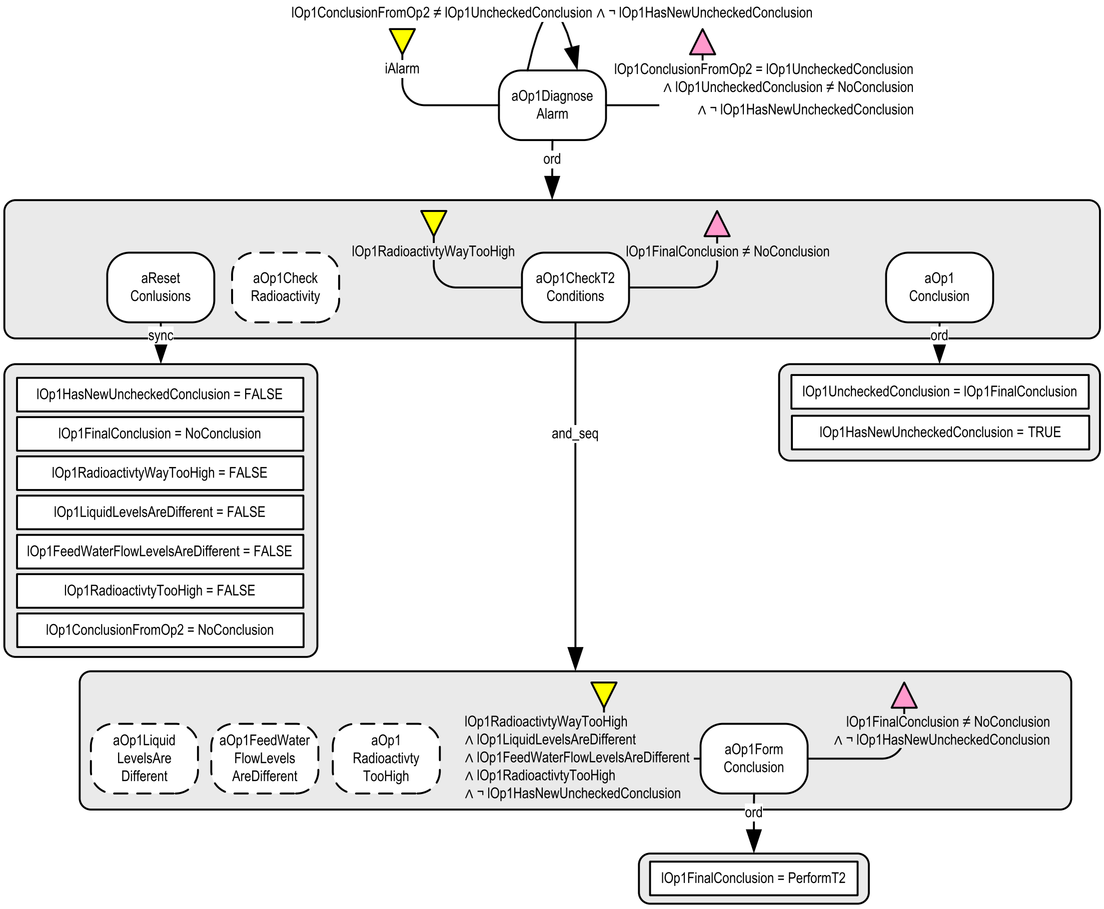
lOp1ConclusionFromOp2 ≠ lOp1UncheckedConclusion ∧ ¬ lOp1HasNewUncheckedConclusion
iAlarm
aOp1Diagnose Alarm
lOp1ConclusionFromOp2 = lOp1UncheckedConclusion
∧ lOp1UncheckedConclusion ≠ NoConclusion
∧ ¬ lOp1HasNewUncheckedConclusion
ord
lOp1RadioactivtyWayTooHigh
lOp1FinalConclusion ≠ NoConclusion
aReset Conlusions
aOp1Check Radioactivity
aOp1CheckT2 Conditions
aOp1 Conclusion
sync
ord
lOp1HasNewUncheckedConclusion = FALSE
lOp1FinalConclusion = NoConclusion
lOp1RadioactivtyWayTooHigh = FALSE
lOp1LiquidLevelsAreDifferent = FALSE
lOp1FeedWaterFlowLevelsAreDifferent = FALSE
lOp1RadioactivtyTooHigh = FALSE
lOp1ConclusionFromOp2 = NoConclusion
and_seq
lOp1UncheckedConclusion = lOp1FinalConclusion
lOp1HasNewUncheckedConclusion = TRUE
aOp1Liquid LevelsAre Different
aOp1FeedWater FlowLevels AreDifferent
aOp1 Radioactivty TooHigh
lOp1RadioactivtyWayTooHigh
∧ lOp1LiquidLevelsAreDifferent
∧ lOp1FeedWaterFlowLevelsAreDifferent
∧ lOp1RadioactivtyTooHigh
∧ ¬ lOp1HasNewUncheckedConclusion
aOp1Form Conclusion
lOp1FinalConclusion ≠ NoConclusion
∧ ¬ lOp1HasNewUncheckedConclusion
ord
lOp1FinalConclusion = PerformT2

Fig. 6. Visualization of the instantiated EOFMC collaborative procedure 2 representing the task performed by Op1. Note that activities with dotted lines are defined in Fig. 7.

perfect performance, the procedure does provide some guarantees that the correct conclusion will be reached. Thus, the presented work gives analysts who wish to formally evaluate human-human communication and coordination procedures formally deeper analysis capabilities.

The presented analyses also illustrate how the presented method and performance levels can be useful for comparing different collaborative procedures. For example, procedure 2 exhibited the same or better performance levels for all comparable combinations of `cMax` and `sMax` than procedure 1. Thus, procedure 2, with its use of readbacks instead of confirmations, will be more robust than procedure 1. This ability to illustrate the difference in the performance levels of similar collaborative procedures should be generalizable and applicable in other domains.

It is true that the use of maximums to control the number of miscommunications and slips is artificial. However, this should not be viewed as an impediment to the usefulness of the method. As we did in our analyses, analysts wishing to use our approach can iteratively increase the maximums until their results stabilize, scalability becomes a restriction, or they are satisfied with the reliability of the system.

The presented application is also illustrative of the power of formal methods. Other techniques such as testing, simulation, and stochastic analyses will always be incomplete because they will not exhaustively explore all of the conditions that can arise from the different interactions in the system. The erroneous behavior generation techniques used here are particularly useful in this respect because analyst can use formal method to determine how erroneous behavior can occur and/or interact in ways that were unanticipated. Though not the focus of the presented work, other published research has explored how verification results can be used to correct discovered problems (Bolton et al., 2012; Bolton and Bass, 2013; Bolton, 2015). Thus, the method clearly supports this capability.

However, as with any method, there are limitations of this approach. We discuss how these should be addressed in future work below.

8.1. Other analysis domains

Communication is important to system safety in a number of domains beyond nuclear power. For example, in the area of rail transit and highway, 75% of all roadway crashes are related to human-human interaction, and more than 92% of repair and maintenance issues are attributed to communication errors (Murphy, 2001). In medical accidents resulting in death in Australia, the rate of communication errors is two times higher than that of medical skill errors (Wilson et al., 1995). In the aviation industry, NASA researchers analyzed the causes of jet transport accidents and incidents between 1968 and 1976 (Cooper et al., 1980) and concluded that pilot error was more likely to reflect failures in team communication and coordination than deficiencies in technical proficiency. A report by the Aviation Safety Reporting System (ASRS) showed that the proportion of aviation mishaps due to communication error was above 70% (Connell, 1996). Further, communications between surgical teams ensure that surgical protocols are properly executed and coordinated. Handoff of care protocols in hospitals attempt to communicate all relevant patient health information between teams and between shift changes.

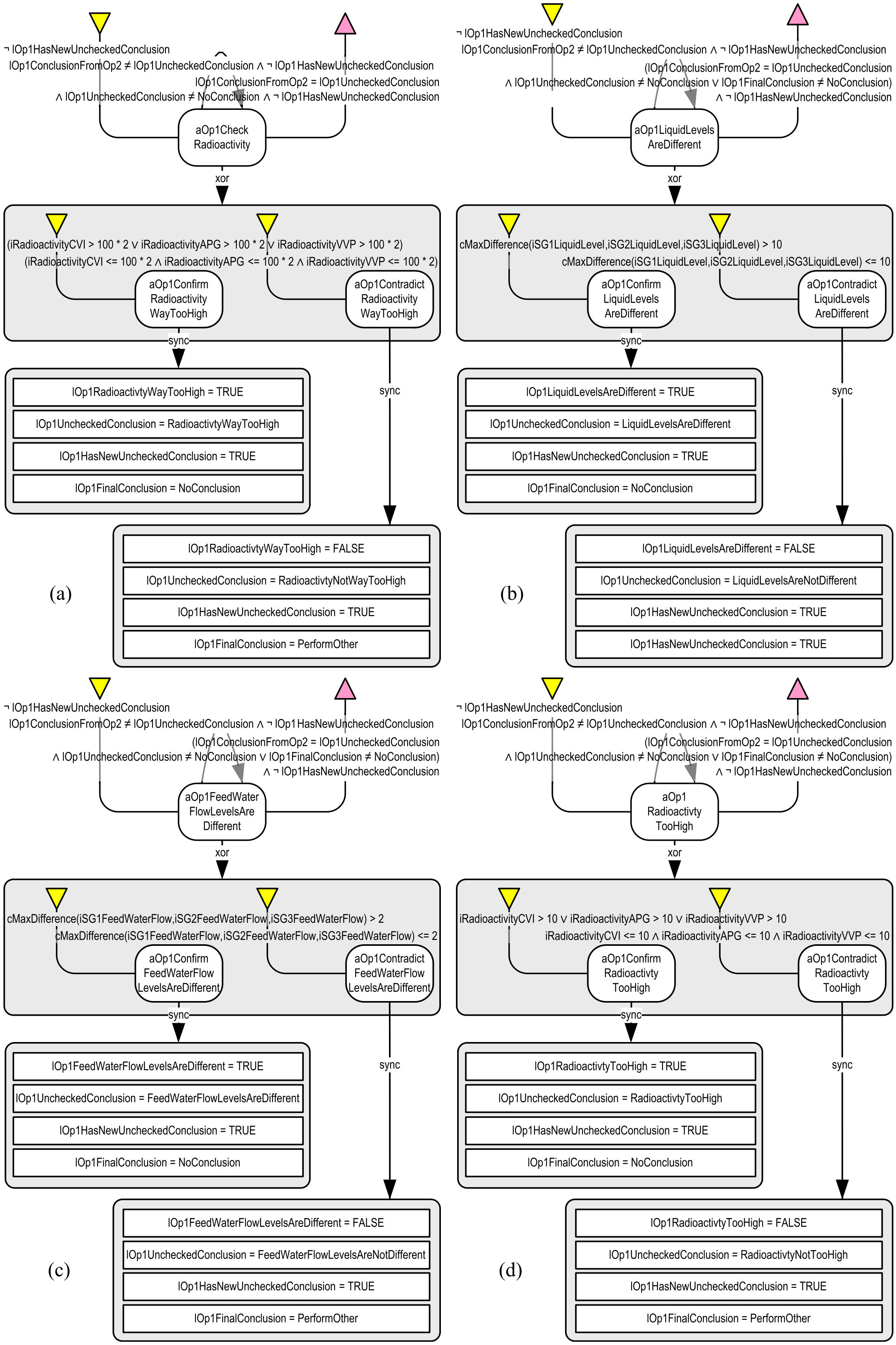
¬ lOp1HasNewUncheckedConclusion
lOp1ConclusionFromOp2 ≠ lOp1UncheckedConclusion ∧ ¬ lOp1HasNewUncheckedConclusion
lOp1ConclusionFromOp2 = lOp1UncheckedConclusion
∧ lOp1UncheckedConclusion ≠ NoConclusion ∧ ¬ lOp1HasNewUncheckedConclusion
aOp1Check Radioactivity
xor
(iRadioactivityCVI > 100 * 2 ∨ iRadioactivityAPG > 100 * 2 ∨ iRadioactivityVVP > 100 * 2)
(iRadioactivityCVI <= 100 * 2 ∧ iRadioactivityAPG <= 100 * 2 ∧ iRadioactivityVVP <= 100 * 2)
aOp1Confirm Radioactivity WayTooHigh
aOp1Contradict Radioactivity WayTooHigh
sync
lOp1RadioactivtyWayTooHigh = TRUE
lOp1UncheckedConclusion = RadioactivtyWayTooHigh
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = NoConclusion
sync
lOp1RadioactivtyWayTooHigh = FALSE
lOp1UncheckedConclusion = RadioactivtyNotWayTooHigh
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = PerformOther
(a)
¬ lOp1HasNewUncheckedConclusion
lOp1ConclusionFromOp2 ≠ lOp1UncheckedConclusion ∧ ¬ lOp1HasNewUncheckedConclusion
(lOp1ConclusionFromOp2 = lOp1UncheckedConclusion
∧ lOp1UncheckedConclusion ≠ NoConclusion ∨ lOp1FinalConclusion ≠ NoConclusion)
∧ ¬ lOp1HasNewUncheckedConclusion
aOp1LiquidLevels AreDifferent
xor
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) > 10
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) <= 10
aOp1Confirm LiquidLevels AreDifferent
aOp1Contradict LiquidLevels AreDifferent
sync
lOp1LiquidLevelsAreDifferent = TRUE
lOp1UncheckedConclusion = LiquidLevelsAreDifferent
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = NoConclusion
sync
lOp1LiquidLevelsAreDifferent = FALSE
lOp1UncheckedConclusion = LiquidLevelsAreNotDifferent
lOp1HasNewUncheckedConclusion = TRUE
lOp1HasNewUncheckedConclusion = TRUE
(b)
¬ lOp1HasNewUncheckedConclusion
lOp1ConclusionFromOp2 ≠ lOp1UncheckedConclusion ∧ ¬ lOp1HasNewUncheckedConclusion
(lOp1ConclusionFromOp2 = lOp1UncheckedConclusion
∧ lOp1UncheckedConclusion ≠ NoConclusion ∨ lOp1FinalConclusion ≠ NoConclusion)
∧ ¬ lOp1HasNewUncheckedConclusion
aOp1FeedWater FlowLevelsAre Different
xor
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) > 2
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) <= 2
aOp1Confirm FeedWaterFlow LevelsAreDifferent
aOp1Contradict FeedWaterFlow LevelsAreDifferent
sync
lOp1FeedWaterFlowLevelsAreDifferent = TRUE
lOp1UncheckedConclusion = FeedWaterFlowLevelsAreDifferent
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = NoConclusion
sync
lOp1FeedWaterFlowLevelsAreDifferent = FALSE
lOp1UncheckedConclusion = FeedWaterFlowLevelsAreNotDifferent
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = PerformOther
(c)
¬ lOp1HasNewUncheckedConclusion
lOp1ConclusionFromOp2 ≠ lOp1UncheckedConclusion ∧ ¬ lOp1HasNewUncheckedConclusion
(lOp1ConclusionFromOp2 = lOp1UncheckedConclusion
∧ lOp1UncheckedConclusion ≠ NoConclusion ∨ lOp1FinalConclusion ≠ NoConclusion)
∧ ¬ lOp1HasNewUncheckedConclusion
aOp1 Radioactivty TooHigh
xor
iRadioactivityCVI > 10 ∨ iRadioactivityAPG > 10 ∨ iRadioactivityVVP > 10
iRadioactivityCVI <= 10 ∧ iRadioactivityAPG <= 10 ∧ iRadioactivityVVP <= 10
aOp1Confirm Radioactivty TooHigh
aOp1Contradict Radioactivty TooHigh
sync
lOp1RadioactivtyTooHigh = TRUE
lOp1UncheckedConclusion = RadioactivtyTooHigh
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = NoConclusion
sync
lOp1RadioactivtyTooHigh = FALSE
lOp1UncheckedConclusion = RadioactivtyNotTooHigh
lOp1HasNewUncheckedConclusion = TRUE
lOp1FinalConclusion = PerformOther
(d)

Fig. 7. EOFMC visualizations of the EOFMC activities used by Op1 from the task from Fig. 6.

Thus, future work should investigate how the presented method and performance levels could be adapted for use in these domains. Such an undertaking would be significantly facilitated by a more general theory for identifying performance levels.

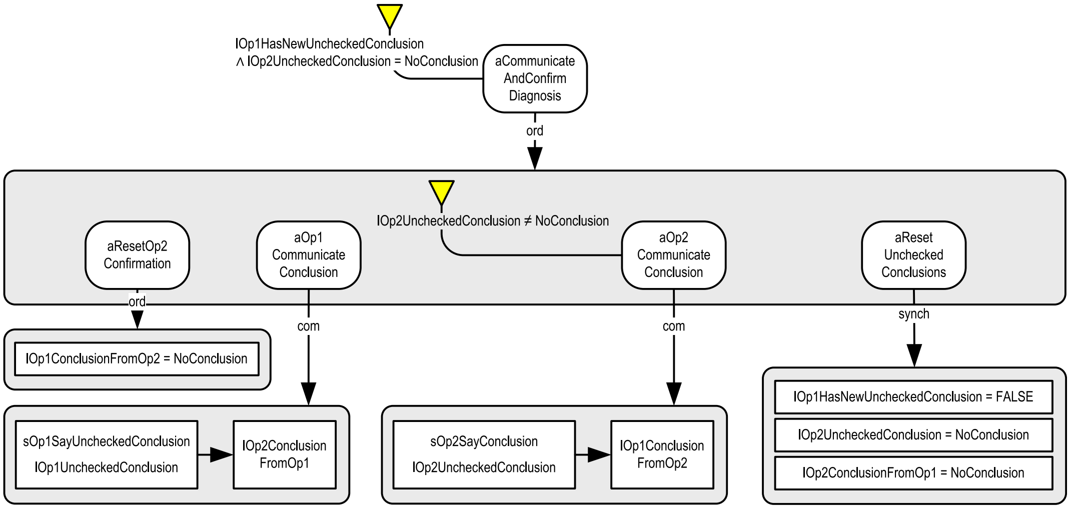
lOp1HasNewUncheckedConclusion
∧ lOp2UncheckedConclusion = NoConclusion
aCommunicate AndConfirm Diagnosis
ord
lOp2UncheckedConclusion ≠ NoConclusion
aResetOp2 Confirmation
aOp1 Communicate Conclusion
aOp2 Communicate Conclusion
aReset Unchecked Conclusions
ord
com
com
synch
lOp1ConclusionFromOp2 = NoConclusion
sOp1SayUncheckedConclusion
lOp1UncheckedConclusion
lOp2Conclusion FromOp1
sOp2SayConclusion
lOp2UncheckedConclusion
lOp1Conclusion FromOp2
lOp1HasNewUncheckedConclusion = FALSE
lOp2UncheckedConclusion = NoConclusion
lOp2ConclusionFromOp1 = NoConclusion

Fig. 8. Visualization of the EOFMC task representing shared collaborative procedure 2.

Table 6
Formal verification Results for Collaborative procedure 2.

	sMax								
cMax	0			1			2		
	Level	# States	Time (s)	Level	# States	Time (s)	Level	# States	Time (s)
0	I	25,211,740,288	17.02	III	881,824,480,476	208.8	VI	12,801,219,833,968	1671.98
1	II	732,831,393,520	84.00	VI	31,826,874,171,324	2227.49			
2	III	16,612,537,230,760	361.02						
3	III	115,923,457,773,650	1500.36						
4	III	376,025,372,194,960	2204.77						

8.2. Generalized performance level identification

The specifications used to assess performance levels that we presented here are specific to the application we used. Ideally, we would be able to create specifications representing different performance levels for any procedure, domain, or number of humans based on a generic theory. For example, grounding theory (Clark and Brennan, 1991) models human behavior based on the behaviors humans use to establish a shared understand (a common ground) of a situation through communication. Such a theory could provide the foundation for a generic process for establishing formal performance levels for different communication protocols with different numbers of human operators and different distributions of authority and autonomy. As we mentioned previously, the diagnosis outcomes are consistent with the outcomes Jones and Endsley (2002) identified for team situation awareness assessment. This perspective has been extended to account for team situation awareness for teams with more than two members (Saner et al., 2009). Such framework could serve as the basis for extending our method to more general use. Beyond increasing the applicability of our findings, such a general framework could enable the automatic generation of specification properties for assessing different levels of performance similar to what is currently done for single human operator systems with EOFM (Bolton et al., 2014). Future work should investigate all of these possibilities.

8.3. Interactions with other erroneous behaviors

The effort presented here is unique in that it is the first formal analyses to investigate how miscommunications and other erroneous human behaviors can interact to impact the performance of a collaborative procedure. However, there are many opportunities for expanding the scope of erroneous behaviors considered in this analysis. Besides miscommunications and slips, EOFMCs can generate other types of erroneous human behavior (Bolton et al., 2012). Further, there are other types of human-human communication failures (Jones, 1999) that can occur beyond miscommunication. Future work should investigate how these and other types of erroneous behavior and communication problems can be incorporated into the method.

8.4. General limitations of formal methods

Formal methods have traditionally been used in the analysis of computer hardware and software systems. However, their usefulness in industrial applications has been acknowledged, albeit with some skepticism. This is largely because formal methods have some distinct limitations. For example, while IEC 61508-7 (International Electrotechnical Commission, 2010) does acknowledge that formal methods offer “unambiguous system description … which increase understanding of the underlying system”, they have a number of problems. These include (International Electrotechnical Commission, 2010): (a) a “fixed level of abstraction”; (b) “limitations to capture all functionality that is relevant at the given stage”; (c) the “difficulty that implementation engineers have to understand the model”; (d) the “high efforts to develop, analyze and maintain model over the lifecycle of system”; (e) the “availability of efficient tools which support the building and analysis of model”; and (f) the “availability of staff capable to develop and analyze model.” The work here goes some distance towards addressing these issues. Specifically, by only focusing on the analysis of collaborative procedures, we are working within a fixed level of abstraction where we can provide useful analysis insights. Thus we avoid issues (a) and (b) by appropriately scoping our analyses. Further, we use task analytic models in our analyses, which are familiar to human factors engineers. We also provide automated tools (a translator, erroneous behavior generation, a counterexample visualizer, etc.). All of these facilitate the use of formal methods by non-experts help address issues (b) – (f). As research of formal methods continues, progress will continue to be made in these areas and formal methods will be both easier to apply on their own and synergistically with other analyses.

Because the presented approach uses model checking, it is limited by the combinatorial explosion problem that faces all model checking analyses. That is, the statespace of the formal model being verified increases exponentially as concurrent elements are added to the model (Clarke et al., 1999). The verification results of procedures 1 and 2 (Tables 5 and 6 respectively) do seem to suggest that exponential increases in statespace size and verification time occur as cMax and sMax increase given that we observe order of magnitude increases in both measure as cMax and sMax increase. This behavior could limit the procedures our approach could be used to evaluate and/or the number of miscommunications and slips that can be considered in an analysis. This is especially true as the number of human team members and the complexity of their tasks increases. Recent developments have shown that a significant improvement (often by several orders of magnitude) in the scalability of EOFMC analyses can be achieved by optimizing the way the translator represents EOFMC-captured behavior formally (Bolton et al., ND). Latter work should see if these scalability improvements can be applied to the method presented here. Additional scalability improvement should also be pursued in the future.

References

Bass, E.J., Bolton, M.L., Feigh, K., Griffith, D., Gunter, E., Mansky, W., Rushby, J., 2011. Toward a multi-method approach to formalizing human-automation interaction and human-human communications. In: Proceedings of 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1817–1824.

Bochmann, G.V., Sunshine, C.A., 1980. Formal methods in communication protocol design. IEEE Trans. Commun. 28 (4), 624–631.

Bolton, M.L., 2015. Model checking human-human communication protocols using task models and miscommunication generation. J. Aerosp. Comput. Inf. Commun. 12, 476–489.

Bolton, M.L., Bass, E.J., 2010. Using task analytic models to visualize model checker counterexamples. In: Proceedings of the 2010 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, Piscataway, pp. 2069–2074.

Bolton, M.L., Bass, E.J., 2013. Generating erroneous human behavior from strategic knowledge in task models and evaluating its impact on system safety with model checking. IEEE Trans. Syst. Man Cybern. Syst. 43 (6), 1314–1327.

Bolton, M.L., Bass, E.J., Siminiceanu, R.I., 2012. Generating phenotypical erroneous human behavior to evaluate human–automation interaction using model checking. Int. J. Hum. Comput. Stud. 70 (11), 888–906.

Bolton, M.L., Bass, E.J., Siminiceanu, R.I., 2013. Using formal verification to evaluate human-automation interaction: a review. IEEE Trans. Syst. Man Cybernetics Syst. 43 (3), 488–503.

Bolton, M.L., Jimenez, N., van Paassen, M.M., Trujillo, M., 2014. Automatically generating specification properties from task models for the formal verification of human-automation interaction. IEEE Trans. Hum. Mach. Syst. 44 (5), 561–575.

Bolton, M.L., Siminiceanu, R.I., Bass, E.J., 2011. A systematic approach to model checking human–automation interaction using task analytic models. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 41 (5), 961–976.

Bolton, M.L., Zheng, X., Molinaro, K., Houser, A., Li, M., ND. Improving the scalability of formal human-automation interaction verification analyses that use task analytic models. Under review in Innovation in Systems and Software Engineering: A NASA Journal, http://dx.doi.org/10.1007/s11334-016-0272-z. Print ISSN: 1614-5046; Online ISSN: 1614-5054..

Clark, H.H., Brennan, S.E., 1991. Grounding in communication. In: Resnick, L.B., Levine, J.M., Teasley, J.S.D. (Eds.), Perspectives on Socially Shared Cognition. American Psychological Association.

Clarke, E.M., Grumberg, O., Peled, D.A., 1999. Model Checking. MIT Press, Cambridge.

Clarke, E.M., Wing, J.M., 1996. Formal methods: state of the art and future directions. ACM Comput. Surv. (CSUR) 28 (4), 626–643.

Connell, L., 1996. Pilot and controller communication issues. In: Methods and Metrics of Voice Communication, pp. 19–27.

Cooper, G.E., White, M.D., Lauber, J.K., 1980. Resource management on the flight-deck. In: Proceedings of a NASA/Industry Workshop (NASA CP-2120).

De Moura, L., Owre, S., Rueß, H., Rushby, J., Shankar, N., Sorea, M., Tiwari, A., 2004. SAL 2. In: Alur, R., Peled, D.A. (Eds.), Proceedings of 16th International Conference on Computer Aided Verification. Springer Berlin, Heidelberg, pp. 496–500.

Dong, X., 2010. Influence of Human-system Interface Design Method and Time Pressure on Human Error (Master thesis). Tsinghua University, Beijing, China.

Emerson, E.A., 1990. Temporal and modal logic. In: van Leeuwen, J. (Ed.), Handbook of Theoretical Computer Science. MIT Press, Cambridge, pp. 995–1072.

Farris, M.C., Barshi, I., 2013. Misunderstandings in ATC Communication: Language, Cognition, and Experimental Methodology. Ashgate.

Guangdong nuclear power training center, 2005. Devices and Systems of 900 MW Pressurized Water Reactor. Atomic Energy Press, Beijing.

Hirotsu, Y., Suzuki, K., Kojima, M., Takano, K., 2001. Multivariate analysis of human error incidents occurring at nuclear power plants: several occurrence patterns of observed human errors. Cognit. Technol. Work 3 (2), 82–91.

International Electrotechnical Commission, 2010. IEC 61508-7:2010: Functional Safety of Electrical/electronic/programmable Electronic Safety-related Systems - Part 7: Overview of Techniques and Measures. IEC, Geneva, Switzerland.

Jones, D.G., Endsley, M.R., 2002. Measurement of Shared SA in Teams: Initial Investigation (No. SATech-02–05). SA Technologies, Marrietta, GA.

Jones, P.M., 1999. Human error and its amelioration. In: Sage, A.P., Rouse, W.B. (Eds.), Handbook of Systems Engineering and Management. John Wiley, pp. 687–702.

Kim, M.C., Park, J., Jung, W., Kim, H., Kim, Y.J., 2010. Development of a standard communication protocol for an emergency situation management in nuclear power plants. Ann. Nucl. Energy 37 (6), 888–893.

MacDonald, P.E., Shah, V.N., Ward, L.W., Ellison, P.G., 1996. Steam Generator Tube Failures. NUREG/CR-6365, INEL-95/0393. Nuclear Regulatory Commission, Washington, DC, United States.

Murphy, P., 2001. The role of communications in accidents and incidents during rail possessions. Eng. Psychol. Cogn. Ergon. 5, 447–454.

Pan, D., Bolton, M.L., 2015. A formal method for evaluating the performance level of human-human collaborative procedures. In: Proceedings of HCI International 2015, Cross-cultural Design Methods, Practice and Impact. Springer International Publishing, pp. 186–197.

Paternò, F., Mancini, C., Meniconi, S., 1997. Concurtasktrees: a diagrammatic notation for specifying task models. In: Howard, S., Hammond, J., Lindgaard, G. (Eds.), Human-Computer Interaction INTERACT'97. Springer, US, pp. 362–369.

Paternò, F., Santoro, C., Tahmassebi, S., 1998. Formal models for cooperative tasks: concepts and an application for en-route air traffic control. In: Proceedings of the 5th International Conference on Design, Specification, and Verification of Interactive Systems. Springer, Vienna, pp. 71–86.

Reason, J., 1990. Human Error. Cambridge University Press, New York.

Saner, L.D., Bolstad, C.A., Gonzalez, C., Cuevas, H.M., 2009. Measuring and predicting shared situation awareness in teams. J. Cogn. Eng. Decis. Mak. 3 (3), 280–308.

Sidhu, D., Leung, T.K., 1989. Formal methods for protocol testing: a detailed study. IEEE Trans. Softw. Eng. 15 (4), 413–426.

Sträter, O., 2003. Investigation of Communication Errors in Nuclear Power Plants. Communication in High Risk Environments. In: Linguistische Berichte, Sonderheft, vol. 12, pp. 155–179.

Traum, D., Dillenbourg, P., 1996. Miscommunication in multi-modal collaboration. In: AAAI Workshop on Detecting, Repairing, and Preventing Human-machine Miscommunication, pp. 37–46.

Wilson, R.M., Runciman, W.B., Gibberd, R.W., Harrison, B.T., Newby, L., Hamilton, J.D., 1995. The quality in Australian health care study. Med. J. Aust. 163 (9), 458–471.

Please cite this article in press as: Pan, D., Bolton, M.L., Properties for formally assessing the performance level of human-human collaborative procedures with miscommunications and erroneous human behavior, International Journal of Industrial Ergonomics (2016), http://dx.doi.org/10.1016/j.ergon.2016.04.001