Using Model Checking to Detect Simultaneous Masking in Medical Alarms

Bassam Hasanain, Andrew D. Boyd, and Matthew L. Bolton, *Member, IEEE*

***Abstract*—The ability of people to hear and respond to auditory medical alarms is critical to the health and safety of patients. Unfortunately, concurrently sounding alarms can perceptually interact in ways that mask one or more of them: making them impossible to hear. Because masking may only occur in extremely specific and/or rare situations, experimental evaluation techniques are insufficient for detecting masking in all of the potential alarm configurations used in medicine. Thus, a real need exists for computational methods capable of determining if masking exists in medical alarm configurations before they are deployed. In this paper, we present such a method. Using a combination of formal modeling, psychoacoustic modeling, temporal logic specification, and model checking, our method is able to prove whether a modeled of a configuration of alarms can interact in ways that produce masking. This paper provides the motivation for this method, presents its details, describes its implementation, demonstrates its power with a case study, and outlines future work.**

***Index Terms*—Formal methods, medical alarms, model checking, patient safety, psychoacoustics, psychophysics.**

I. Introduction

MEDICAL alarms (which are usually auditory) are used by automation to notify human observers that monitored patient health measures have passed a threshold, indicating a potentially unsafe condition that requires immediate attention. The ability of humans to perceive, understand, and respond to alarms is critical to patient safety.

Unfortunately, there are many limitations of modern medical alarm systems [2]. Significant numbers of false alarms can desensitize humans to them (a condition known as alarm fatigue); alarms can be poorly designed, reducing their effectiveness [2]; and concurrently sounding alarms can perceptually interact in ways that make them difficult to identify [3] or mask each other (make one or more of them imperceptible) [4]. Unfortunately, problems caused by the masking of concurrently sounding alarms can be very difficult to identify because they may occur under rare or unusual conditions or through the interaction of particular alarms within or between alarming systems. Thus, while auditory masking has been experimentally detected in clinical settings [5], [6], the vast majority of the work has focused on other aforementioned problem areas [7]. Thus, there is a very real and urgent need for methods capable of identifying if masking is present in medical alarm configurations before they are used in medical practice.

We describe a method we developed that is capable of doing such analyses. Our method makes use of two computational analysis techniques: model checking and psychoacoustic modeling. Model checking is an analysis tool, widely used in the analysis of safety critical computer systems that is specifically designed to find problems in models of concurrent systems using a form of automated theorem proving [8]. Psychoacoustic models are capable of mathematically indicating if concurrently sounding alarms might interact in ways that could produce masking [4], [9], [10]. When used together in our method [1], these techniques can allow healthcare providers to determine if masking exists in a modeled configuration of alarms computationally. With such a detection capability, healthcare providers should be able to deploy systems that will avoid masking: enabling medical personnel to respond to alarms appropriately and potentially save patient lives.

In this paper, we describe this method and illustrate its utility. We first discuss the literature relevant to understanding our method. This includes material on masking in medical alarms, psychoacoustic models of masking, and model checking. We then describe our method: its conceptualization, design, and implementation. To illustrate its power, we use it to evaluate a realistic configuration of medical alarms. We conclude by discussing our results and future avenues of research.

II. Related Work

A. Concurrently Sounding Medical Alarms

Auditory medical alarms have a number of problems [7] making them one of the most significant technological hazards to patient safety for over a decade [11], [12]. The Pennsylvania Patient Safety Authority reports that there have been 194 documented problems with operators' responses to telemetry monitoring alerts from June 2004 to December 2008 resulting in at least 12 deaths [13]. Medical device manufacturers have reported 216 "alarm-related" deaths to the FDA between January 2005 and June 2010 [14]. An event alert issued by the Joint

Manuscript received April 27, 2014; revised November 2, 2014; accepted November 10, 2014. This paper is an expanded version of [1], originally presented at the 2014 International Annual Meeting of the Human Factors and Ergonomics Society. This article significantly extends the description of the utilized modeling and analysis approach. It also builds on the original methodology with additional detection capabilities, presents new modeling and verification results, and features a deeper discussion. This paper was recommended by Associate Editor N. Rungta.

B. Hasanain is with the Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: b.hasanain@hotmail.com).

A. D. Boyd is with the Department of Biomedical and Health Information Sciences, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: boyda@uic.edu).

M. L. Bolton is with the Department of Systems and Industrial Engineering, University at Buffalo, State University of New York, Amherst, NY 14260 USA (e-mail: mbolton@buffalo.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2014.2379661

Commission in April 2013 stated that reports voluntarily submitted to the Joint Commission's Sentinel Event database contained 98 incidents related to alarms from January 2009 to June 2012: 80 of these resulted in patient death, 13 produced "permanent loss of function," and 5 extended the stay of patients in the hospital [15].

There are a number of different perceptual problems that can arise with medical alarms [7]. High numbers of false alarms can degrade human response performance to the point where a person completely fails to notice or respond [16]–[23], a condition known as alarm fatigue. Individual alarms can be designed in ways that are irritating or startling [24], [25], difficult to learn [26]–[28], difficult to interpret [28], do not follow a consistent design philosophy [2], are difficult to distinguish between when concurrently sounding [3], or do not take into account ecological considerations such as background noise [28], [29]. All of these issues can result in alarms that are not detected and/or are not given the proper response. For the purpose of this paper, we are primarily concerned with perceptibility of concurrently sounding alarms. Specifically, alarms that sound in close temporal proximity may produce auditory masking [4], [30], a condition where multiple sounds interact in a way that prevents the human perceptual system from hearing one of or more of them.

There are a number of different sounds that can be used for auditory alarms [31]. However, most alarms are either represented abstractly as sounds with a distinctive tone [2], or as a melody of such sounds [32]. Unfortunately, these types of sounds are particularly susceptible to masking in the presence of other alarms. Although many medical alarm experts have acknowledged auditory masking between concurrent medical alarms as a threat to patient safety [5], [6], [25], [33]–[37], it has been given very little research attention. In an analysis of 49 different alarms used in the intensive care unit and the operating rooms of a Canadian teaching hospital, Momtahan *et al.* [5] found several instances where alarms masked other alarms using a combination of physical auditory measurement, psychophysical modeling, and human subject psychophysical experiments. In a separate analysis, Toor *et al.* [6] used psychoacoustic models to evaluate audio data recorded for medical alarms and other common hospital alarm noises (including phone ringing and beeper notifications) also found masking of medical alarms.

While there are a number of ways that auditory masking can occur [4], [9], [30], the most important for tonal alarms is simultaneous masking. Simultaneous masking describes particular relationships between frequencies and volumes (determined by the human perceptual system) that can result in sounds being undetectable.

As the number of medical alarms increases and more and more alarms from different systems interact, the presence of these masking conditions will likely significantly increase [25]. Further, it is impractical to expect hospitals to use the experimental techniques of [5] and [6] to detect masking conditions in all of the possible alarm configurations that could occur in the hospital. Luckily, psychoacoustic models exist that are capable of detecting if simultaneous masking will occur between concurrent sounds.

B. Psychoacoustic Models of Simultaneous Masking

A number of models exist for predicting auditory masking [4], [9], [10], [30], [38]–[41]. However, psychoacoustic models are the most appropriate to this study because they quantitatively relate a sound's physical characteristics (its frequency/tone and volume) to the masking effect the sound has on human perception using mathematical formulas. The most successful of these use heuristics based on the expected excitation patterns of the human ear's basilar membrane (the physical structure largely responsible for allowing humans to distinguish between different sounds) [9], [42]–[46].

These psychoacoustic models represent a sound's masking threshold for different frequencies of concurrently occurring sounds (its masking curve) as a function of the sound's volume in decibels (dB) and frequency in Barks. The Bark scale is psychoacoustic in that it represents a sound's frequency from 1 to 24 [47], indicating which of the 24 critical bands of hearing the sound falls in. For a given sound (sound) with a frequency (f_{sound}) in hertz, the frequency is converted into the Bark scale using the formula

$$\begin{aligned} z_{\mathrm{sound}} &= 13 \cdot \arctan(0.00076 \cdot f_{\mathrm{sound}}) \\ &\quad + 3.5 \cdot \arctan\left(\left(f_{\mathrm{sound}}/7500\right)^2\right) \end{aligned} \tag{1}$$

where z_{sound} is the frequency of the sound in Barks [48].

The masking curve for a given sound (a masker) is generally formulated as a function of both the sound and its frequency's distance from another, potentially masked, sound (a maskee) on the Bark scale. This difference, δz, is represented as

$$\delta z = z_{\mathrm{maskee}} - z_{\mathrm{masker}}. \tag{2}$$

Then, the masker's masking curve is represented as

$$\begin{aligned} \mathsf{curve}_{\mathsf{masker}}(v_{\mathrm{masker}}, \delta z) &= \mathsf{spread}_{\mathsf{masker}}(v_{\mathrm{masker}}, \delta z) \\ &\quad + v_{\mathrm{masker}} - \Delta \end{aligned} \tag{3}$$

where v_{masker} is the volume of the masker in dB, spread is a function that defines how the volume changes as δz moves away from zero, and Δ represents the minimum difference between a masker's and maskee's volume under which masking can occur [9].

There are a number of different psychoacoustic spreading functions that have been developed. Each makes tradeoffs between misses and false alarms in the detection of masking [9] and have been tuned to different applications. For example, many of these spreading functions were developed to compute the masking functions that are used in lossy audio compression formats like MPEG 2 and MP3 [9], where masked audio data are removed to reduce file size.

For example, the spreading function used as the basis for the MPEG2 audio codec [43] is formulated as

$$\begin{aligned} \mathsf{spread}_{\mathsf{masker}}(\delta z) &= 15.81 + 7.5 \cdot (\delta z + 0.474) \\ &\quad - 17.5 \cdot \sqrt{1 + (\delta z + 0.474)^2}. \end{aligned} \tag{4}$$

[1] A critical band is the frequency bandwidth of the filters produced by the inner ear's basilar membrane in the cochlea.

This spreading function is tuned to normal hearing. It also has only one independent variable (δz). However, other spreading functions can take volume (v_{masker}) as an argument [9].

There can be different formulations of Δ depending on the nature of the sound. For tonal maskers [49], like those used in most medical alarms, Δ (in dB) is formulated as

$$\Delta = 14.5 + z_{\mathrm{masker}}. \tag{5}$$

For a given masking curve, we know that the masker (with volume v_{masker} and Bark frequency z_{masker}) is masking the maskee (with volume v_{maskee} and frequency z_{maskee}) if

$$\text{curve}_{\text{masker}}(v_{\mathrm{masker}}, z_{\mathrm{maskee}} - z_{\mathrm{masker}}) \geq v_{\mathrm{maskee}}. \tag{6}$$

C. Formal Verification With Model Checking

Formal verification is an analysis technique that falls within the broader category of formal methods. Formal methods are a set of well-defined mathematical languages and techniques for the specification, modeling, and verification of systems [50]. Specifications are formulated to describe desirable system properties in rigorous unambiguous notations. Systems are modeled using mathematically based languages that support well-established theoretical formalisms such as finite-state automata. The verification process mathematically proves whether or not the model satisfies the specification. Formal verification has been used successfully in a number of applications, especially computer hardware design, where performance must be guaranteed.

Model checking is a highly automated approach used to verify that a formal model of a system satisfies a set of desired properties (a specification) [51]. A formal model describes a system as a set of variables and transitions between variable states. Specification properties are usually represented in a temporal logic using formal system model variables and temporal operators to construct propositions asserting temporal relationships between system elements [52]. Verification is performed automatically by exhaustively searching a system's state space to determine if these propositions hold. If they do, the model checker returns a confirmation. If there is a violation, an execution trace called a counterexample is produced. This counterexample depicts a model state (the value of the model's variables) corresponding to a specification violation along with a list of the incremental model states leading to the violation. Because of its approach, model checking is particularly good at finding problems in systems with concurrency, where independent system elements can interact in ways unanticipated by designers.

The majority of formal verification analyses are concerned with discrete event systems. However, hybrid modeling and analysis techniques can allow formal verification to be used with models that contain continuous quantities [53]–[55]. In such models, a discrete state (such as a particular configuration of sounding alarms) can be associated with continuous quantities (this could include precise times, frequencies, and volumes) that can also be used in the assertion of specifications. For example, to model time formally, formal analysts use timed automata [53], [56], a modeling approach where every discrete transition in a formal model is assigned a real numbered time.

Researchers have used formal verification to evaluate issues related to human–automation interaction (see [57] for a review). These techniques focus on abstract models from the human factors literature that can be represented with discrete mathematical models and used in analyses of a scope such that specific human factors problems can be discovered. Collectively, these studies have shown that formal verification can be very useful for finding problems related to human factors in automated systems. However, none of them have explored how human perception and problems associated with it can be included in these formal analyses.

III. Objective

Because of its ability to detect problems in complex concurrent systems, formal verification should be capable of detecting if masking can manifest in a particular configuration of medical alarms. The work presented here demonstrates that this is possible. We developed a method that allows an analyst to specify a configuration of alarms and use formal verification to detect if there are any situations where each alarm is masked. This method is built around a formal modeling architecture that allows for the sounding behavior of medical alarms to be represented formally. Our framework includes psychoacoustic functions capable of indicating when masking can occur and temporal logic specification property patterns for asserting the absence of masking conditions. Thus, formal verification with model checking can be used to detect if masking exists in models constructed around the framework.

The following section describes the method we developed. This includes an overview of the framework and a detailed description of its components. Then, to demonstrate the utility of the method, we use it to evaluate a realistic medical alarm configuration. Finally, our results are discussed and avenues of future work are explored.

IV. Methods

In the method we have developed (see Fig. 1), an analyst must: 1) examine the documentation associated with a configuration of medical alarms and model their behavior using our formal modeling architecture (see Fig. 2); 2) specify the absence of masking using specification property patterns we provide; and 3) use model checking to formally verify that the specification properties hold for the model. If no masking exists, the model checker will return a confirmation in its verification report. Otherwise, a counterexample will be produced, which will illustrate how masking can occur. This can be used by the analyst to determine how the discovered masking condition might be avoided.

Timing of concurrently sounding alarms can have a profound impact on whether alarms are masked or not; thus, we need to evaluate all of the different ways alarms can temporally overlap. Therefore, we have designed our formal modeling architecture (see Fig. 2) to be based on timed automata. Timed automata [53] provide a means of modeling time as a real-valued continuous quantity in a formal model. This architecture has multiple submodels that are synchronously composed together: a clock (the timed automaton) that keeps and advances time; models of

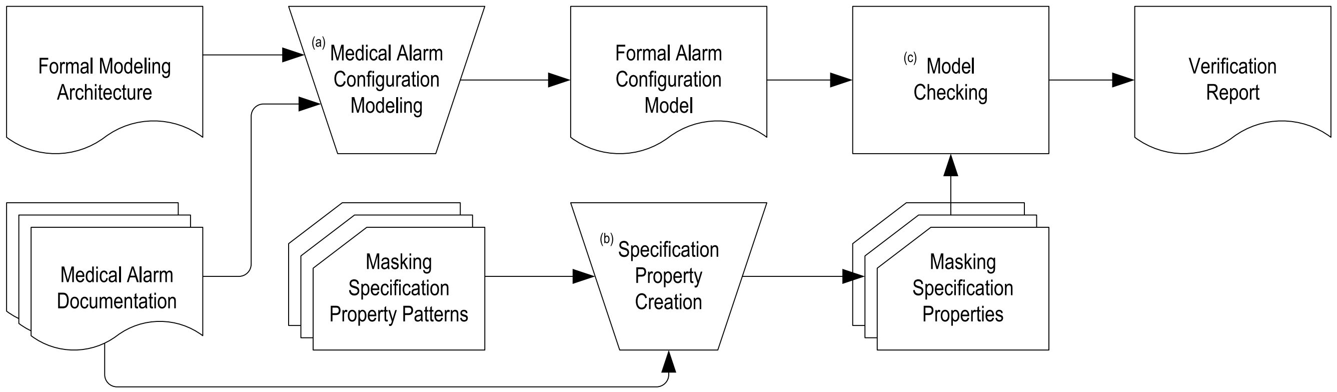
Formal Modeling Architecture
(a) Medical Alarm Configuration Modeling
Formal Alarm Configuration Model
(c) Model Checking
Verification Report
Medical Alarm Documentation
Masking Specification Property Patterns
(b) Specification Property Creation
Masking Specification Properties

Fig. 1. Method for using formal verification to detect auditory masking in medical alarm configurations

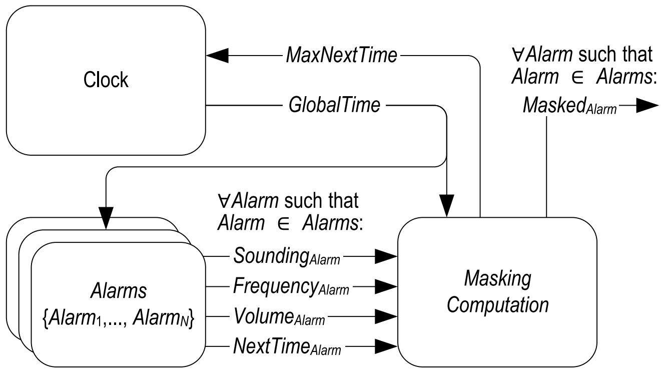

Fig. 2. Architecture for formally modeling a configuration of auditory medical alarms. Boxes represent submodels of the larger system model and arrows represent variables with input–output relationships between the submodels. Arrows with no target indicate outputs.

the behavior of the alarms in a given configuration and a model that computes whether masking is occurring for each alarm and determines the maximum advance of the clock.

We have implemented this method using the tools available in the Symbolic Analysis Laboratory (SAL) [58], [59]. In particular, we have designed our method to work with SAL's infinite bounded model checker [53], [59], a tool capable of evaluating formal models containing timed automata. SAL's infinite bounded model checker uses satisfiability modulo theories to check properties in formal models that contain continuous variables. It is bounded in the sense that it takes a number of steps (the bound) as input. The model checker then proves whether or not the checked specification properties hold for up to the specified number of steps through the model.

Our implementation of the method is designed so that it will require a limited amount of analyst-created code. What is required follows systematic patterns. The remainder of this section describes how our implementation of the method was realized. First, we describe the details of the formal modeling architecture. This is followed by a description of the specification property patterns analysts can use to assert the absence of masking. Finally, we explain how the model checker can be used to evaluate a medical alarm configuration Throughout, we highlight where analyst effort is required.

A. Formal Modeling Architecture

An overview of the SAL implementation of the architecture can be seen in Fig. 3. This has eight distinct parts. First, it contains a collection of type definitions These represent variable types that are used by other elements in the modeling architecture for representing alerting concepts and include nonnegative real-valued time, volume, and frequency.

Next, the model contains two constants that are used to represent standard values used in other parts of the architecture. The first, `delatConst`, represents that constant volume used in the computation of Δ (5). The second, `bigMax`, represents an arbitrarily large maximum on the amount time can increase in a given step through the model.

The constant definitions are followed by function definitions These represent mathematical expressions that are used by other model constructs to compute quantities used in the detection of masking. These are discussed in Section IV-A3.

The clock submodel, which is responsible for maintaining and advancing time, is next. It is described in Section IV-A1.

A series of submodels representing the behavior of the different alarms in the alarm configuration follow. Each of these represents the behavior of a given alarm at the global current time indicated by the clock. Section IV-A2 describes the generic formal modeling pattern used for modeling each alarm in a configuration (with `N` alarms) in the architecture.

The masking computation submodel evaluates the outputs of the alarm submodels and uses the defined functions to compute whether masking is occurring at the given clock-indicated time. This is developed further in Section IV-A3.

Each of the submodels is ultimately synchronously composed into a full system model.

Finally, specification properties are used to assert the absence of masking in a model constructed using the architecture. The generic patterns used for composing such specifications are described in Section IV-B.

Of the architectural components, only the alarm submodels, the masking computation submodel, the system model composition, and the specification properties require any analyst effort. All of the other components are standard.

1) Clock: The clock submodel (see Fig. 4) is responsible for advancing time and communicating the current time to the other

```
formalArchitecture : CONTEXT =
BEGIN

  %Type definitions
  TIME        : TYPE   = {X : REAL | X >= 0}; % in s
  VOLUME      : TYPE   = {X : REAL | X >= 0}; % in dB
  FREQUENCY   : TYPE   = {X : REAL | X >= 0}; % in Hz

  %Constant definitions
  deltaConst  : VOLUME = 14.5;
  bigMax      : TIME   = 60;

  %Function definitions
  ...

  %Clock sub-model
  clock : MODULE =
  BEGIN
    ...
  END;

  %Alarm sub-models
  alarm1 : MODULE =
  BEGIN
    ...
  END;

  ...

  alarmN : MODULE =
  BEGIN
    ...
  END;

  %Masking computation sub-model
  maskingComputation : MODULE =
  BEGIN
    ...
  END;

  %Composition of the full system model
  system : MODULE = clock
                  || alarm1
                  || ...
                  || alarmN
                  || maskingComputation;

  %Specification properties
  ...

END
```

Fig. 3. Overview of the implementation of the formal modeling architecture for modeling medical alarm configuration (see Fig. 2). This implementation is written using the notation of SAL (see [58]). Note that in this listing (and all subsequent listings in Figs. 4–7), code highlighting is used to improve readability: SAL language reserved words (including built-in basic types) are blue; declared types are dark blue; constants are green; functions are orange (these appear in subsequent listing); and everything else is black. Ellipses "..." are used to indicate the omission of content that is either detailed in subsequent listings (see Figs. 4–7) or indicates an incremental series of like components or operations (e.g., the synchronous compositions of the alarm submodels: `alarm1 || ... || alarmN`).

elements of the system. It receives a maximum on the amount that time can advance to (`maxNextTime`) as input and outputs the current and/or global time (`globalTime`). The global time is initially set to 0. Then, for every subsequent step through the model, the global time is advanced to an arbitrary new time that is always greater than the current global time and less than or equal to the maximum next time.

```
clock : MODULE =
BEGIN

  INPUT  maxNextTime : TIME
  OUTPUT globalTime  : TIME

  INITIALIZATION
    globalTime = 0;

  TRANSITION
    globalTime' IN {X: TIME | (X > globalTime)
                    AND (X <= maxNextTime)};

END;
```

Fig. 4. SAL code for representing the clock in the formal model.

2) Alarms: The behavior of each alarm (which is assumed to be a pattern of tones) is described in a separate model, where each alarm model follows a similar implementation pattern (see Fig. 5). Each alarm has a constant value representing the length of its sounding cycle in seconds (`alarmCycleTime` with analyst specified value `[TCycle]` in Fig. 5). Each alarm also has a variable start time (`alarmStartTime`, which is initially 0) that is used to indicate if an alarm is sounding (`alarmSounding = alarmStartTime > 0`) and, if it is, when the alarm started doing so.

The alarm model is responsible for setting the start time and computing the amount of time the alarm has been sounding. Our model assumes that an alarm will sound for a single cycle and then stop (it can restart at any later time). Thus, the amount of time the alarm has been sounding is computed as the difference between the global time and the alarm's start time (`alarmTimeInCycle = globalTime - alarmStartTime`). At any given global time, an alarm that is not sounding can begin sounding in the next state by setting the start time to the global time in the next state (see `TRANSITION` in Fig. 5). If the alarm is sounding and has not been sounding for longer than its cycle time in the next state's global time, the alarm keeps its current start time in the next state. If the alarm has been sounding for its full cycle time at the next global time, the alarm ceases to sound (sets the start time to zero) in the next state.

If the alarm is sounding, then the alarm model must update its frequency (`alarmFrequency`), volume (`alarmVolume`), and next time (`alarmNextTime`) output variables based on the alarm's time in cycle. Specifically, for set times less than or equal to the alarm's cycle time (i.e., `[TFreq1]` – `[TFreqN]` from Fig. 5), the alarm will assume different values for frequency and volume. It should be noted that, in the model shown in Fig. 5, the value of the alarm's volume is determined by the alarm's frequency. However, analysts can have the alarm volume change independently of the alarm's frequency if desired. The purpose of the next time output is to communicate the next global time that the alarm will experience a change in its frequency and/or volume. Thus, the next time variable should update to reflect this based on the current time in cycle.

All alarm models follow the implementation pattern shown in Fig. 5. Within this, the analyst needs to describe the

```
alarm : MODULE =
  BEGIN

  INPUT  globalTime       : TIME

  LOCAL  alarmStartTime   : TIME
  LOCAL  alarmCycleTime   : TIME
  LOCAL  alarmTimeInCycle : TIME

  OUTPUT alarmSounding    : BOOLEAN
  OUTPUT alarmFrequency   : FREQUENCY
  OUTPUT alarmVolume      : VOLUME
  OUTPUT alarmNextTime    : TIME

  INITIALIZATION
    alarmStartTime    =  0;

  DEFINITION
    alarmCycleTime    =  [TCycle];
    alarmTimeInCycle =  globalTime - alarmStartTime;
    alarmSounding     =  alarmStartTime > 0;
    alarmFrequency    =  IF    alarmSounding AND alarmTimeInCycle < [TFreq1] THEN [Freq1]
                         ...
                         ELSIF alarmSounding AND alarmTimeInCycle < [TFreqN] THEN [FreqN]
                         ELSE                                                      0
                         ENDIF;
    alarmVolume       =  IF    alarmSounding AND alarmFrequency = [Freq1] THEN [Vol1]
                         ...
                         ELSIF alarmSounding AND alarmFrequency = [FreqN] THEN [VolN]
                         ELSE                                                   0
                         ENDIF;
    alarmNextTime     =  IF    NOT alarmSounding           THEN bigMax
                         ELSIF alarmTimeInCycle < [TFreq1] THEN alarmStartTime + [TFreq1]
                         ...
                         ELSIF alarmTimeInCycle < [TFreqN] THEN alarmStartTime + [TFreqN]
                         ELSE                                   alarmStartTime + alarmCycleTime
                         ENDIF;

  TRANSITION
    alarmStartTime' IN {X: TIME | ((alarmStartTime = 0) AND ((X = globalTime') OR (X = 0)))
                        OR ((alarmStartTime > 0)
                            AND (globalTime' < (alarmStartTime + alarmCycleTime))
                            AND (X = alarmStartTime))
                        OR ((alarmStartTime > 0)
                            AND (globalTime' >= (alarmStartTime + alarmCycleTime))
                            AND (X = 0))
                       };
END;
```

Fig. 5. Generic SAL code for representing alarm behavior. Note that bracketed words in red represent numerical values that should be explicitly specifie by the analyst. `[TCycle]` represents the alarm's cycle time in seconds. `[TFreq1]` – `[TFreqN]` represent relative times (from the start time) that the frequency and volume change in increasing order in seconds. `[Freq1]` – `[FreqN]` and `[Vol1]` – `[VolN]` represent different frequencies (in Hz) and volumes (in dB), respectively.

```
bark(f: FREQUENCY): REAL
  = IF f <= 5    THEN 0    ELSIF  f <= 15    THEN 0.1  ... ELSIF  f <= 15145 THEN 23.9 ELSE 24   ENDIF;

spread(dz: REAL): REAL
  = IF dz <= -24 THEN -572 ELSIF dz <= -23.9 THEN -570 ... ELSIF dz <= 23.9  THEN -228 ELSE -229 ENDIF;

masking(sMasker: BOOLEAN, fMasker: FREQUENCY, vMasker: VOLUME,
        sMaskee: BOOLEAN, fMaskee: FREQUENCY, vMaskee: VOLUME): BOOLEAN
  = IF    NOT(sMasker AND sMaskee) THEN FALSE
    ELSIF vMaskee = 0              THEN TRUE
    ELSE  ((spread(bark(fMaskee) - bark(fMasker)) + vMasker - (deltaConst + bark(fMasker))) >= vMaskee)
    ENDIF;
```

Fig. 6. Method implementation of (1)–(5) used by the `maskingComputation` submodel to compute whether masking is occurring between two alarms.

```
maskingComputation : MODULE =
BEGIN

  INPUT  globalTime      : TIME
  INPUT  alarm1Volume    : VOLUME
  INPUT  alarm1Frequency : FREQUENCY
  INPUT  alarm1Sounding  : BOOLEAN
  INPUT  alarm1NextTime  : TIME
  ...
  INPUT  alarmNVolume    : VOLUME
  INPUT  alarmNFrequency : FREQUENCY
  INPUT  alarmNSounding  : BOOLEAN
  INPUT  alarmNNextTime  : TIME

  OUTPUT alarm1Masked    : BOOLEAN
  ...
  OUTPUT alarmNMasked    : BOOLEAN

  OUTPUT maxNextTime     : TIME

  DEFINITION
    alarm1Masked    =  masking(alarm2Sounding,   alarm2Frequency,   alarm2Volume,
                               alarm1Sounding,   alarm1Frequency,   alarm1Volume)
                       OR ...
                       OR masking(alarmNSounding,   alarmNFrequency,   alarmNVolume,
                                  alarm1Sounding,   alarm1Frequency,   alarm1Volume);
    ...
    alarmNMasked    =  masking(alarm1Sounding,   alarm1Frequency,   alarm1Volume,
                               alarmNSounding,   alarmNFrequency,   alarmNVolume)
                       OR ...
                       OR masking(alarmN-1Sounding, alarmN-1Frequency, alarmN-1Volume,
                                  alarmNSounding,   alarmNFrequency,   alarmNVolume);
    maxNextTime     IN {X: TIME | (X = alarm1NextTime OR ... OR X = alarmNNextTime)
                        AND (X <= alarm1NextTime AND ... AND X <= alarmNNextTime)};

END;
```

Fig. 7. Generic SAL code for the masking computation submodel.

`alarmCycleTime` (the `[TCycle]` value in Fig. 5) and the logic definin the `alarmFrequency`, `alarmVolume`, and `alarmNextTime` by specifying the appropriate values (`[TFreq1]`, `[Freq1]`, `[Vol1]`, etc.).

3) Masking Computation: The masking computation model (see Fig. 7) has two roles. First, at every time assumed by the clock, it looks at the frequency and volume of each alarm and computes whether it is being masked by the other sounding alarms. These computations are synthesized into a single Boolean variable for each alarm that indicates if that alarm is being masked (`alarm1Masked` – `alarmNMasked`). They are performed using a set of functions (see Fig. 6) that implement the equations in (1)–(5) in the formal model. Because model checkers are limited in their ability to consider nonlinear variable arithmetic in their formal input models, two of these functions were implemented using lookup tables: values of the functions over a range of acceptable values are precomputed and accessed in the formal model using a large `IF...THEN...ELSIF` statement. For the conversion of frequency (in hertz) to the Bark scale, the lookup table was computed using (1) rounded to the nearest tenth of a Bark for the full range of Bark scale values (0–24). For the spreading function (`spread`), (4) was used to compute the spread rounded up the nearest dB for the full range of possible values for δz (`dz` in Fig. 6). Note that this computation rounds up so that it biases the masking curve slightly in favor of detection. Further, the spreading function from (4) was chosen because it uses only δz in its computation, making the corresponding lookup table 1-D and thus much simpler to implement formally. This is discussed in more detail in Section IV-A.

The masking function (`masking`) uses both the `bark` and `spread` functions to compute whether or not a given alarm (the masker) masks another alarm (the maskee). `sMasker`, `fMasker`, `vMasker`, `sMaskee`, `fMaskee`, `vMaskee` represent the following concepts for masker and maskee alarms respectively: whether or not the alarm is sounding; its frequency (in hertz); and volume (in dB). If neither the masker nor the maskee are sounding, no masking is possible. If this is not true and the volume of the maskee is 0 (for example, when the alarm is sounding but in a pause between alarm tones), then the maskee is being masked. Otherwise, the function computes whether the masker masks the maskee using (6).

The second thing the masking computation model does is calculate the next time variables ($NextTime_{Alarm}$) from all of the alarms and communicates it to the clock as `maxNextTime`. It does this by selecting a time value from a set of times that are equal to at least one of the alarm's next times and less than or equal to all of the alarm next times.

In creating the masking computation model, an analyst is responsible for creating the "`alarmMasked`" variable and its

```
alarmPartialMasking : THEOREM system
  |- G(NOT(alarmVolume > 0 AND alarmMasked));
alarmTotalMasking   : THEOREM system
  |- G(NOT((NOT alarmSounding)
       AND X(alarmSounding AND alarmMasked
            AND U(alarmSounding AND alarmMasked,
                 NOT alarmSounding))));
```

Fig. 8. Specification property patterns for a given alarm. alarmPartialMasking asserts the absence of masking for a given alarm. alarmPartialMasking asserts that a given alarm will never be completely masked.

definition for each alarm using the pattern shown in Fig. 7. A model must also create the maxNextTime definition again using the pattern in Fig. 7.

B. Specification

To model check whether or not masking is present in a model, specification must assert its absence. Our method uses property patterns to do this, where an analyst must instantiate the specification pattern for each alarm in a configuration In this study, we are interested in detecting whether, for a given alarm, there is any situation where the alarm is masked and if the alarm can be totally masked (completely imperceptible). Thus, specification property patterns are created for each alarm asserting the absence of both phenomena (see Fig. 8). For a given alarm (alarm), alarmPartialMasking uses linear temporal logic to assert that: For all paths through the model (G), there should never be a situation where the alarm is making noise (alarmVolume > 0) and the alarm is masked. alarmTotalMasking asserts that: For all (G) paths through the model, we never want it to be true that the alarm goes from not sounding to sounding and masked in the next (X) state such that, from then on, the alarm is sounding and masked until (U) it is no longer sounding.

When creating the specifications an analyst needs to replicate the specification property patterns (see Fig. 8) for each alarm.

V. Case Study

To illustrate the ability of our method to detect masking in a realistic configuration of medical alarms, we have used it to evaluate a simple case study. In this target configuration (see Table I), there were three alarms. In a given cycle, each alarm played a two-tone pattern with a pause in between. Each sound used a frequency commonly found in tonal alarms [5], [32]. Durations and volumes were also consistent with the IEC 60601-1-8 international standard [32].

These alarms were used to construct four separate formal models using the above implementation of the method. Three of these models contained the implementation of each pair of alarms from Table I. The last contained the implementation of all three alarms. Each model was implemented by instantiating the formal model architecture discussed in Section IV-A. This entailed creating an alarm module for each alarm included in the configuration using the pattern in Fig. 5. This meant explicitly including the bracketed values in the pattern based on the desired alarm behavior. For the three alarms in Table I (for which N = 3), these are shown in Table II.

TABLE I
ALARM CONFIGURATION PROFILES

Alarm	Tone 1 Freq. (Hz)	Vol. (dB)	Time (s)	Pause Time (s)	Tone 2 Freq. (Hz)	Vol. (dB)	Time (s)
Alarm 1	261	80	0.25	0.100	370	80	0.25
Alarm 2	277	60	0.15	0.050	277	60	0.15
Alarm 3	524	85	0.20	0.075	294	85	0.20

TABLE II
VALUES FOR IMPLEMENTING THE ALARMS IN TABLE I USING THE PATTERN IN FIG. 5

Value	Alarm 1	Alarm 2	Alarm 3
[TCycle]	$0.25 + 0.1 + 0.25 = 0.6$	$0.15 + 0.05 + 0.15 = 0.35$	$0.2 + 0.075 + 0.2 = 0.475$
[TFreq1]	0.25	0.15	0.2
[Freq1]	261	277	524
[Vol1]	80	60	85
[TFreq2]	$0.25 + 0.1 = 0.35$	$0.15 + 0.05 = 0.2$	$0.2 + 0.075 = 0.275$
[Freq2]	0	0	0
[Vol2]	0	0	0
[TFreq3]	$0.25 + 0.1 + 0.25 = 0.6$	$0.15 + 0.05 + 0.15 = 0.35$	$0.2 + 0.075 + 0.2 = 0.475$
[Freq3]	370	277	294
[Vol3]	80	60	85

In all of the models, specification were created using the patterns from Fig. 8 to assert that each of the included alarms should never be partially or totally masked (see http://fhsl.eng.buffalo.edu/resources/ for full listings of all models). By evaluating each of these models separately, we are able to test the ability of the method to detect masking within the possible interactions of any two alarms as well as all of the alarms together.

Every specification in each model was evaluated using SAL's infinite bounded model checker [59] (with search depth 12) on a Linux workstation with a 3.3-GHz Intel Xeon processor and 64 GB of RAM.

All verification results can be seen in Table III. This shows that no masking was detected when only Alarm 1 and Alarm 3 were in the model. However, in the model where Alarm 1 and Alarm 2 were present and the model where Alarm 2 and Alarm 3 were present, partial masking was detected but not total masking. In the model with all three of the alarms, both partial and total masking were detected. The counterexamples returned by the model checker for each specification (which can be seen at http://fhsl.eng.buffalo.edu/resources/) were visualized (see Fig. 9) to determine exactly how the detected masking manifested. This revealed that the first tone of Alarm 1 and the second tone of Alarm 3 were both capable of masking the tones of Alarm 2.

VI. Discussion

This study has introduced a novel method for identifying masking in configuration of medical alarms. This method uses

TABLE III
VERIFICATION RESULTS

Model with			Verification Output	
Alarms	Alarm	Specification	Time (s)	Outcome
1 & 2	Alarm 1	Partial Masking	87.26	✓
		Total Masking	63.76	✓
	Alarm 2	Partial Masking	99.31	×
		Total Masking	56.11	✓
1 & 3	Alarm 1	Partial Masking	67.05	✓
		Total Masking	32.88	✓
	Alarm 3	Partial Masking	66.37	✓
		Total Masking	127.57	✓
2 & 3	Alarm 2	Partial Masking	180.56	×
		Total Masking	95.82	✓
	Alarm 3	Partial Masking	85.49	✓
		Total Masking	69.52	✓
1, 2, & 3	Alarm 1	Partial Masking	392.76	✓
		Total Masking	320.62	✓
	Alarm 2	Partial Masking	1281.26	×
		Total Masking	492.76	×
	Alarm 3	Partial Masking	297.92	✓
		Total Masking	1205.43	✓

Note. ✓ indicates a verification confirmation and × indicates a verification failure with a counterexample.

a formal modeling architecture, psychoacoustic models of masking, specification property patterns, and formal verification with model checking to prove whether or not each alarm in a modeled configuration will be perceptible with normal hearing. We have implemented a version of this method in SAL using timed automata. To demonstrate the method's power, we presented a realistic medical alarm configuration and showed how our method could be used to find masking conditions. The power of the method is particularly well illustrated by the multiple verifications that were performed. While partial masking was detected in models that only contained two alarms, total masking was only observed when the interactions between all three alarms were considered simultaneously.

As the number of alarms in medical environments increases and causes more and more alarm interactions, there will be even more chances for total alarm masking conditions. As such, the presented method could be used by hospital personnel to evaluate the safety of different medical alarm configurations by considering all of the possible alarm interactions. Thus, this study has the potential to significantly improve patient safety. Further, our method is a contribution because it represents the first successful attempt to model psychophysical concepts in a formal model. However, despite its success, this method has some limitations, which will be addressed in future work.

A. Additional Masking Considerations

Our current implementation of the method uses the spreading function from the MPEG2 audio codec for normal hearing (4). This particular spreading function was chosen because it is 1-D: only varied as a function of δz. This made it easier to include the spreading function calculations in the formal model. However, a number of different spreading functions have been developed, all with slightly different shapes and thus different tradeoffs between miss and false alarm rates for detection [9]. An alarm that is not totally masked but very close to a masking threshold will still be difficult for an operator to perceive. Thus, for our method, there is greater utility in creating a liberally biased masking detector (one that errs toward false alarms) rather than a conservative one (one that errs toward misses). Future work should investigate which spreading function provides the best desired detection behavior and integrate it into our method.

Currently, the method is only set up to detect simultaneous masking for individuals with normal hearing. However, this may be an unrealistic expectation for everybody that may need to perceive different medical alarms. Thus, future work should investigate how different spreading functions could be used to account for individuals with different hearing proficiencies

Another limitation of the current approach is that it does not account for additive masking. Additive masking describes a condition where two simultaneous sounds can produce masking greater than or equal to the sum of their respective masking curves [9], [60]. Specifically, additive masking is computed as

$$I_N = \left(\sum_{n=1}^{N} I_n^{\alpha} \right)^{1/\alpha} \tag{7}$$

where I_N represents the combined masking intensity of N maskers at a given frequency, I_n represents the masking intensity of masker n (its masking curve value) at that frequency, and α is a constant with range $(0, \infty)$. With an α of 1, masking effects are purely additive. However, Lutfi [61] found that an α of 0.33 was best suited to tonal sounds, a condition that leads to the "over adding" of different sounds' masking effects. For medical alarms, additive masking can manifest in two different contexts: either multiple alarms can be sounding and interact to create an aggregate masking curve and/or the alarm may contain multiple prominent auditory harmonics which can contribute to a given alarm's masking [32]. Temporal masking can also occur [4], where the temporal relationships between sounds can mask those not concurrently sounding. Given that these phenomena suggest many additional alarm interactions that can cause masking beyond those considered in the current implementation, future work should investigate how to include them in our method.

It is also uncommon for alarms to be operating in a completely quiet environment. Thus, alarms may interact with environmental noise in ways that could exacerbate masking conditions. Future work should investigate how other environmental sounds could be incorporated into our method.

Many alarms are periodic in that they will sound repeatedly until a problem is dealt with. However, while our current method does allow alarms to sound over multiple cycles, it only checks whether an alarm is partially or completely masked in any given single cycle. It is our contention that even seconds of delay in operator response could have profound impacts on human health in medical environments. Thus, even the imperceptibility of a single cycle is dangerous and thus indicative of a problem that needs to be addressed. However, it may not always be possible to completely eliminate masking. In these situations, analysts may wish to determine if alarms will be perceivable within a

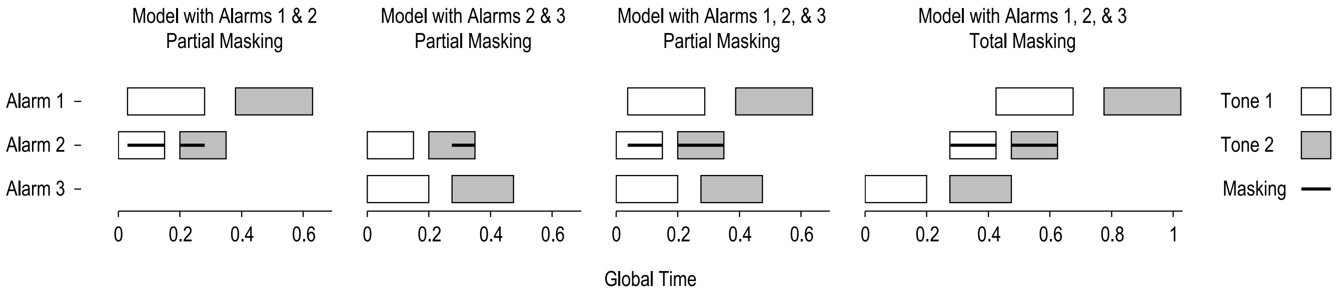

Fig. 9. Illustration of the counterexamples returned when the model checker failed to prove that Alarm 2 would not be partially or completely masked. For the partial masking results, Alarm 2's tones were masked by Alarm 1's first tone and Alarm 2's second tone was masked by Alarm 3's second tone. For the total masking result, the second tone of Alarm 3 masks the first tone of Alarm 2 and Alarm 2's second tone was masked by Alarm 1's first tone.

particular amount of time. To do such an analysis in the current method, an analyst would need to incorporate many repeating alarm cycles into a single modeled cycle so that the alarm would sound continuously over a desired interval of time. In doing this, the analyst would also need to increase the depth used during the model checking process to ensure that the model checker could search over all possible alarm interactions in the expanded cycle time. Future extensions of the method should investigate how to make such modeling easier and allow analysts to reason about masking in terms of detection time.

B. Experimental Validation

Our method is based on established psychological principles and is thus expected to give accurate predictions. However, to show the validity of the method, it would be good to validate its predictions against actual human subject experimental results in realistic operational environments. Future work should pursue this. Additionally, human subject experiments could help us choose and tune spreading functions to achieve the desired prediction results. This should also be explored in future work.

C. Additional Case Studies and Use in Design

While the presented case study illustrates the method's utility, there are many medical alarm configurations [62]. Further, standards such as IEC 60601-1-8 [32] have a number of open parameters that can be used to represent the behavior of alarms. Slight modifications to our method could allow us to search this parameter space to determine if they allow for the existence of masking. Future work will investigate this.

To date, the method has only been used to detect masking, not prevent it. However, through iterative modeling and verification, an analyst could use the method to find alarm settings or subsets of alarms that would not produce masking. However, it is also conceivable that an analyst will encounter situations where he or she must use an alarm configuration that produces masking. Even in this situation, the presented method should have utility as it could allow the analyst to identify interventions (such as alarm positioning to target different listener ears and thus support localization) that could improve the chances of alarm perceptibility. However, it is not clear how feasible such a solution would be in a dynamic healthcare environment. Future work should investigate how the method could be used to provide analysts with realistic decision support for alarm configuration design, selection, and positioning.

Finally, alarms are critical to safety in domains beyond medicine including automotives, aviation, and industrial settings. Future work should explore how our method could be used to detect masking in these environments.

D. Scalability

All model checking-based verifications have scalability problems: where the size of the model grows exponentially as concurrent elements are included. This can quickly lead to models that are too big or take too long to verify due to computational and/or physical memory limitations [51]. The case study presented here took slightly more than 21 min to verify. Thus, for more complex case studies, it is likely that scalability problems will be constraining. Such problems will likely be exacerbated as more complex masking conditions (i.e., additive or temporal masking) are included. However, there are potential opportunities for improving the scalability of our approach. In particular, some of the computations the formal model performs could be done using preprocessing. For example, instead of dynamically calculating alarm masking curves for each alarm in a given configuration, these could be precalculated so that the associated lookup tables would be optimized for the configuration. Future work will investigate how to incorporate this and other scalability improvements into our method.

Even if scalability proves to be a persistent problem with this approach, the method may still have utility. Specifically, rather than evaluating dynamically created alarm configurations for immediate deployment in a healthcare environment, the method could be used to preassemble a database of alarm configurations across medical devices certified to avoid masking. Such analyses would not need to be done dynamically in the field and could be done without temporal constraints and on more sophisticated hardware. Thus, if scalability proves to be a significant constraint on the utility of our method, we will pursue this certification approach.

E. Other Alarm Considerations

As mentioned in Section I, there are many other problems facing medical alarms. As this is the first attempt to explore alarm problems formally, there may be many future opportunities for extending our work to explore other alarm issues. For example, there is good evidence suggesting that human mental workload can contribute to alarm mistrust, fatigue, and inattentional deafness [20], [23]. Formal methods could help researchers discover when these conditions could occur. Such an analysis will need to integrate formal approaches for modeling alarm perception, workload [63], and task behavior [64]–[66] to be successful. This should be explored in future work. Additionally, excessive false alarms can cause people to intentionally ignore alarms thought to be spurious (the "cry wolf" phenomena [67]). By reducing the number of false alarms, masking potential could inherently be reduced. Future work should explore how formal methods could be used to help reduce false alarms in addition to the masking analyses discussed here.

F. Tool Usability

Finally, the method as currently implemented requires that the analyst be familiar with and implement formal models of alarm behavior. While our formal modeling architecture can assist analysts in this task, it will be cumbersome for those with no formal modeling experience. Ideally, our method would allow analysts to easily explore different alarm configuration with little to no knowledge of formal techniques. Future work should investigate how to develop tools that will enable analysts to quickly construct and analyze alarm configuration without the need for manual formal modeling and specification

Acknowledgment

The authors would like to thank A. House and K. Molinaro for their help proofreading this manuscript.

References

[1] B. Hasanain, A. Boyd, and M. L. Bolton, "An approach to model checking the perceptual interactions of medical alarms," in *Proc. 2014 Int. Annu. Meeting Human Factors Ergonomics Soc*, 2014, pp. 822–826.

[2] J. Edworthy, "Designing effective alarm sounds," *Biomed. Instrum. Technol.*, vol. 45, no. 4, pp. 290–294, 2011.

[3] P. Lacherez, E. Seah, and P. Sanderson, "Overlapping melodic alarms are almost indiscriminable," *Human Factors*, vol. 49, no. 4, pp. 637–645, 2007.

[4] H. Fastl and E. Zwicker, *Psychoacoustics: Facts and Models*, vol. 22. New York, NY, USA: Springer, 2006.

[5] K. Momtahan, R. Hetu, and B. Tansley, "Audibility and identificatio of auditory alarms in the operating room and intensive care unit," *Ergonomics*, vol. 36, no. 10, pp. 1159–1176, 1993.

[6] O. Toor, T. Ryan, and M. Richard, "Auditory masking potential of common operating room sounds: A psychoacoustic analysis," in *Proc. Anesthesiology 2008*, vol. 109, 2008, p. A1207.

[7] J. Edworthy, "Medical audible alarms: A review," *J. Amer. Med. Informat. Assoc.*, vol. 20, no. 3, pp. 584–589, 2013.

[8] E. M. Clarke and J. M. Wing, "Formal methods: State of the art and future directions," *ACM Comput. Surveys*, vol. 28, no. 4, pp. 626–643, 1996.

[9] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards*. New York, NY, USA: Springer, 2003.

[10] R. P. Carlyon and S. McAdams, "The psychophysics of concurrent sound segregation," *Philosophical Trans. Royal Soc. London. Ser. B, Biol. Sci.*, vol. 336, no. 1278, pp. 347–355, 1992.

[11] ECRI Institute, "Top 10 health technology hazards for 2013," *Health Devices*, vol. 41, no. 11, pp. 1–23, 2012.

[12] W. W. Stead and H. S. Lin, Eds., *Computational Technology for Effective Health Care: Immediate Steps and Strategic Directions*. Atlanta, GA, USA: National Academies Press, 2009.

[13] ECRI Institute and ISMP. (2009). Connecting remote cardiac monitoring issues with care areas. *Pennsylvania Patient Safety Authority. 6(3)*, pp. 79–83. [Online]. Available: http://patientsafetyauthority.org/ADVISORIES/AdvisoryLibrary/2009/Sep6(3)/Pages/79.aspx

[14] ECRI Institute. (2013, May 27). "Alarm safety resource site: Guidance and tools to help healthcare facilities improve alarm safety," [Online]. Available: https://www.ecri.org/Forms/Pages/Alarm_Safety_Resource.aspx

[15] The Joint Commission, "Medical device alarm safety in hospitals," *Sentinel Even Alert*, vol. 50, pp. 1–3, Apr. 8, 2013.

[16] E. M. J. Koski, A. Mäkivirta, T. Sukuvaara, and A. Kari, "Frequency and reliability of alarms in the monitoring of cardiac postoperative patients," *J. Clin. Monitoring Comput.*, vol. 7, no. 2, pp. 129–133, 1991.

[17] S. T. Lawless, "Crying wolf: False alarms in a pediatric intensive care unit," *Critical Care Med.*, vol. 22, no. 6, pp. 981–985, 1994.

[18] S. Siebig, S. Kuhls, M. Imhoff, U. Gather, J. Schölmerich, and C. E. Wrede, "Intensive care unit alarms-how many do we need?" *Critical Care Med.*, vol. 38, no. 2, pp. 451–456, 2010.

[19] J. P. Bliss and S. A. Acton, "Alarm mistrust in automobiles: How collision alarm reliability affects driving," *Appl. Ergonomics*, vol. 34, no. 6, pp. 499–509, 2003.

[20] J. P. Bliss and M. C. Dunn, "Behavioural implications of alarm mistrust as a function of task workload," *Ergonomics*, vol. 43, no. 9, pp. 1283–1300, 2000.

[21] J. P. Bliss, R. Gilson, and J. Deaton, "Human probability matching behaviour in response to alarms of varying reliability," *Ergonomics*, vol. 38, no. 11, pp. 2300–2312, 1995.

[22] J. P. Keller, R. Diefes, K. Graham, M. Meyers, and K. Pelczarski, "Why clinical alarms are a 'top ten' hazard how you can help reduce the risk," *Biomed. Instrum. Technol.*, vol. Suppl, pp. 17–23, 2011.

[23] F. Dehais, M. Causse, F. Vachon, N. Régis, E. Menant, and S. Tremblay, "Failure to detect critical auditory alerts in the cockpit evidence for inattentional deafness," *Human Factors*, vol. 56, no. 4, pp. 631–644, 2014.

[24] R. D. Patterson, *Guidelines for Auditory Warning Systems on Civil Aircraft*. London, U.K.: Civil Aviation Authority, 1982.

[25] J. Edworthy, "The design and implementation of non-verbal auditory warnings," *Appl. Ergonomics*, vol. 25, no. 4, pp. 202–210, 1994.

[26] P. Sanderson, A. Wee, and P. Lacherez, "Learnability and discriminability of melodic medical equipment alarms," *Anaesthesia*, vol. 61, no. 2, pp. 142–147, 2006.

[27] A. N. Wee and P. M. Sanderson, "Are melodic medical equipment alarms easily learned?" *Anesthesia Analgesia*, vol. 106, no. 2, pp. 501–508, 2008.

[28] F. E. Block, "Evaluation of users' abilities to recognize musical alarm tones," *J. Clin. Monitoring Comput.*, vol. 8, no. 4, pp. 285–290, 1992.

[29] C. Laroche, H. T. Quoc, R. Hétu, and S. McDuff, "'Detectsound': A computerized model for predicting the detectability of warning signals in noisy workplaces," *Appl. Acoust.*, vol. 32, no. 3, pp. 193–214, 1991.

[30] B. C. J. Moore, "Masking in the human auditory system," in *Proc. Audio Eng. Soc. Conf.: Collected Papers Digital Audio Bit-Rate Reduction*, 1996, pp. 9–19.

[31] RSSB. (2012, May 23). Alarms and alerts guidance and evaluation tool. [Online]. Available: http://www.rssb.co.uk/sitecollectiondocuments/pdf/research-toolkits/T32 6/main.html

[32] *Medical Electrical Equipment—Part 1-8*, IEC 60601-1-8, Aug. 14, 2003.

[33] C. Meredith and J. Edworthy, "Are there too many alarms in the intensive care unit? An overview of the problems," *J. Adv. Nursing*, vol. 21, no. 1, pp. 15–20, 1995.

[34] A. Konkani, B. Oakley, and T. J. Bauld, "Reducing hospital noise: A review of medical device alarm management," *Biomed. Instrum. Technol.*, vol. 46, no. 6, pp. 478–487, 2012.

[35] J. Edworthy and E. Hellier, "Alarms and human behaviour: Implications for medical alarms," *Brit. J. Anaesthesia*, vol. 97, no. 1, pp. 12–17, 2006.

[36] J. Edworthy and E. Hellier, "Fewer but better auditory alarms will improve patient safety," *Quality Safety Health Care*, vol. 14, no. 3, pp. 212–215, 2005.

[37] R. D. Patterson, T. F. Mayfield R. D. Patterson, and T. F. Mayfield "Auditory warning sounds in the work environment," *Philosophical Trans. Royal Soc. London B, Biol. Sci.*, vol. 327, no. 1241, pp. 485–492, 1990.

[38] L. G. Huettel and L. M. Collins, "Using computational auditory models to predict simultaneous masking data: Model comparison," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 12, pp. 1432–1440, Dec. 1999.

[39] L. E. Humes and W. Jesteadt, "Models of the additivity of masking," *J. Acoust. Soc. Amer.*, vol. 85, no. 3, pp. 1285–1294, 1989.

[40] A. J. Oxenham and B. C. J. Moore, "Additivity of masking in normally hearing and hearing-impaired subjects," *J. Acoust. Soc. Amer.*, vol. 98, pp. 1921–1934, 1995.

[41] A. J. Oxenham and B. C. J. Moore, “Modeling the additivity of nonsimultaneous masking,” *Hearing Res.*, vol. 80, no. 1, pp. 105–118, 1994.

[42] F. Baumgarte, C. Ferekidis, and H. Fuchs, “A nonlinear psychoacoustic model applied to ISO/MPEG layer 3 coder,” in *Proc. 99th convention Audio Eng. Soc.*, Oct. 1995, preprint 4087.

[43] M. R. Schroeder, B. S. Atal, and J. Hall, “Optimizing digital speech coders by exploiting masking properties of the human ear,” *J. Acoust. Soc. Amer.*, vol. 66, pp. 1647–1652, 1979.

[44] E. Ambikairajah, A. Davis, and W. Wong, “Auditory masking and MPEG-1 audio compression,” *Electron. Commun. Eng. J.*, vol. 9, no. 4, pp. 165–175, 1997.

[45] K. Brandenburg and G. Stoll, “ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio,” *J. Audio Eng. Soc.*, vol. 42, no. 10, pp. 780–792, 1994.

[46] K. Brandenburg and M. Bosi, “Overview of MPEG audio: Current and future standards for low bit-rate audio coding,” *J. Audio Eng. Soc.*, vol. 45, no. 1/2, pp. 4–21, 1997.

[47] E. Zwicker and R. Feldtkeller, *Das Ohr als Nachrichtenempfnger*. Stuttgart, Germany: Hirzel Verlag, 1967.

[48] E. Zwicker and E. Terhardt, “Analytical expressions for critical-band rate and critical bandwidth as a function of frequency,” *J. Acoust. Soc. Amer.*, vol. 68, no. 5, pp. 1523–1525, 1980.

[49] N. Jayant, J. Johnston, and R. Safranek, “Signal compression based on models of human perception,” *Proc. IEEE*, vol. 81, no. 10, pp. 1385–1422, Oct. 1993.

[50] J. M. Wing, “A specifier s introduction to formal methods,” *Computer*, vol. 23, no. 9, pp. 8, 10–22, 24, 1990.

[51] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.

[52] E. A. Emerson, “Temporal and modal logic,” in *Handbook of Theoretical Computer Science*, J. van Leeuwen, A. R. Meyer, M. Nivat, M. Paterson, and D. Perrin, Eds., Cambridge, MA, USA: MIT Press, 1990, ch. 16, pp. 995–1072.

[53] B. Dutertre and M. Sorea, “Timed systems in SAL,” SRI Int., Menlo Park, CA, USA, Tech. Rep. NASA/CR-2002-211858, 2004.

[54] T. A. Henzinger, “The theory of hybrid automata,” in *Proc. 11th Annu. IEEE Symp. Logic Comput. Sci.*, 1996, pp. 278–292.

[55] A. Podelski and S. Wagner, “Model checking of hybrid systems: From reachability towards stability,” in *Hybrid Systems: Computation and Control*. New York, NY, USA: Springer, 2006, pp. 507–521.

[56] R. Alur and D. L. Dill, “A theory of timed automata,” *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994.

[57] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu, “Using formal verificatio to evaluate human-automation interaction in safety critical systems: A review,” *IEEE Trans. Syst., Man Cybern.: Syst.*, vol. 43, no. 3, pp. 488–503, May 2013.

[58] L. de Moura, S. Owre, and N. Shankar, “The SAL language manual,” Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Tech. Rep. CSL-01-01, 2003.

[59] L. de Moura, S. Owre, H. Rue, J. Rushby, N. Shankar, M. Sorea, and A. Tiwari, “SAL 2,” in *Computer Aided Verification* (ser. Lecture Notes in Computer Science), vol. 3114. Berlin, Germany: Springer, 2004, pp. 496–500.

[60] D. M. Green, “Additivity of masking,” *J. Acoust. Soc. Amer.*, vol. 41, no. 6, pp. 1517–1525, 1967.

[61] R. A. Lutfi “Additivity of simultaneous masking,” *J. Acoust. Soc. Amer.*, vol. 73, no. 1, pp. 262–267, 1983.

[62] A. D. Boyd, “Centralized telemetry monitoring center human factors report,” Univ. Illinois, Chicago, IL, USA, Tech. Rep., 2010.

[63] J. Moore, R. Ivie, T. Gledhill, E. Mercer, and M. Goodrich, “Modeling human workload in unmanned aerial systems,” in *Proc. 2014 AAAI Spring Symp. Series*, pp. 44–49, 2014.

[64] M. L. Bolton, R. I. Siminiceanu, and E. J. Bass, “A systematic approach to model checking human-automation interaction using task-analytic models,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 5, pp. 961–976, Sep. 2011.

[65] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu, “Generating phenotypical erroneous human behavior to evaluate human-automation interaction using model checking,” *Int. J. Human-Comput. Stud.*, vol. 70, no. 11, pp. 888–906, 2012.

[66] M. L. Bolton and E. J. Bass, “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.*, vol. 43, no. 6, pp. 1314–1327, Nov. 2013.

[67] S. Breznitz, *Cry Wolf: The Psychology of False Alarms*. London, U.K.: Lawrence Erlbaum, 1984.

Bassam Hasanain received the B.S. degree in electrical engineering from the University of Toledo, Toledo, OH, USA, in 2008, and the M.S. degree in engineering management from the University of Dayton, Dayton, OH, in 2009. He is currently working toward the Ph.D. degree with the Department of Industrial Engineering, University of Illinois at Chicago, Chicago, IL, USA.

His research interests include model checking, formal methods, psychoacoustics, and psychophysics, in particular model checking applied to medical alarms.

Andrew D. Boyd received the B.S. degree in chemistry and mathematics for Southern Methodist University, Dallas, TX, USA, and the M.D. degree from the University of Texas Southwestern Medical School, Dallas.

He is currently an Assistant Professor with the Department of Biomedical and Health Information Sciences, University of Illinois at Chicago, Chicago, IL, USA. His research interests include data simplificatio to improve clinical outcomes in cardiology and oncology.

Matthew L. Bolton (S’05–M’10) received the B.S. degree in computer science in 2003, the M.S. degree in systems engineering in 2006, and the Ph.D. degree in systems engineering in 2010, all from the University of Virginia, Charlottesville, VA, USA.

He is currently an Assistant Professor with the Department of Industrial and Systems Engineering, University at Buffalo, State University of New York, Amherst, NY, USA. His research interests include the development of tools and techniques for using human performance modeling, task analysis, and formal methods to analyze, design, and evaluate complex safety-critical systems.