A Formal Method for Evaluating the Performance Level of Human-Human Collaborative Procedures

Dan Pan[1] and Matthew L. Bolton[2](✉)

[1] Department of Industrial Engineering, Tsinghua University, Beijing 100084, People's Republic of China
pandl0@mails.tsinghua.edu.cn

[2] Department of Industrial and Systems Engineering, University at Buffalo, State University of New York, Amherst, NY 14260, USA
mbolton@buffalo.edu

Abstract. Human-human interaction is critical to safe operations in domains like nuclear power plants (NPP) and air transportation. Usually collaborative procedures and communication protocols 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 between humans. However, these procedures can be complex and thus fail in unanticipated ways. To address this, researchers have been investigating how formal verification can be used to prove the robustness of collaborative procedures to miscommunications. However, previous efforts have taken a binary approach to assessing the success of such procedures. This can be problematic because some failures may be more desirable than others. In this paper, we show how specification properties can be created to evaluate the level of success of a collaborative procedure formally. We demonstrate the capability of these properties to evaluate a realistic procedure for a NPP application.

Keywords: Formal method · Human communication · Human error

1 Introduction

Human collaboration is essential to team performance. By collaborating, team members perform different tasks and share information to ensure mutual understanding. Although the importance of human collaboration is self-evident, its successful execution is anything but guaranteed. Failures and breakdowns in human collaboration have been associated with many accidents and incidents. For example, communication errors have been implicated in a significant percentage of accidents in the workplace in general [12], roadway accidents [10], medical deaths [14] and aviation accidents [6]. Communication error is also one of the main causes of accidents and incidents in nuclear power plants (NPPs). For example, 25 % of Japanese [8] and 10 % of German [12] NPP incidents were caused by communication failure.

From these data, we can conclude that if the human teammates could communicate and collaborate better, the safety of many systems would be improved. Standard

P.L.P. Rau (Ed.): CCD 2015, Part I, LNCS 9180, pp. 186–197, 2015.
DOI: 10.1007/978-3-319-20907-4_17

collaborative procedures and communication protocols are used to ensure effective and efficient collaboration in many safety-critical systems. Such procedures are used in air traffic control communications, operations in the MCR of nuclear power plants, practices of surgical teams, and handoff of care protocols in hospitals. However, ensuring that collaborative procedures are robust to all operational conditions is difficult. There is concurrency between the parts of procedures which different operators execute, which can induce unanticipated interactions between people. Further, humans are fallible and can miscommunicate. Thus, it can be difficult to evaluate the safety of collaborative procedures using conventional analyses like experimentation and simulation since they can miss unexpected interactions.

Formal verification, which is a form of mathematical proof, offers analysis techniques capable of considering all of the possible interactions. While formal methods have been used to evaluate machine communication protocols, these methods are ill-suited for use with human collaborative procedures for several reasons. First, humans behave differently from machines. Humans follow tasks as opposed to machine code and human communication must be contextualized as part of a task [13]. Second, humans are fallible in ways that are different from machines. Thirdly, human collaborative procedures are inherently less fragile because of the looser dynamics of human-human communication. As such, the outcome of a human communication may represent degrees of success beyond the binary (correct or incorrect). For example, if two operators are attempting to diagnose a problem, it is problematic if the operators 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.

Work has evaluated procedures in both collaborative and non-collaborative contexts formally to determine if they are safe, even with generated erroneous behavior and/or miscommunications [2, 3]. However, these analyses are still limited in that they only consider the binary success of human collaboration. This is constraining because it does not give analysts the tools they need to fully evaluate the robustness of such procedures. Thus, there is a real need for an approach that will account for miscommunication while giving analysts metrics for evaluating the degrees of a procedure's success in different conditions.

In this paper, we extend an approach [12] 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.

2 Background

2.1 Formal Method

Formal methods are tools and techniques for proving that a system will always perform as intended [5]. Model checking is an automated approach to formal verification. In model checking, a system model represents of a system's behavior in a mathematical formalism (usually a finite state machine). A specification represents a formal description of a desirable property about the system, usually in a temporal logic.

Finally, model checking produces a verification report either a confirmation or a counterexample. A counterexample illustrates incremental model states that resulted in the specification being violated.

There are a variety of temporal and modal logics that have been used for specification. The most common one, and the one used in the presented work, is linear temporal logic (LTL). LTL allows one to assert properties about all of the paths through a model. It does this using model variables and basic Boolean logic operators ($\wedge$, $\vee$, $\neg$, $\Rightarrow$, and $\Leftrightarrow$). Additionally, it has temporal operators that allow for assertions about how variables ordinally change over time. Thus, using LTL, an analyst can assert that something (Φ) should always be true G Φ; that it will always be true in the next state X Φ; that it will be true in the future F Φ; or that it will be true until something else (Ψ) is true Φ U Ψ.

2.2 Formal Methods for Human-Human Communication and Coordination

While formal methods have traditionally been used in the analysis of computer hardware and software systems [12], a growing body of work has been investigating how to use them to evaluate human factors issues [3]. However, when it comes to issues of human-human communication and coordination, there has been very little work. Paternò et al. [11] extended the Concur Task Trees formalism 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). They used this to formally evaluate pilot and air traffic control radio communications during runway operations using different shared task representations. While useful, this method did not easily distinguish between separate and shared operator tasks, nor did it account for potential miscommunications. Both limitations were addressed by the Enhanced Operator Function Model with Communications.

2.3 Enhanced Operator Function Model with Communication (EOFMC)

EOFMC [1] extended the Enhanced Operator Function Model (EOFM) [4] 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 as an input/output system. Inputs represent human interface, environment, and/or mission goal concepts. Outputs are human actions. The operators' task models (local variables) describe how human actions are produced and how the internal state of the human (perceptual or cognitive) changes.

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 (acts) can execute and what the temporal relationship is between them. In the presented work, only the following decomposition operators are important:

`sync` – all acts must be performed synchronously (at the exact same time);
`xor` – exactly one act must be performed;
`and_seq` – all of the acts must be performed, one at a time, in any order;
`ord` – all of the acts 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 acts in the given decomposition, the children that decompose from it, and its strategic knowledge. These semantics are the basis for the EOFMC translator that allows EOFMC models to be automatically incorporated into the input language of the Symbolic Analysis Laboratories family of model checkers.

Bass et al. [1] used EOFMC to model and evaluate communication protocols used to convey clearances between air traffic control and pilots. Bolton [2] 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 (`Max`). The net effect of this is that analysts can evaluate how robust a protocol is for all possible ways that `Max` or fewer miscommunications can occur. Bolton 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 [2], 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.

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 with and without miscommunication. This paper describes an extension of the approach found in [1, 2] 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 and a given number of miscommunications. 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 a 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. 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. Finally, we discuss the results and outline future areas of research.

4 Application

In pressurized water reactors (PWR), a SGTR accident is quite frequent since a variety of degradation processes from the steam generator tubing system can lead to tube cracking, wall thinning, and potential leakage or rupture [9]. The SGTR accidents involve a leak from the reactor coolant system (RCS) to the steam generator (SG) that leads to the primary coolant flowing into the secondary system. If the safety systems are unavailable, or operators take incorrect or late actions, the secondary pressure will increase rapidly. The secondary water or vapor with radioactive substances will be released into the environment. Even more seriously, the loss of reactor coolant may cause core damage. Once the core damage occurs, and if the containment is bypassed, serious radioactivity release will happen [9]. With its high occurrence frequency and capacity for causing serious radioactivity consequences, the operators' successful intervention after a SGTR accident is vital for system safety.

The example used in this study occurs in a 900MWe pressurized water reactor NPP where an alarm indicates if safety injection has lasted over 5 min. Two operators (operator 1 and operator 2) in main control need to collaborate to diagnose whether it is a SGTR accident.

4.1 SGTR Diagnosis Procedure and Communication Protocol

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.

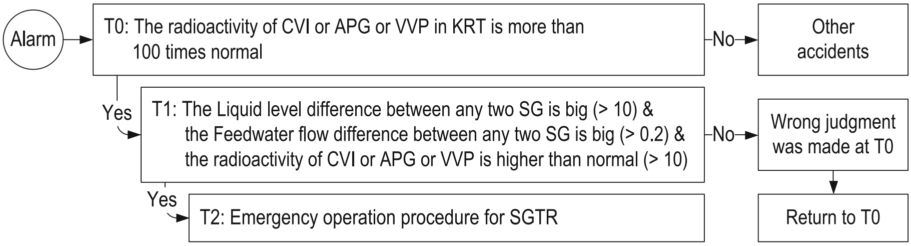
Alarm
T0: The radioactivity of CVI or APG or VVP in KRT is more than 100 times normal
No
Other accidents
Yes
T1: The Liquid level difference between any two SG is big (> 10) & the Feedwater flow difference between any two SG is big (> 0.2) & the radioactivity of CVI or APG or VVP is higher than normal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

Fig. 1. SGTR diagnosis procedure

At 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 a SGTR accident and proceed to other diagnostic procedures (not discussed here). If true, the operators should proceed to 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, namely 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 a SGTR accident has occurred and perform other diagnostic procedures (see [7]).

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, feedwater flow, 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.

4.2 Modeling

The SGTR diagnosis procedure was implemented as an instantiated EOFMC (visualized in Figs. 2, 3 and 4). This model has two human operators: Op1 and Op2. Op1 is responsible for working through the SGTR diagnosis procedure (Fig. 2). 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`). If it isn't, he concludes that something else is wrong with the system. If it is, he must check the liquid levels, the feedwater flow, and the radioactivity in any order in accordance with the SGTR procedure (`aOp1CheckT1`). If all of these are consistent with a SGTR accident, he should conclude (`aOp1FormConclusion`) 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.

At any stage in this process, when Op1 reaches an intermediate or final conclusion (that radioactivity is too high, that feedwater flows are different, that liquid levels are different, that procedure T2 must be performed, etc.), he must confirm that conclusion with Op2 before he can complete the associated activity.

This confirmation process occurs via the previously discussed communication protocol, which is represented in the model as a shared task (Fig. 3). In this, when Op1 has an unchecked conclusion, he must communicate that conclusion to Op2. If Op2 agrees with the conclusion he will communicate back an "Agree", otherwise he will communicate a "Disagree". Note that at the beginning and end of the communication protocol, variables are reset to ensure proper communications between the different tasks (Figs. 2, 3 and 4) in the model.

Op2 is responsible for the procedures he uses to determine whether he agrees or disagrees with Op1's conclusions using the tasks pattern in Fig. 4. 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.

The complete, instantiated EOFMC task model was converted into the language of the Symbolic Analysis Laboratory (SAL) using the EOFMC java-based translator [4]. 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 (`Max`) was set from 0 to 4 in increments of 1.

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

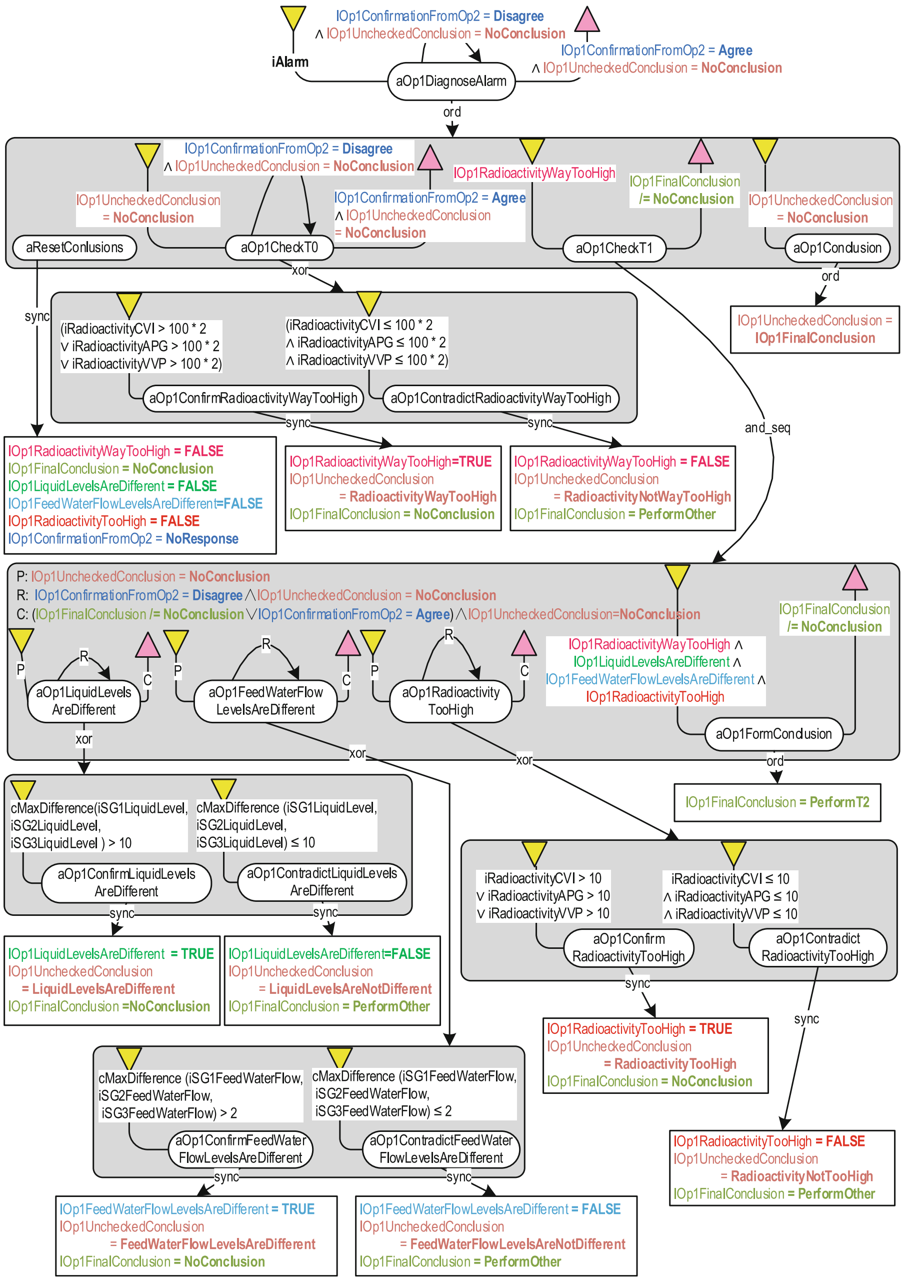
iAlarm
lOp1ConfirmationFromOp2 = Disagree
∧ lOp1UncheckedConclusion = NoConclusion
lOp1ConfirmationFromOp2 = Agree
∧ lOp1UncheckedConclusion = NoConclusion
aOp1DiagnoseAlarm
ord
lOp1ConfirmationFromOp2 = Disagree
∧ lOp1UncheckedConclusion = NoConclusion
lOp1UncheckedConclusion
= NoConclusion
aResetConlusions
aOp1CheckT0
lOp1ConfirmationFromOp2 = Agree
∧ lOp1UncheckedConclusion
= NoConclusion
lOp1RadioactivityWayTooHigh
aOp1CheckT1
lOp1FinalConclusion
/= NoConclusion
lOp1UncheckedConclusion
= NoConclusion
aOp1Condusion
ord
lOp1UncheckedConclusion =
lOp1FinalConclusion
xor
sync
(iRadioactivityCVI > 100 * 2
∨ iRadioactivityAPG > 100 * 2
∨ iRadioactivityVVP > 100 * 2)
(iRadioactivityCVI ≤ 100 * 2
∧ iRadioactivityAPG ≤ 100 * 2
∧ iRadioactivityVVP ≤ 100 * 2)
aOp1ConfirmRadioactivityWayTooHigh
aOp1ContradictRadioactivityWayTooHigh
sync
sync
and_seq
lOp1RadioactivityWayTooHigh = FALSE
lOp1FinalConclusion = NoConclusion
lOp1LiquidLevelsAreDifferent = FALSE
lOp1FeedWaterFlowLevelsAreDifferent=FALSE
lOp1RadioactivityTooHigh = FALSE
lOp1ConfirmationFromOp2 = NoResponse
lOp1RadioactivityWayTooHigh=TRUE
lOp1UncheckedConclusion
= RadioactivityWayTooHigh
lOp1FinalConclusion = NoConclusion
lOp1RadioactivityWayTooHigh = FALSE
lOp1UncheckedConclusion
= RadioactivityNotWayTooHigh
lOp1FinalConclusion = PerformOther
P: lOp1UncheckedConclusion = NoConclusion
R: lOp1ConfirmationFromOp2 = Disagree ∧ lOp1UncheckedConclusion = NoConclusion
C: (lOp1FinalConclusion /= NoConclusion ∨ lOp1ConfirmationFromOp2 = Agree) ∧ lOp1UncheckedConclusion=NoConclusion
lOp1FinalConclusion
/= NoConclusion
P
R
C
aOp1LiquidLevels
AreDifferent
P
R
C
aOp1FeedWaterFlow
LevelsAreDifferent
P
R
C
aOp1Radioactivity
TooHigh
lOp1RadioactivityWayTooHigh ∧
lOp1LiquidLevelsAreDifferent ∧
lOp1FeedWaterFlowLevelsAreDifferent ∧
lOp1RadioactivityTooHigh
aOp1FormCondusion
xor
xor
xor
ord
lOp1FinalConclusion = PerformT2
cMaxDifference(iSG1LiquidLevel,
iSG2LiquidLevel,
iSG3LiquidLevel) > 10
cMaxDifference (iSG1LiquidLevel,
iSG2LiquidLevel,
iSG3LiquidLevel) ≤ 10
aOp1ConfirmLiquidLevels
AreDifferent
aOp1ContradictLiquidLevels
AreDifferent
sync
sync
iRadioactivityCVI > 10
∨ iRadioactivityAPG > 10
∨ iRadioactivityVVP > 10
iRadioactivityCVI ≤ 10
∧ iRadioactivityAPG ≤ 10
∧ iRadioactivityVVP ≤ 10
aOp1Confirm
RadioactivityTooHigh
aOp1Contradict
RadioactivityTooHigh
lOp1LiquidLevelsAreDifferent = TRUE
lOp1UncheckedConclusion
= LiquidLevelsAreDifferent
lOp1FinalConclusion =NoConclusion
lOp1LiquidLevelsAreDifferent=FALSE
lOp1UncheckedConclusion
= LiquidLevelsAreNotDifferent
lOp1FinalConclusion = PerformOther
sync
sync
lOp1RadioactivityTooHigh = TRUE
lOp1UncheckedConclusion
= RadioactivityTooHigh
lOp1FinalConclusion = NoConclusion
cMaxDifference (iSG1FeedWaterFlow,
iSG2FeedWaterFlow,
iSG3FeedWaterFlow) > 2
cMaxDifference (iSG1FeedWaterFlow,
iSG2FeedWaterFlow,
iSG3FeedWaterFlow) ≤ 2
aOp1ConfirmFeedWater
FlowLevelsAreDifferent
aOp1ContradictFeedWater
FlowLevelsAreDifferent
lOp1RadioactivityTooHigh = FALSE
lOp1UncheckedConclusion
= RadioactivityNotTooHigh
lOp1FinalConclusion = PerformOther
sync
sync
lOp1FeedWaterFlowLevelsAreDifferent = TRUE
lOp1UncheckedConclusion
= FeedWaterFlowLevelsAreDifferent
lOp1FinalConclusion = NoConclusion
lOp1FeedWaterFlowLevelsAreDifferent = FALSE
lOp1UncheckedConclusion
= FeedWaterFlowLevelsAreNotDifferent
lOp1FinalConclusion = PerformOther

Fig. 2. Visualization of the instantiated EOFMC collaborative procedure representing the task performed by Op1.

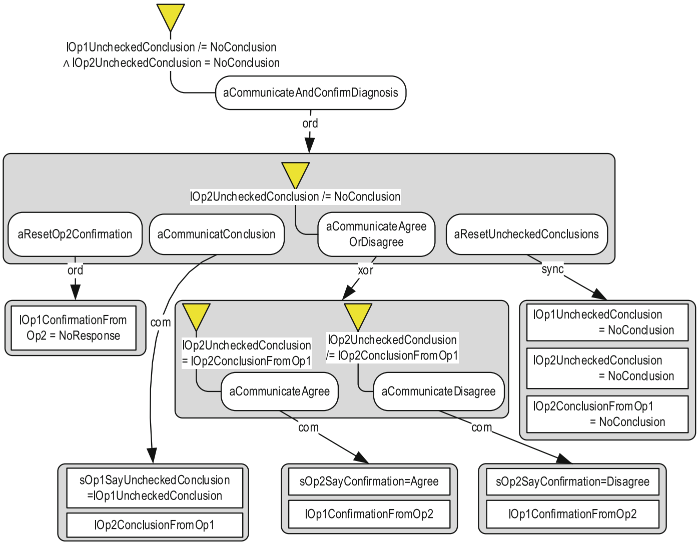
lOp1UncheckedConclusion /= NoConclusion
∧ lOp2UncheckedConclusion = NoConclusion
aCommunicateAndConfirmDiagnosis
ord
lOp2UncheckedConclusion /= NoConclusion
aResetOp2Confirmation
aCommunicatConclusion
aCommunicateAgree OrDisagree
aResetUncheckedConclusions
ord
xor
sync
lOp1ConfirmationFrom Op2 = NoResponse
com
lOp2UncheckedConclusion = lOp2ConclusionFromOp1
lOp2UncheckedConclusion /= lOp2ConclusionFromOp1
aCommunicateAgree
aCommunicateDisagree
lOp1UncheckedConclusion = NoConclusion
lOp2UncheckedConclusion = NoConclusion
lOp2ConclusionFromOp1 = NoConclusion
com
com
sOp1SayUncheckedConclusion =lOp1UncheckedConclusion
lOp2ConclusionFromOp1
sOp2SayConfirmation=Agree
lOp1ConfirmationFromOp2
sOp2SayConfirmation=Disagree
lOp1ConfirmationFromOp2

Fig. 3. Visualization of the EOFMC task representing the shared communication protocol

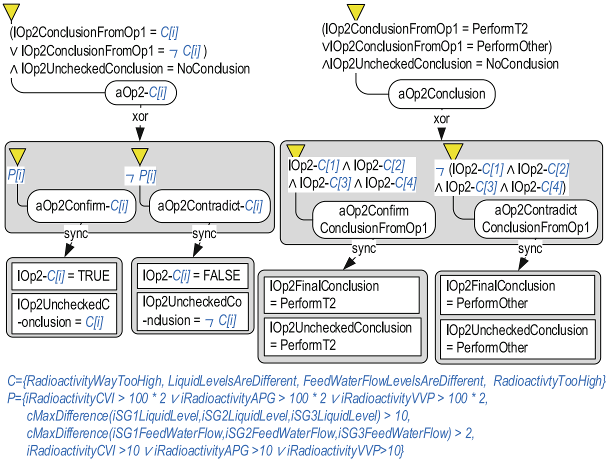
(lOp2ConclusionFromOp1 = C[i]
∨ lOp2ConclusionFromOp1 = ¬ C[i])
∧ lOp2UncheckedConclusion = NoConclusion
aOp2-C[i]
xor
P[i]
¬ P[i]
aOp2Confirm-C[i]
aOp2Contradict-C[i]
sync
sync
lOp2-C[i] = TRUE
lOp2UncheckedC -onclusion = C[i]
lOp2-C[i] = FALSE
lOp2UncheckedCo -nclusion = ¬ C[i]
(lOp2ConclusionFromOp1 = PerformT2
∨lOp2ConclusionFromOp1 = PerformOther)
∧lOp2UncheckedConclusion = NoConclusion
aOp2Conclusion
xor
lOp2-C[1] ∧ lOp2-C[2]
∧ lOp2-C[3] ∧ lOp2-C[4]
¬ (lOp2-C[1] ∧ lOp2-C[2]
∧ lOp2-C[3] ∧ lOp2-C[4])
aOp2Confirm ConclusionFromOp1
aOp2Contradict ConclusionFromOp1
sync
sync
lOp2FinalConclusion = PerformT2
lOp2UncheckedConclusion = PerformT2
lOp2FinalConclusion = PerformOther
lOp2UncheckedConclusion = PerformOther
C={RadioactivityWayTooHigh, LiquidLevelsAreDifferent, FeedWaterFlowLevelsAreDifferent, RadioactivtyTooHigh}
P={iRadioactivityCVI > 100 * 2 ∨ iRadioactivityAPG > 100 * 2 ∨ iRadioactivityVVP > 100 * 2,
cMaxDifference(iSG1LiquidLevel,iSG2LiquidLevel,iSG3LiquidLevel) > 10,
cMaxDifference(iSG1FeedWaterFlow,iSG2FeedWaterFlow,iSG3FeedWaterFlow) > 2,
iRadioactivityCVI >10 ∨ iRadioactivityAPG >10 ∨ iRadioactivityVVP>10}

Fig. 4. Visualization of the EOFMC tasks Op2 uses to determine whether he agrees with Op1. Note that the left task structure presents a pattern Op2 uses to confirm or contradict intermediate conclusions. The right task is used for final conclusions. The parameters that describe both tasks are shown in the figure key.

4.3 Specification

To assess the degree of success of this procedure for different maximum numbers of miscommunications, we needed to derive specifications representing 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 could be 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 1).

Table 1. Diagnosis outcomes

	Description
A	Op1 and Op2 reach the correct final conclusion and the same intermediary conclusions
B	Op1 and Op2 reach the correct final conclusion but differ on the intermediary conclusions
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

In the most desirable outcome (A), both Op1 and Op2 reach the correct final conclusion and the same intermediate conclusions. 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 to A because the difference in intermediary conclusions represents a disagreement in the 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.

The levels were expressed specification properties (Table 2). Each was designed so that, if it verified true, its corresponding level of performance was guaranteed.

4.4 Formal Verification and Results

Formal verifications were performed using SAL's Symbolic Model Checker (SAL-SMC). For each system model with different values of `Max`, all six of the specifications (Table 2) were checked starting with I and working towards VI. At any

Table 2. Specifications of different performance levels

Performance Level	Specification
I	*G (A)*
II	*G (A or B)*
III	*G (A or B or C)*
IV	*G (A or B or C or D)*
V	*G (A or B or C or D or E)*
VI	*G (A or B or C or D or E or F)*

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

point in this process, if a specification verified to true, verifications on that model stopped. The specification that verified to true indicated the performance level guaranteed by that model. These analyses revealed that this collaborative procedure achieves different performance levels in different conditions. For no miscommunications, the model performed at level I. For all other values of `Max`, the model performed at level III (guaranteeing at least an outcome of C). 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 `Max` 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.

5 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 procedure using formal verification. Specifically, by allowing analyst to assess the level of performance guaranteed by a procedure, 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 procedure were formally evaluated in the traditional way, just at level I, it would be considered a failure for all `Max` values greater than 0. By verifying our novel specifications, it is now clear that, although it does not provide 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.

There are a number of directions that could be explored in future work. First, besides miscommunication, other erroneous human behavior can be generated in the formal representation of the EOFMCs [3]. Future work will investigate this possibility. Second, an analyst may wish to compare the performance of different procedures. The presented approach should give analysts means of doing this based on procedure performance levels. Future work should explore how our method could be used to compare procedures. Finally, the specifications presented here are specific to the application we used. Ideally, we would be able to create specifications representing different performance levels for any procedure or domain based on a generic theory. Future work should investigate if such a generic approach is possible.

References

1. Bass, E.J., Bolton, M.L., Feigh, K., Griffith, D., Gunter, E., Mansky, W., Rushby, J.: Toward a multi-method approach to formalizing human-automation interaction and human-human communications. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1817–1824 (2011)
2. Bolton, M.L.: Model checking human-human communication protocols using task models and miscommunication generation. J. Aerosp. Comput. Inf. Commun. doi:10.2514/1.I010276. (in press, 2015)
3. Bolton, M.L., Bass, E.J., Siminiceanu, R.I.: Using formal verification to evaluate human-automation interaction: A review. IEEE Trans. Syst. Man, Cyberne.: Syst. **43**(3), 488–503 (2013)
4. Bolton, M.L., Siminiceanu, R.I., Bass, E.J.: 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 (2011)
5. Clarke, E.M., Wing, J.M.: Formal methods: state of the art and future directions. ACM Comput. Surv. (CSUR) **28**(4), 626–643 (1996)
6. Connell, L.: Pilot and controller communication issues. In: Methods and Metrics of Voice Communication, pp. 19–27 (1996)
7. Dong, X.: Influence of Human-system Interface Design Method and Time Pressure on Human Error. Master thesis. Tsinghua University, Beijing, China (2010)
8. Hirotsu, Y., Suzuki, K., Kojima, M., Takano, K.: Multivariate analysis of human error incidents occurring at nuclear power plants: several occurrence patterns of observed human errors. Cogn. Technol. Work **3**(2), 82–91 (2001)
9. MacDonald, P.E., Shah, V.N., Ward, L.W., Ellison, P.G.: Steam generator tube failures. NUREG/CR-6365, INEL-95/0393. Nuclear Regulatory Commission, Washington, DC, United States (1996)
10. Murphy, P.: The role of communications in accidents and incidents during rail possessions. In: Engineering Psychology and Cognitive Ergonomics, vol. 5, pp. 447–454 (2001)
11. Paternò, F., Santoro, C., Tahmassebi, S.: Formal models for cooperative tasks: concepts and an application for en-route air traffic control. In: Markopoulos, P., Johnson, P. (eds.) Proceedings of the 5th International Conference on Design, Specification, and Verification of Interactive Systems, pp. 71–86. Springer, Vienna (1998)
12. Strater, O.: Investigation of communication errors in nuclear power plants. Communication in High Risk Environments. Linguistische Berichte, Sonderheft **12**, 155–179 (2003)
13. Traum, D., Dillenbourg, P.: Miscommunication in multi-modal collaboration. In: AAAI Workshop on Detecting, Repairing, And Preventing Human–Machine Miscommunication, pp. 37–46 (1996)
14. Wilson, R.M., Runciman, W.B., Gibberd, R.W., Harrison, B.T., Newby, L., Hamilton, J.D.: The quality in Australian health care study. Med. J. Aust. **163**(9), 458–471 (1995)