CÁLCULO DO IPI USANDO CEP

Henrique Lange Severino
Acadêmico do curso de Ciência da Computação – Universidade de Passo Fundo
111957@upf.br
Leonardo Rauber
Acadêmico do curso de Ciência da Computação - Universidade de Passo Fundo
132789@upf.br
Marcelo Trindade Rebonatto
Professor do curso de Ciência da Computação - Universidade de Passo Fundo
rebonatto@upf.br
Luiz Eduardo Schardong Spalding
Professor do curso de Física - Universidade de Passo Fundo, Hospital São Vicente de Paulo
spalding@upf.br

Resumo. *No presente trabalho, é apresentado o Índice Pulmonar Integrado (IPI), que possibilita uma análise rápida sobre o estado respiratório de um paciente. Foi desenvolvida uma aplicação com uso de Processamento de Eventos Complexos (CEP) para a geração do índice e envio de alertas correspondentes à equipe médica. A implementação usou o Drools, um motor CEP open source e na geração de eventos, são obtidos dados em uma base de dados que armazena internações de pacientes em uma UTI.*

Palavras-chave: *Índice Pulmonar Integrado. Processamento de Eventos Complexos, Drools*

1. INTRODUÇÃO

O Índice Pulmonar Integrado (IPI) utiliza as medidas de quatro parâmetros, frequência respiratória (*respiratory rate*), nível de dióxido de carbono exalado pelo corpo (etCO2), saturação do oxigênio no sangue (SpO2) e frequência cardíaca (*pulse rate*) para fornecer uma avaliação simples do estado respiratório de um paciente (criança e adulto). O IPI é apresentado como um único índice de 1-10, onde 10 indica um estado respiratório normal e 1, um estado respiratório que necessita de intervenção médica imediata. Ele fornece ao médico uma rápida avaliação do estado respiratório do paciente, simplificando a integração de informações complexas e a avaliação clínica (Berkenstadt et al., 2012).

Um conjunto de dados, gerados por qualquer aplicação, que tem como objetivo ser transmitido para algum outro ponto ou aplicação pode ser chamado de “fluxo de dados”. A principal diferença entre um fluxo comum de dados e um evento é a forma como podem ser computadas as informações a eles relacionadas. Enquanto um processamento comum tem a custosa tarefa de acessar o banco de dados e buscar as informações, o processamento de um evento dá-se em tempo quase próximo ao real, no momento de sua detecção (Palmer et al., 2008).

O Multiparameter Intelligent Monitoring in Intensive Care (MIMIC II) é um banco de dados mantido pelo Massaschusetts Institute of Technology (MIT) e contém dados fisiológicos e sinais vitais de milhares de pacientes internados entre 2001 e 2008. O objetivo do armazenamento de todos os registros de entrada e internação de cada paciente nesse banco de dados é a sua utilização em pesquisas (Physionet, 2013).

Isso posto, este artigo descreve o cálculo do IPI, bem como uma implementação desse cálculo, gerando alertas à equipe médica, com uso de processamento de eventos complexos.

2. COMPLEX EVENT PROCESSING

Um evento corresponde a qualquer fluxo de dados que tenha um valor significativo no mundo real. Esse método de análise de variados fluxos de múltiplas fontes e geração do evento resultante baseando-se em regras pré-determinadas denomina-se Complex Event Processing (CEP). O processamento de eventos complexos pode reagir a situações sensíveis e críticas em tempo próximo ao real, o que é uma característica determinante nesses casos, quando, por exemplo, se trabalha com a vida de um paciente (Yao, 2011). Pode-se chamar de "evento complexo" o evento resultante da comparação de diversos outros eventos singulares.

Um motor CEP constitui a estrutura responsável pela organização e execução desses processos. Existem diversos motores CEP disponíveis, que variam entre motores comerciais e *software* livre. Todo o processo de análise de um fluxo, filtragem dos eventos significativos, ativação e saída de eventos complexos por meio da comparação entre os eventos simples ou complexos é realizado dentro da estrutura denominada CEP Engine – um motor CEP.

Os motores podem ser classificados como comerciais ou de software livre. Como exemplos de motores comerciais, destacam-se: Coral8 Engine, CEP Processor 11g e Esper. Como exemplos de motores de *software* livre, citam-se: Etalis, JRule Engine, Amit e Drools, entre outros.

Drools é um Business Rule Management System (BRMS), que realiza o encadeamento e a relação de eventos de entrada, baseando-se no seu motor de regras para gerar uma saída composta de um ou mais eventos complexos. Também conhecido como JBoss Rules, ou plataforma BRMS, trata-se de um projeto *open source* mantido pela empresa Red Hat. Atualmente, sua versão estável em mercado é a 5.5.0, lançada em 13 de novembro de 2012 (JBoss, 2012). O motor Drools foi implementado com base em um aprimoramento do algoritmo Rete (Forgy, 1981).

3. MIMIC II

O MIMIC II é uma base de dados confiável e consistente que contém dados de um grande número de pacientes internados em um centro de tratamento intensivo. Essa base contém informações suficientemente satisfatórias para a geração de eventos e é disponibilizada para pesquisas e experimentações em centros de pesquisa (Physionet, 2013).

O MIMIC II é dividido em dois bancos de dados principais – o Waveform Database, onde estão contidas ondas fisiológicas, em alta resolução, além de medições de minuto a minuto de pacientes internados. O segundo banco de dados é o Clinical Database, que contém os dados clínicos dos pacientes, bem como seus arquivos de internação no hospital.

O Clinical Database apresenta 38 tabelas, das quais 11 são reservadas a informações referentes à identificação de pacientes, médicos, códigos de equipamentos, equipamentos médicos e de laboratório, itens demográficos, entre outros. Três tabelas são referentes à duração de internações, a tempo de atendimento e tempo de utilização dos equipamentos. Dentre as demais tabelas, pode-se salientar a tabela ChartEvents, que reúne os registros clínicos do paciente, itens hospitalares utilizados, tempo de internação e os resultados do tratamento.

4. CÁLCULO DO IPI

O cálculo do IPI é realizado usando os valores dos sensores fisiológicos de quatro parâmetros: *respiratory rate*, etCO2, SpO2 e *pulse rate*. Os parâmetros etCO2 e *respiratory rate* são enquadrados em cinco classificações em intervalos definidos: *very low* (VL), *low* (L), *normal* (N), *high* (H) e *very high* (VH). A classificação dos valores do etCO2 é dividida nos seguintes intervalos: etCO2 < 20 definido como VL; etCO2 > 23 e < 30, como L; etCO2 > 32 e < 44, como N; etCO2 > 58 e < 64, como H; e etCO2 > 64, como VH.

O parâmetro *respiratory rate* (resp) é dividido nos seguintes intervalos: resp < 5 definido como VL; resp > 5 e < 8, como L; resp > 9 e < 18, como N; resp > 24 e < 35, como H; e resp > 38, como VH.

O cálculo leva em consideração a sobreposição entre os intervalos, proporcionando graus para cada intervalo. Isso ajuda a tornar a lógica usada para definir o IPI precisa e sensível a alterações dos valores de parâmetros. O IPI começa com o cruzamento dos valores do *respiratory rate* e do etCO2, de acordo com a Tabela 1.

Tabela 1 – Relação entre Respiratoty Rate e etCo2 no cálculo do IPI

SpO2 e pulse rate **normal**		Respiration rate (RR) values ranges, bpm								
		VH		H		N		L		VL
EtCO2 values ranges mmHg	VH	2		5		4		3		2
	H	3		8		7		5		3
	N	4		8		10		7		6
	L	5		6		8		4		2
	VL	3		3		4		2		2

Fonte: Berkenstadt et al., 2012.

A tabela mostra o resultado do cruzamento entre os valores de *respiratory rate* e etCO2. Os valores de SpO2 (< 94) e *pulse rate* (> 59 e < 100) são normais para não alterar o resultado do cruzamento. Nas linhas cinzas, quando um parâmetro tem adesão parcial em faixas adjacentes, o cálculo torna-se mais complexo. O resultado do cruzamento dos dois parâmetros será comparado com o valor do SpO2. Caso o SpO2 seja normal (> 94) não irá modificar o IPI; caso seja muito baixo (< 86), o IPI será 1, não importando o valor dos outros parâmetros. Se o valor de SpO2 ficar entre 95 e 85, o IPI será modificado de várias formas.

Após o resultado do IPI comparado com o SpO2, o IPI será calculado com o *pulse rate* da seguinte forma:

- Se o *pulse rate* for < 40, o etCO2 > 50 e o *respiratory rate* > 39, o resultado do IPI se reduz em um;
- Se o *pulse rate* for > 119, o etCO2 < 30 e o *respiratory rate* < 8, o resultado do IPI se reduz em um.

Caso o valor do *respiratory rate* seja 0 (parada respiratória), o IPI é imediatamente classificado como 1, informando que o paciente não está respirando (Berkenstadt et al., 2012).

5. IMPLEMENTAÇÃO E RESULTADOS

A implementação foi realizada na linguagem de programação Java, em ambiente Linux (Ubuntu). A aplicação, que trabalha com quatro medidas diferentes, utiliza o SpO2, *pulse rate* e *respiratory rate* oriundos do banco de dados MIMIC II. A aquisição dos dados utilizados como eventos é feita com base na internação do paciente de id 507. A última medida, ETCO2, é obtida com valores randomicamente gerados por meio de um gerador construído dentro da própria aplicação, devido ao fato de o paciente 507 não possuir dados de etCO2 em sua internação.

A quantidade total de eventos utilizados na execução da aplicação, de todos os parâmetros analisados, foi de 3824. A Tabela 2 mostra as faixas de valores, a ação da equipe médica e o número de ocorrências de cada um dos possíveis alertas de IPI.

Tabela 2. Resultados IPI

Faixa de valores	Ação	Número de ocorrências
10	Normal	201
9 e 8	Within normal range	1 516
7	Close to normal range; requires attention	533
6 e 5	Requires attention and may require intervention	4
4 e 3	Requires intervention	1 555
2 e 1	Requires immediate intervention	15

Pelos dados mostrados na Tabela 2, percebe-se a importância do cálculo automatizado do IPI em situações em que o quadro de saúde de um paciente é delicado. Dentre os resultados gerados, aproximadamente 41% (IPI de 1 a 4) correspondem a situações em que o paciente necessita de intervenção, sendo esta urgente em 15 casos. Esse índice pode melhorar a qualidade dos serviços de atenção prestados aos pacientes internados em situação de risco à saúde.

6. CONSIDERAÇÕES FINAIS

O IPI é um fator importante no monitoramento da saúde de um paciente. Tornou-se possível a realização do cálculo desse índice sobre um motor CEP, recebendo eventos por meio de geradores. Os eventos utilizados, por sua vez, são obtidos em uma base de dados própria para esse tipo de aplicação. Com o trabalho realizado, percebeu-se o funcionamento do CEP sob as condições propostas, bem como o seu funcionamento adequado.

No seguimento desse trabalho, o desenvolvimento de técnicas de tolerância a falhas e processamento em tempo real de aplicações com uso de CEP faz-se necessário. Tais funcionalidades buscam obter garantias do correto processamento dentro de um tempo adequado. Além disso, a implementação de um motor de CEP, embarcado num dispositivo simples e pequeno como uma RaspBerry PI, ou, mesmo, uma BeagleBoard, pode trazer significativo avanço no monitoramento de pacientes.

7. REFERÊNCIAS

Berkenstadt, H., Ben-Menachem, E., Herman, A., & Dach, R. (2012). An evaluation of the Integrated Pulmonary Index (IPI) for the detection of respiratory events in sedated patients undergoing colonoscopy. Journal of clinical monitoring and computing, 26(3), 177–81. doi:10.1007/s10877-012-9357-x

Forgy, C. L. (1981). Rete : A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, 19(1982), 17–37.

jBoss. (2012). Drools. Retirado de http://www.jboss.org/drools/

Palmer, M. Džmuráň, M. (2009). An Introduction to Event Processing, Powering Real-Time, Intelligent Business Applications. Retrieved October 24, 2012, from http://www.cnetdirectintl.com/direct/fr/2009/progress/0907_centre_ressources/ressources/1-1_LB_UK_introduction_CEP.pdf

Physionet. (2013). MIMIC II. Retirado de http://physionet.org/mimic2/

Yao, W., Chu, C.-H., & Li, Z. (2011). Leveraging complex event processing for smart hospitals using RFID. Journal of Network and Computer Applications, 34(3), 799–810. doi:10.1016/j.jnca.2010.04.020