MÉTODO DE MODELAGEM E VERIFICAÇÃO FORMAL APLICADO A SISTEMAS DE CONTROLE DE TRÁFEGO AÉREO

Rafael Leme Costa*, Fabio Seiti Aguchiku*, Douglas D.S. Santana*, Newton Maruyama*

*Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da Escola Politécnica da USP
Av. Prof. Mello Moraes, 2231, Cidade Universitária
São Paulo, SP, Brasil

Emails: lemerafael@usp.br, fabio.aguchiku@usp.br, douglas.santana@usp.br, maruyama@usp.br

Abstract— Developing complex systems is one of the most challenging problems in Engineering nowadays. In the recent years, there has been an increase in air traffic, that demands a modernization in the current air traffic systems, which are really dependent in the human controller. Air traffic systems are considered safety critical and real time systems. The objective of the present work is applying a method of modeling and formal verification to the specific domain of air traffic systems. By adopting modeling and formal verification techniques, it is expected that an efficiency is ensured in the systems' project and development, besides ensuring also its correctness. A case study of three real modules of the Brazilian control centers is described. For the formal verification, the NuSMV tool is used and the properties to be checked are described in the linear temporal logic (LTL) and in the computational tree logic (CTL) to ensure that the system satisfies requirements of liveness and safety type.

Keywords— air traffic, CTL, formal verification, LTL, model checking.

Resumo— O desenvolvimento de sistemas complexos é atualmente um dos problemas mais desafiadores enfrentados pela Engenharia. Nos últimos anos, houve um aumento no tráfego aéreo, o que demanda uma modernização dos sistemas de tráfego aéreo atuais, muito dependentes na figura do controlador. Sistemas de tráfego aéreo são sistemas considerados críticos em segurança e de tempo real. O objetivo do presente trabalho é a aplicação de um método de modelagem e verificação formal para o domínio específico de sistemas de tráfego aéreo. Com a adoção de técnicas de modelagem e verificação formal, pretende-se garantir eficiência no projeto e no desenvolvimento desses sistemas, além da sua corretude. Descreve-se um estudo de caso de três módulos reais do sistema de tráfego aéreo, em operação atualmente nos centros de controle brasileiros. Para verificação formal, é utilizada a ferramenta NuSMV e as propriedades a serem verificadas são descritas nas lógicas linear temporal (LTL) e computacional de árvore (CTL) para garantir que o sistema satisfaça requisitos dos tipos vivacidade e segurança.

Palavras-chave— CTL, LTL, modelagem, tráfego aéreo, verificação formal.

1 Introdução

Nos anos 90, identificou-se o problema crônico de software: a maior parte dos sistemas ainda eram feitos por "artesãos" e, assim, principalmente os sistemas críticos, sofriam com atrasos, aumentos de custos e dificuldade de se lidar com eventuais falhas operacionais (Gibbs, 1994). Desde essa época, buscou-se realizar previsões para o comportamento do software no mundo real, através de análises matemáticas, apoiadas em métodos formais.

O desenvolvimento de sistemas complexos é atualmente um dos problemas mais desafiadores enfrentados pela Engenharia. Nota-se que há, frequentemente, uma necessidade de se atender requisitos restritivos, por muitas vezes justapostos ou ainda conflitantes e também de se garantir o suporte a certa modularidade, a uma adaptação fácil e ao reuso (Navasa et al., 2009). Simultaneamente ao aumento no nível de complexidade dos sistemas, também há uma crescente pressão para se reduzir o tempo total de desenvolvimento, o que dificulta a entrega de sistemas com um mínimo aceitável de defeitos (Baier & Katoen, 2008).

O objetivo principal na preparação de um projeto é assegurar que os possíveis defeitos sejam encontrados e eliminados o mais cedo possível (McConnell, 2004). Alguns estudos feitos nos últimos 25 anos por empresas como Hewlett-Packard, IBM, dentre outras, mostram que os custos com retrabalho aumentam drasticamente quanto mais o tempo passa, como demonstrado pela Tabela 1. Nota-se que, quanto mais cedo o defeito for detectado e corrigido, menos dano será causado nas fases posteriores e menor será o custo envolvido.

Tabela 1: Custo relativo de correção de defeitos, com base no momento em que são introduzidos e detectados - Adaptado de McConnell (2004)

Custo de correção	Deteção de defeitos				
	Req	Arq	Const	Testes	Pós-Entrega
Requisitos	1x	3x	5-10x	10x	10-100x
Arquitetura	-	1x	10x	15x	25-100x
Construção	-	-	1x	10x	10-25x

Pela utilização de modelagem e verificação formal, realiza-se uma avaliação do sistema anteriormente à sua construção e assim espera-se evitar tais problemas que podem ter um custo muito elevado ao projeto.

Nos últimos anos, o tráfego aéreo vem se intensificando e isso demanda uma modernização dos sistemas atuais, que são muito dependentes da figura do controlador. Trabalhos relacionados sugerem que a tendência é que uma maior auto-

mação seja cada vez mais adotada e, consequentemente, haverá uma maior exigência no desenvolvimento dos sistemas para garantir o seu pleno funcionamento. O objetivo do presente trabalho é a aplicação de um método de modelagem e verificação formal para o domínio específico de sistemas de tráfego aéreo. Sugere-se a adoção de técnicas de modelagem e verificação formal para garantir eficiência no projeto e no desenvolvimento desses sistemas, além da sua corretude.

2 Verificação Formal

O raciocínio formal para sistemas de hardware e software pode seguir duas linhas distintas: a prova de teoremas e a verificação de modelos. Enquanto na prova de teoremas utiliza-se uma técnica formal para confirmar se a implementação de um modelo está de acordo com a sua especificação, na verificação de modelos são criadas máquinas de estados finitos para assegurar a validade de uma propriedade (Sobeih et al., 2004).

Mais detalhadamente, para a verificação de modelos é necessário fornecer um modelo de estados finitos de um sistema e uma propriedade a ser verificada, para então confirmar a validade dessa propriedade através da inspeção dos caminhos possíveis do espaço de estados. Caso ela não se sustente, é gerado um contra-exemplo com o caminho em que a propriedade se tornou inválida (Baier & Katoen, 2008). O objetivo principal dos métodos formais é garantir a assertividade do sistema, com rigor matemático. Os métodos formais representam um grande potencial para a solução de problemas relacionados a estados incompletos, ambiguidades e inconsistências em uma fase inicial do projeto.

Dentre as lógicas utilizadas para Verificação Formal, destaca-se a Lógica Linear Temporal (LTL) e a Lógica de Árvore Computacional (CTL) (Baier & Katoen, 2008). A LTL é um formalismo que tem o objetivo de tratar aspectos de execução do sistema e questões de imparcialidade. É baseada em uma notação intuitiva e matematicamente precisa e estende a lógica proposicional ou de predicado.

O domínio temporal da LTL é o tempo discreto, sendo que a natureza do tempo na LTL é linear, ou seja, há apenas um sucessor a cada estado. A CTL, por sua vez, conta com uma estrutura de árvore, ou seja, há um nó principal e diversas ramificações. A CTL é baseada no tempo ramificado.

A notação utilizada nas lógicas possui os seguintes símbolos básicos e seus agrupamentos:

Notação da LTL:

$\Diamond$ eventualmente	$\bigcirc$ próximo
$\Box$ sempre	$\bigcup$ até que
$\wedge$ conjunção	$\Box\Diamond$ por infinitas vezes
$\neg$ negação	$\Diamond\Box$ eventualmente para sempre

Notação da CTL:

$\exists$ existe
$\forall$ para todo
$\bigcirc$ próximo
$\bigcup$ até que

3 Sistemas de Tráfego Aéreo

Os sistemas de Tráfego Aéreo podem ser considerados críticos, já que o controle do espaço aéreo é uma atividade que convive frequentemente com o risco de graves acidentes. São considerados também sistemas de tempo real, pois dependem de respostas em períodos de tempos bem definidos. Segundo Bessam & Kimour (2008), uma tarefa de tempo real é caracterizada por ter um agendamento para sua execução, e pode ser periódica ou aperiódica, ativada por uma interrupção externa. Sistemas de tempo real devem completar suas rotinas dentro de prazos bem determinados, para garantir a segurança da sua execução.

Segundo ICAO (2008), o sistema de Gerenciamento do Tráfego Aéreo (ATM) é uma composição de diferentes tipos de atores, que incluem componentes humanos, de informação, serviços e tecnologia. A tarefa do sistema de ATM é "manter o tráfego aéreo global organizado, interoperável, confiável, eficiente e seguro" (Azzopardi, 2015). O ATM é composto tanto pelo Controle de Tráfego Aéreo (ATC), quanto pelo Gerenciamento de Fluxo e Capacidade de Tráfego Aéreo (ATFM). O sistema ATC, que é foco do presente trabalho e que está em operação atualmente, é um sistema distribuído por construção, já que o espaço aéreo é divido em áreas e setores de controle, cada um com poder de decisão em sua área. Cada controlador de tráfego aéreo é responsável por um setor, que comporta aproximadamente 15 aeronaves (Erzberger, 2004).

Segundo de Oliveira (2007), a estrutura básica dos sistemas de tráfego aéreo não evoluiu muito desde os anos 60, quando se introduziu a tecnologia Radar. Os sistemas de tráfego aéreo atuais funcionam como uma ferramenta de suporte à decisão e seus principais atores são os controladores. A automação nesses sistemas ainda é considerada secundária, porém se torna cada vez mais relevante, uma vez que possibilita que as aeronaves voem com uma distância menor entre elas, permitindo assim uma melhor utilização do espaço aéreo (Pizzo, 2008).

3.1 Conceitos básicos

Pista Radar ou apenas pista é o nome dado para o conjunto de informações, tais como posição, altitude, velocidade, código de identificação (denominado indicativo da aeronave/*callsign*), entre outras, que tem origem em uma detecção de uma aeronave por um radar (ou em uma fusão de várias detecções). Cada pista é enviada aos ATC para composição de uma síntese da situação aérea, apresentada aos controladores (Pizzo, 2008).

O **plano de voo** (em inglês *flight plan* - FPL) é um documento submetido à autoridade responsável pela Aviação Civil anterior ao voo e, em linhas gerais, contém informações de identificação da aeronave (ex: código, modelo da aeronave), de equipamentos, de partida (ex: local e horário de partida), de rota (ou seja, por quais pontos - também chamados de aeródromos - a aeronave deverá passar) e de destino (ex: local e horário da chegada) (ICAO, 2007).

3.2 Iniciativas de Modelagem e Verificação Formal em Sistemas de Tráfego Aéreo

O trabalho de Zhao & Rozier (2014) trata o problema de perda de separação de aeronaves, ou seja, quando as aeronaves estão se aproximando e há uma possibilidade de colisão futura. Esse cenário pode ser dividido em três fases conforme o tempo que falta para a colisão. As aeronaves podem ser divididas conforme o tipo de equipamento que possuem: as aeronaves comuns que precisam necessariamente de comandos do Centro de Controle em solo para operar, aeronaves que possuem sensores para envio de sua posição e recebimento de outras aeronaves e por isso é capaz de fazer o controle autônomo. A modelagem e verificação são feitas utilizando-se a ferramenta NuSMV e as propriedades a serem verificadas são descritas em LTL/CTL. O motivo para a escolha do NuSMV (e cita-se também outra ferramenta chamada CadenceSMV) é a facilidade de se encontrar documentação, além de ser uma plataforma livre e frequentemente usada na indústria.

Esse trabalho está inserido no contexto do NextGen e faz parte de uma iniciativa da NASA. Ele propõe um método que vai desde a descrição do sistema de interesse, passando pela modelagem e verificação formal, incluindo recomendações de reformulação do modelo e/ou da propriedade de interesse quando a verificação falha e é gerado um contraexemplo.

O trabalho de Muñoz et al. (2004) trata de modelagem e verificação formal da operação de um centro de controle remoto automatizado. Em linhas gerais, o modelo trata as diversas posições por que uma aeronave passa desde o momento em que chega em uma região próxima ao aeroporto até o momento do seu pouso em solo ou vice-versa - do momento em que sai do solo para a decolagem até o momento em que sai dessa áerea limite do aeroporto. O modelo foi escrito na linguagem PVS (Sistema de Verificação de Protótipos), que é adequada para provadores de teoremas, e foi verificado usando um algoritmo de exploração de estados explícito.

Ambos os trabalhos tratam de modelagem de comportamento de sistemas relacionados ao domínio de Tráfego Aéreo. Porém não tratam de funções específicas do sistema de software implementado para ATC, por exemplo, mas sim de validação de cenários para auxiliar na tomada de decisão em nível de sistema. O objetivo do presente trabalho é chegar em um nível mais próximo ao comportamento do software.

O objetivo do trabalho de Kanoun et al. (1999) é a definição de uma nova arquitetura para o Coordenador Automático do Tráfego Aéreo (CAUTRA) francês, um sistema de tempo real distribuído, tolerante a falhas e que demanda um alto grau de disponibilidade. O CAUTRA é responsável pelo gerenciamento de planos de voo (Flight Plan Processing - FPP) e de dados de radar (Radar Data Processing - RDP). A arquitetura atual do sistema é composta de dois computadores redundantes de Dados Gerais (DG1 e DG2). O software de cada aplicação é replicado em quatro nós: dois principais (RDPpal, FPPpal) e dois secundários (RDPsec, FPPsec).

São propostas algumas alternativas de arquitetura, levando-se em conta estratégias de reconfiguração e de tolerância a falha, utilizando-se Redes de Petri Estocásticas Generalizadas (GSPNs). Examinam-se algumas modificações na arquitetura, como a quantidade de computadores ou a distribuição de nós, visando-se principalmente garantir a disponibilidade do sistema.

Dentre outros trabalhos relacionados, o trabalho de Coimbra et al. (2007) trata da modelagem em alto nível de um sistema ATC, porém não se utiliza a verificação formal. As decisões de arquitetura são descritas de forma informal, não sendo possível dessa forma utilizar técnicas de validação automática. Assim, a enumeração de riscos necessita da análise de um profundo conhecedor do sistema de interesse, para que estejam alinhados com a realidade. Esses riscos também são de difícil deteção. O foco principal desse trabalho é na simulação das possíveis soluções apontadas para resolução dos problemas enumerados.

4 Metodologia

Para o presente trabalho, idealizou-se um método de modelagem e verificação formal de sistemas críticos (Figura 1), com base em outros trabalhos de referência (Pohl, 2015; de Assis, 2009; Zhao & Rozier, 2014). Para a aplicação do método, são necessários três papéis distintos: o idealizador do sistema, geralmente se não for o usuário do sistema,

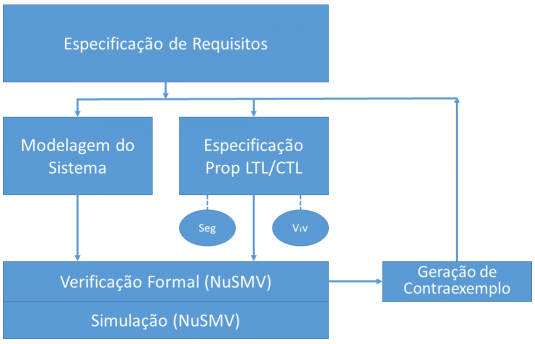

Figura 1: Método para Verificação Formal

é o responsável pela equipe que vai utilizá-lo; o arquiteto de sistemas, responsável pelo desenho teórico da arquitetura do sistema; engenheiro de sistemas, responsável pela modelagem do sistema e pela especificação de propriedades para sua verificação.

Inicialmente, é feita a especificação de requisitos pelo idealizador, geralmente em conjunto com o arquiteto. Nessa fase, é necessário delimitar o domínio de aplicação e também definir o escopo do desenvolvimento, ou seja, definem-se os subsistemas de interesse, as entradas e saídas do sistema, as interfaces e a forma de comunicação com os elementos externos e a descrição do Hardware utilizado. O foco no presente trabalho é o sistema de software, logo ao ser utilizada a palavra "sistema", subentende-se que seja de software.

Em seguida, o engenheiro de sistemas faz a modelagem do sistema em paralelo com a especificação de propriedades LTL/CTL, que na Figura 1 representam requisitos de Segurança e Vivacidade.

Então, faz-se a verificação formal com a ferramenta NuSMV[1], com o objetivo de assegurar que o sistema modelado está de acordo com os requisitos especificados no início. Idealmente, a Especificação de Requisitos deve ser verificada com o idealizador, para que todos estejam de acordo que as propriedades especificadas representam todos os requisitos especificados.

Por fim, tanto o modelo quanto as propriedades projetadas passam por uma fase de evolução. Esse aperfeiçoamento do modelo inicial e das propriedades se dá de uma forma iterativa, aplicando-se a verificação formal e fazendo modificações necessárias.

Como descrito por Zhao & Rozier (2014), a utilização do NuSMV com a especificações de propriedades utilizando-se lógicas LTL/CTL se justifica pela facilidade de se encontrar documentação, além de ser uma plataforma livre e frequentemente usada na indústria. Adiciona-se a isso a facilidade de se realizar a verificação formal, feita através da execução de um arquivo binário e também a facilidade de geração e compreensão de contraexemplos. Há ainda a possibilidade de se fazer simulação pela entrada de dados específicos no programa para verificar manualmente a progressão de estados do modelo.

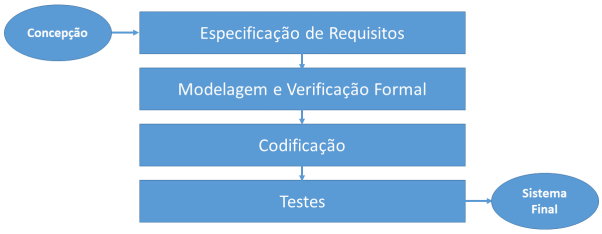

Figura 2: Ciclo da construção de um sistema

O diagrama da Figura 2 abrange o ciclo completo de desenvolvimento do sistema, desde sua concepção e Especificação até sua codificação propriamente dita, passando pela Modelagem e Verificação formal. O início desse processo é a idealização de um sistema de interesse e o fim é o sistema propriamente dito, como utilizado pelo usuário final.

O foco do presente trabalho é na aplicação desse método para sistemas de Tráfego Aéreo através da modelagem de comportamento de três módulos de um sistema com a ferramenta NuSMV, baseada em requisitos preestabelecidos, e na especificação e na verificação de propriedades na lógica CTL. As fases subsequentes do método (de codificação e testes) não serão abordadas. Dentre as limitações do trabalho, tem-se que os requisitos do sistema de interesse já foram previamente especificados e não é possível obter nenhum tipo de feedback do idealizador do sistema.

5 Estudo de Caso

O estudo de caso representa uma continuidade do trabalho apresentado em Aguchiku et al. (2015) - modelagem e verificação de uma versão simplificada do subsistema de Processamento de Dados de Voo, presente no sistema ATC brasileiro. Em linhas gerais, a tarefa desse subsistema é gerenciar dados de voos em uma região específica, seguindo uma série de protocolos pré-estabelecidos. A estrutura dos dados de voo é representada pelo FPL, que possui uma máquina de estados bem definida. O foco nesse trabalho foi a verificação de requisitos de segurança para a operação de transferência do controle de um FPL entre ATCs, que se baseia na troca de mensagens de coordenação entre os controladores e na consequente alteração de estados do FPL. Foi utilizada a ferramenta AutoFocus3, com a verificação de propriedades LTL/CTL através do plugin NuSMV.

[1]Para mais informações, consulte http://nusmv.fbk.eu/NuSMV

Os sistemas de Tráfego Aéreo são sistemas de consciência situacional, ou seja, seu objetivo é coletar informações de diversas fontes pertinentes, como radares, estações meteorológicas etc, e reuni-las da melhor forma possível para fornecer ao controlador uma visão da situação aérea atual, auxiliando-o na sua rápida tomada de decisão. Para isso, diversos algoritmos são empregados nas tarefas de, por exemplo, fusionar diferentes detecções de vários radares para exibir a posição aproximada de uma aeronave, realizar a correlação de planos de voo com as aeronaves que foram previamente detectadas e coordenar o envio e a recepção de mensagens entre centros para garantir que todo o ciclo de vida de um FPL seja percorrido. Para o presente trabalho, o foco será no sistema de software, em especial em três módulos relacionados: o módulo de vigilância e detecção de aeronaves (*DET*), o módulo de recepção e difusão de planos de voo (*REC*) e o módulo de correlação de planos de voo e pistas (*CORR*). O foco da verificação formal será na garantia de funcionamento do sistema como um todo, com funcionamento de seus módulos em paralelo. Foram especificadas e verificadas formalmente algumas propriedades identificadas como do tipo vivacidade (*liveness*) e segurança (*safety*) em CTL.

A principal tarefa do DET é receber, processar e armazenar no banco de dados as detecções de pistas provenientes do Radar, representado na Figura 3 como um sensor externo ao sistema. A máquina de estados referente ao DET é apresentada na Figura 4.

A principal tarefa do REC é receber e processar as mensagens de planos de voo e armazenar no banco de dados (BD) novos FPL e alterações nos já existentes. Essas mensagens são provenientes de outros ATCs, como representado na Figura 3. A máquina de estados referente ao REC é apresentada na Figura 5.

A principal tarefa do CORR é fazer a correlação entre pistas e planos de voo que apresentarem o mesmo indicativo. Este módulo é representado na Figura 3 como um módulo interno ao sistema, que não interage com o meio externo, porém utiliza-se dos dados guardados em banco para realizar a correlação. O módulo CORR não possui uma máquina de estados associada, mas sua tarefa é descrita através de operações feitas diretamente na linguagem SMV.

O CORR funciona de maneira assíncrona, ou seja, é acionado a cada nova rodada de pistas enviadas pelo Radar ou a cada recepção de um novo FPL.

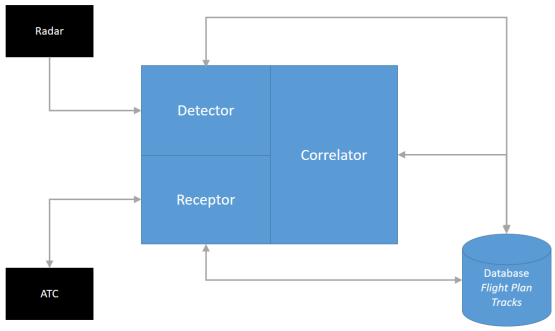

Figura 3: Diagrama representativo do modelo utilizado para o estudo de caso

5.2 Modelagem

5.2.1 Modelo desenvolvido em SMV

As máquinas de estados descritas anteriormente (Figuras 4 e 5) foram implementadas em SMV e funcionam paralelamente e independentemente.

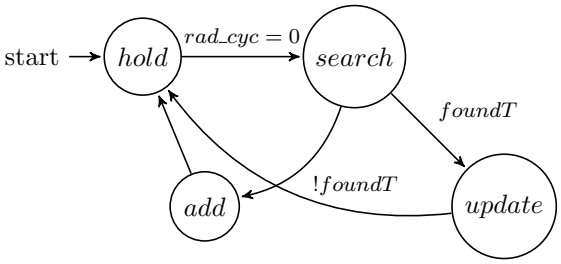

Figura 4: Máquina de Estados representativa do modelo do DET, utilizado para o estudo de caso

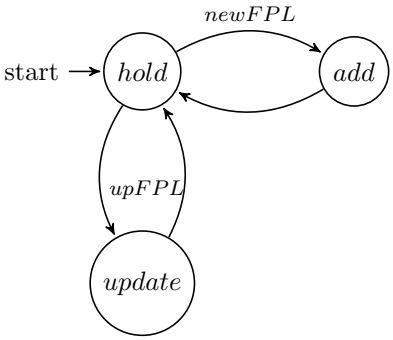

Figura 5: Máquina de Estados representativa do modelo do REC, utilizado para o estudo de caso

A representação das estruturas é feita através de variáveis dos tipos *enum* (E) e *boolean* (B). Vetores são identificados por colchetes ([]), processos (*process*) com máquinas de estado independentes por P e banco de dados é identificado por BC. Variáveis declaradas como globais:

- *FPL* [E]: BD FPL. Cada FPL pode representar três valores de indicativo (AAA, BBB ou

CCC);

- *track1..3* P: pistas, com indicativo (*CSN*) e número de pista;
- *CR* [B]: representação de FPL correlacionado (três posições) - se *TRUE*, há uma correlação daquele FPL, se *FALSE*, não há correlação;
- *radar_cycle* E: indica o ciclo de um radar de 4 unidades de tempo.

Variáveis declaradas no módulo DET:

- *new_track* E: nova pista recebida;
- *found_track* B: pista encontrada no BD *track1..3*.

Variáveis declaradas no módulo REC:

- *new_FPL* E: novo FPL recebido;
- *new_FPL_message* B: mensagem de novo FPL recebida;
- *update_FPL_message* B: mensagem de atualização de FPL recebida.

Operações declaradas no módulo DET:

- Gravação de pistas no BD: a pista será gravada no banco de dados caso não seja encontrada anteriormente e a operação será realizada sequencialmente até preencher completamente o BD *track*.
- Recepção de novas pistas: no início de cada ciclo, a pista identificada por *new_track* pode mudar de valor de indicativo, o que, em um cenário real, representa que o radar identificou uma nova pista nesse instante.

Operações declaradas no módulo REC:

- Gravação de FPL no BD: o FPL será gravado no BD caso a mensagem seja identificada como novo FPL. A operação funciona de modo similar à operação de gravação de pistas.
- Recepção de novos FPL e de novas mensagens: FPL identificado por *new_FP* pode mudar de valor de identificação.

Operação declarada no módulo CORR:

- Correlação de pista e FPL: o recebimento de pistas pelo módulo REC funciona como um gatilho para o módulo CORR tentar fazer a correlação entre os FPL recebidos e as pistas guardadas no BD. Logo, essa operação é teoricamente assíncrona, com o acionamento através de um gatilho. Porém, torna-se síncrona no presente caso de estudo, em que se desconsidera o eventual recebimento de um novo FPL como gatilho, situação que pode ocorrer no sistema real, e leva em conta apenas o recebimento de pistas, que ocorre periodicamente.

5.3 Especificação de propriedades

As propriedades de vivacidade e segurança que foram especificadas são apresentadas nessa seção, acompanhadas de sua equação descrita em símbolos, como apresentado na seção 2, e de sua descrição em linguagem natural.

5.3.1 Recepção de FPL

$$\forall\Box((s.REC = hold \;\&\; REC\_run.new\_FP\_message \;\&\; REC\_run.new\_FP = AAA) \rightarrow \forall\Diamond(s.FP[1] = AAA \mid s.FP[2] = AAA \mid s.FP[3] = AAA))$$

Vivacidade Quando o módulo de Recepção está no estado de espera (estado *hold*) e recebe-se uma mensagem de novo FPL, indicado pelo indicativo AAA, **sempre** ele será adicionado ao BD, em alguma das posições. A propriedade exemplificada é parcial, sendo que a propriedade completa contempla todos os tipos de FPL (AAA, BBB ou CCC).

5.3.2 Detecção de pista

$$\forall\Box((s.DET = hold \;\&\; s.radar\ cycle = 0 \;\&\; DET\_run.new\_track = AAA) \rightarrow \exists\Diamond(s.track1.csn = AAA \mid s.track2.csn = AAA \mid s.track3.csn = AAA))$$

Vivacidade Quando o módulo de Detecção está no estado de espera (estado *hold*), o ciclo do radar se inicia e recebe-se uma nova pista, **eventualmente** ela será adicionada ao BD, em alguma das posições. A propriedade exemplificada é parcial, sendo que a propriedade completa contempla todos os tipos de pistas (AAA, BBB e CCC). Uma propriedade mais forte seria dizer que sempre seria adicionada, porém, como no modelo foram separadas as ações de adicionar e de atualizar, é possível que a pista já tenha sido previamente adicionada e sofra apenas uma atualização.

5.3.3 Correlação entre FPL e pista

O algoritmo de correlação de FPL e pista pode ser realizado de diversas maneiras. Por simplificação, a ativação é feita apenas por um gatilho, que é a recepção de pistas, indicada pela variável *radar_cycle*. A busca no BD por FPL com o mesmo indicativo poderia ser feita por um algoritmo de varredura convencional (pela utilização de um *loop* de iteração), porém em SMV é possível especificar

essa busca através de uma intervenção direta no estado da variável *boolean CR*, ou seja, por comparação do indicativo da nova pista recebida com o valor de cada FPL.

$$\forall\Box((s.FP[1] = AAA \ \& \ s.track1.csn = AAA) \rightarrow \forall\Diamond(s.CR[1] = TRUE))$$

Vivacidade Quando a informação de um FPL no BD é igual à informação de uma pista no BD, **sempre** ocorrerá a correlação. A propriedade exemplificada é parcial, sendo que a propriedade completa contempla todas as combinações de FPL e pista (um total de 27 combinações possíveis).

$$\forall\Box((s.FP[1] = AAA \ \& \ s.track1.csn \ != AAA \ \& \ s.track2.csn \ != AAA \& \ s.track3.csn \ != AAA) \rightarrow \forall\Diamond(s.CR[1] = FALSE))$$

Segurança Essa propriedade visa evitar correlações incorretas entre pista e FPL. Caso o FPL de indicativo AAA esteja salvo na primeira posição do BD FPL e nenhuma pista de indicativo AAA estiver presente no BD de pistas, a correlação para a primeira posição sempre será falsa. A propriedade exemplificada é parcial, sendo que a propriedade completa contempla todas as combinações de FPL e pista (um total de 9 combinações possíveis).

6 Resultados

Foi estabelecido um método de modelagem e verificação formal (seção 4) e de diretrizes para sua aplicação em sistemas de tráfego aéreo, através de um estudo de caso de três subsistemas reais em operação (seção 5).

As propriedades especificadas na seção 5.3 foram verificadas utilizando-se a ferramenta NuSMV. Todas elas foram confirmadas no modelo final especificado em SMV (um exemplo de saída da ferramenta para uma propriedade confirmada é mostrado na Figura 6). Dessa maneira, pôde-se confirmar a corretude do sistema em relação aos aspectos que se propôs verificar. A verificação formal possibilita a evolução do modelo e, se for o caso, também das propriedades de interesse. O procedimento utilizado é através da avaliação dos contraexemplos.

Algumas técnicas para melhor utilizar a ferramenta:

- **Garantir que a propriedade está sendo mesmo verificada e que ela reflete o requisito que se deseja verificar -** Ao descrever uma propriedade, inverter a consequência para garantir que seja gerado um contraexemplo. Em seguida, avaliar o caminho do contraexemplo para assegurar pelo inverso que a propriedade realmente reflete o que deseja-se verificar.

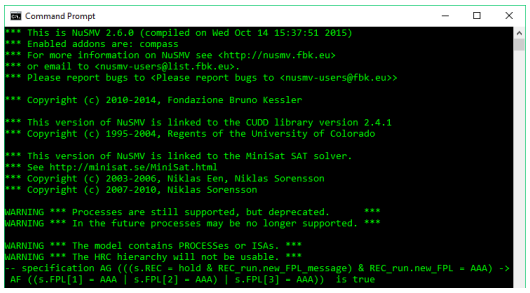

Figura 6: Saída da ferramenta NuSMV

- **Verificação de mais de uma propriedade -** Verificar inicialmente cada propriedade independentemente e depois verificar todas em cadeia com &. Através da verificação das propriedades em conjunto, constatou-se a necessidade de excluir da Correlação o caso em que FP e TR são vazios.

- **Verificação de uma propriedade com múltiplos casos -** Expandir a propriedade em todos os casos, realizar a verificação de cada uma independentemente e depois em cadeia com &. Assim, é possível identificar algum estado indesejado que pode ser alcançado por algum erro na modelagem.

7 Conclusão

Os trabalhos relacionados mostram que os sistemas atuais de tráfego aéreo ainda são muito dependentes da figura do controlador para funcionar adequadamente, o que se traduz em uma limitação com relação à evolução da demanda dos serviços de tráfego aéreo. Há, porém, uma predisposição nessa área para a adoção de técnicas de automação, com o objetivo de se atender um número cada vez maior de aeronaves. Surgem então novos conceitos que permitem uma maior autonomia para as aeronaves e, dessa forma, abrem espaço para que novas formas de organização e controle do espaço aéreo sejam propostas e aplicadas no futuro.

Verifica-se então a necessidade de uma maior exigência no desenvolvimento dos sistemas, para garantir seu funcionamento como planejado. Como os sistemas de tráfego aéreo são críticos em segurança, de tempo real, é indispensável que sejam eficientes, confiáveis e seguros para que não incorram em erros que podem causar acidentes com danos irreparáveis. A utilização de verificação formal mostra-se ainda mais necessária para garantir a corretude desses sistemas computacionais e representa uma forma mais assertiva de projeto e desenvolvimento.

O presente trabalho propôs a aplicação de um método de modelagem e verificação formal para o domínio específico de tráfego aéreo. Para isso, foram fornecidas diretrizes através de um estudo de caso baseado em três módulos reais, em operação nos ATCs brasileiros. Não foi possível encontrar nenhum outro trabalho aplicado a modelos de subsistemas de tráfego aéreo, baseados no sistema real.

Ressalta-se ainda que o modelo proposto não é muito grande em número de estados e a escalabilidade é uma questão importante a ser levada em conta. É possível que o sistema fique muito grande e que não seja possível verificá-lo. Imagina-se que esse problema possa ser contornado com a simplificação das máquinas de estados: uma solução é projetar um modelo mais genérico que trate de uma visão macro ou ainda projetar outros modelos com visões mais específicas que possam ser verificados independentemente.

Referências

Aguchiku, F. S., Costa, R. L., de Souza, E. C. & Maruyama, N. (2015). The model-driven development approach & formal specification for air-traffic control system operations: Practical comments and case studies, *SITRAER Air Transportation Symposium*.

Azzopardi, M. A. (2015). *Computational Air Traffic Management*, PhD thesis, CRANFIELD UNIVERSITY.

Baier, C. & Katoen, J.-P. (2008). *Principles of Model Checking (Representation and Mind Series)*, The MIT Press.

Bessam, A. & Kimour, M. (2008). Multi-view description of real-time systems' architecture, *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* **2**(2): 339–346.

Coimbra, W. A., Fialho, Á. R., de Andrade, M. T., Gomi, E. S., Arakaki, R. & Giannotto, E. C. (2007). Analysis and simulation of non-functional requirements applied to an air-traffic control system, *International Conference on Information Systems and Technology Management*, Vol. 4.

de Assis, F. H. (2009). *Checagem de arquiteturas de controle de veículos submarinos: uma abordagem baseada em especificações formais*, Master's thesis, Escola Politécnica da Universidade de São Paulo.

de Oliveira, Í. R. (2007). *Análise de risco da operação de espaçamento temporal aerotransportado por meio de um modelo em rede de Petri estocástica e dinamicamente colorida*, PhD thesis, Escola Politécnica da Universidade de São Paulo.

Erzberger, H. (2004). *Transforming the NAS: The Next Generation Air Traffic Control System*, National Aeronautics and Space Administration.

Gibbs, W. W. (1994). Software's chronic crisis, *Scientific American* **271**(3): 72–81.

ICAO (2007). Amendment no. 1 to the procedures for air navigation services. Doc 4444 - ATM/501.

ICAO (2008). Manual on air traffic management system requirements. Doc 9882 - AN/467.

Kanoun, K., Borrel, M., Morteveille, T. & Peytavin, A. (1999). Availability of CAUTRA, a subset of the french air traffic control system, *IEEE Transactions on Computers* **48**(5): 528–535.

McConnell, S. (2004). *Code Complete, Second Edition*, Microsoft Press, Redmond, WA, USA.

Muñoz, C. A., Dowek, G. & Carreño, V. (2004). Modeling and verification of an air traffic concept of operations, *SIGSOFT Softw. Eng. Notes* **29**(4): 175–182.

Navasa, A., Pérez-Toledano, M. A. & Murillo, J. M. (2009). An ADL dealing with aspects at software architecture stage, *Inf. Softw. Technol.* **51**(2): 306–324.

Pizzo, W. N. (2008). *Avaliação da disponibilidade de sistemas computacionais críticos para o controle do espaço aéreo por meio de modelo analítico da teoria de filas*, Master's thesis, Escola Politécnica da Universidade de São Paulo.

Pohl, K. (2015). *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant*, Rocky Nook, Inc.

Sobeih, A., Viswanathan, M. & Hou, J. (2004). Check and simulate: a case for incorporating model checking in network simulation, *Formal Methods and Models for Co-Design, 2004. MEMOCODE '04. Proceedings. Second ACM and IEEE International Conference on*, pp. 27–36.

Zhao, Y. & Rozier, K. Y. (2014). Formal specification and verification of a coordination protocol for an automated air traffic control system, *Science of Computer Programming* **96**: 337–353.