Modelo Imunológico de Diagnóstico

Carine G. Webber, Evânia Viganó, Rodrigo Possamai

Centro de Ciências Exatas e Tecnologia - Universidade de Caxias do Sul (UCS)
Caixa Postal 15.064 – 91.501-970 – Caxias do Sul – RS – Brazil

{cgwebber,evigano,rpossama}@ucs.br

Abstract. *Classical approaches to diagnosis (model-based, abductive and constraint -based diagnosis) depend on the definition of rules setting normal and abnormal expected behaviours. In some contexts such previous definition is not so clear. In this article we propose a new diagnosis model inspired by natural immune system, in special Matzinger´s Danger Theory. This approach has the advantage of discriminating among abnormal situations, those which may cause danger to a system.*

Resumo. *As abordagens tradicionais de diagnóstico (diagnóstico baseado em modelos, abdutivo, e baseado em restrições) operam sobre padrões pré-definidos normais e anormais de funcionamento do sistema. Elas não tratam problemas onde tal classificação pode ser variável. Neste artigo propõe-se um novo modelo de diagnóstico inspirado no sistema imunológico natural buscando-se tal flexibilidade. Considerou-se para a elaboração desse modelo a Teoria Imunológica do Perigo, proposta por Matzinger. Tal abordagem traz como vantagem a possibilidade de um sistema computacional discriminar situações anormais de perigo daquelas que embora sejam anormais, não causarão danos ao sistema e seu ambiente.*

1. Introdução

Tradicionalmente, uma tarefa de diagnóstico consiste em observar um processo (industrial, biológico ou social) através de um conjunto de variáveis que o descreve, a fim de caracterizar seu estado e atribuir um significado a ele (normal, anormal, ou falha). Um processo de diagnóstico é iniciado toda vez que mudanças nos valores esperados para as variáveis observadas forem detectadas. Hipóteses devem ser geradas a fim de explicar a origem do funcionamento anormal. Por fim, um diagnóstico deve fornecer uma análise apontando a causa do funcionamento anormal observado.

As abordagens clássicas de diagnóstico são: diagnóstico baseado em modelos (Reiter, 1987), diagnóstico abdutivo, e diagnóstico baseado em restrições (Console, 1989). Essas abordagens são baseadas em conjuntos de regras que descrevem o funcionamento normal e anormal do sistema. Nas abordagens distribuídas segue-se este mesmo modelo, porém as variáveis observadas podem estar situadas em diferentes locais (Frohlich, 1997; Ross, 2003). Nestes casos, agentes de software podem monitorar componentes espacialmente distribuídos, interagir entre si, compartilhar informações coletadas, e construir um diagnóstico completo através de mecanismos de cooperação.

Em alguns cenários, porém, têm-se observado que a caracterização de uma situação como normal ou anormal depende do estado de variáveis temporais e suas combinações, sendo algumas vezes difícil de ser estabelecido. Por exemplo, o tempo de

espera considerado normal para um cliente em uma agência bancária varia para cada dia do mês, e depende de fatores como: número de caixas automáticos em funcionamento, proximidade das férias escolares, quantidade e duração das ligações telefônicas recebidas na agência, etc. Logo, determinar padrões normais e anormais de tempo de espera nestas condições constitui uma tarefa quase impossível. Situações como esta tornam inviável o uso de modelos tradicionais de diagnóstico.

Visando investigar soluções aplicáveis em tais problemas, desenvolveu-se um modelo de diagnóstico inspirado no comportamento do sistema imunológico natural (SIN). Os sistemas imunológicos artificiais têm sido aplicados em domínios como a classificação de dados, o reconhecimento de padrões e o diagnóstico e detecção (De Castro, 1999; De Castro, 2000; Dasgupta, 2006). Neste trabalho nos interessamos às tarefas de diagnóstico que o SIN realiza, em especial a abordagem dada pela Teoria do Perigo (Matzinger, 2002).

A fim de desenvolver apropriadamente estes temas, organizou-se o presente artigo em 4 seções. A seção 2 apresenta inicialmente uma breve introdução ao sistema imunológico, cujas características inspiram este trabalho. Dando continuidade, a seção 3 descreve uma metodologia baseada na Teoria do Perigo e desenvolvida neste trabalho para a concepção de um sistema de diagnóstico. A seção 4 apresenta um exemplo de sistema multiagentes aplicado ao diagnóstico do aluno durante um processo de resolução de exercícios de programação, ilustrando brevemente o funcionamento do sistema. Para concluir, a seção 5 discute alguns aspectos relevantes para pesquisas futuras.

2. Sistema Imunológico Natural

A principal função fisiológica do sistema imunológico natural (SIN) nos vertebrados é a defesa contra microorganismos infecciosos (Janeway, 2000). Uma habilidade importante do SIN está em distinguir o próprio (células do corpo) do não próprio (antígenos). Sabe-se que a presença de um antígeno induz o corpo a produzir anticorpos específicos que geram uma resposta imunológica adaptativa.

O SIN é composto por órgãos (medula óssea, timo, baço, e linfonodos) e células (granulócitos, macrófagos, células dendríticas e linfócitos B e T). Os linfócitos são células muito importantes. Os linfócitos B (ou células B) tem como principal função a produção de anticorpos como resposta a presença de proteínas estranhas de bactérias, vírus e células tumorais. Os anticorpos são células especializadas que reconhecem e se ligam a proteínas específicas. A produção de anticorpos e sua ligação a substâncias estranhas ou antígenos é um meio de sinalizar a outras células para assimilar, matar ou remover tal substância do corpo.

Os linfócitos T se dividem em duas classes. A primeira compreende as células T citotóxicas, que são importantes, pois atacam diretamente células tumorais, infectadas por vírus e alguns parasitas. A segunda classe compreende as células T auxiliares, cuja função principal é aumentar o potencial da resposta imunológica pela secreção de substâncias que ativam células brancas que lutam contra infecções. Os linfócitos não exibem nenhuma atividade funcional a menos que sejam apresentados a um antígeno por uma célula apresentadora de antígenos. Os linfócitos circulam continuamente em nossa corrente sanguínea em direção aos órgãos linfáticos. Quando microorganismos patogênicos são capturados em um tecido linfóide, os linfócitos que os reconhecem ficam retidos para proliferarem e se diferenciarem em células efetoras, capazes de lutar

contra uma infecção.

O primeiro modelo teórico do SIN foi a teoria da Discriminação Próprio e Não-Próprio proposta por Burnet (1959). Segundo este modelo, o SIN tem um comportamento baseado na discriminação entre o próprio (células do corpo) e o não próprio (antígenos). Cada célula do corpo carrega moléculas únicas que a identificam como constituintes do próprio. Todo corpo reconhecido como não-próprio gera automaticamente uma resposta imunológica. O segundo modelo, proposto por Janeway (2000), foi a teoria do Não-Próprio Infeccioso. Este modelo foi reconhecido e elaborado a partir da constatação de que o SIN reage somente a células não próprias infecciosas.

Recentemente, a Teoria do Perigo (TP), proposta por Matzinger (2002), apresentou novos elementos para a compreensão do SIN, defendendo que uma resposta imune é uma reação a um organismo que causa perigo ao corpo. A ideia central da TP é que o SIN não responde ao não-próprio (como nos modelos anteriores) mas a situações de perigo ou dano às células próprias. Isso explica porque nosso corpo tolera certos organismos não-próprios como, por exemplo, alimentos e bactérias que auxiliam na digestão. Um aspecto importante para a TP é o fato de que ela assume que são os tecidos do corpo que iniciam uma resposta imunológica pelo envio de sinais de perigo. Tais sinais de perigo são destinados às células apresentadoras de antígenos (CAA). Estas células são capazes de capturar antígenos (macrófagos e células dendríticas). Como elas não diferenciam próprio do não-próprio, elas capturam quaisquer antígenos que estejam causando o sinal de perigo. As células T reconhecem as CAA e produzem a resposta imune para proteger o corpo.

3. Metodologia

Os SIN inspiraram a concepção de 4 principais abordagens algorítmicas que são: Seleção Negativa (Forrest, 1994), Modelo de Perigo (Matzinger, 2002; Aickelin, 2002), Seleção Clonal (Weinand, 1990; De Castro, 1999), e Redes Imunes (Farmer, 1986; Fukuda, 1993). Em trabalhos precedentes nós aplicamos algoritmos de seleção negativa para a discriminação próprio/não próprio, onde agentes encapsulam regras que descrevem comportamento normal e anormal. Os agentes coletivamente produzem um diagnóstico. Entretanto, através de estudos sobre a TP, observou-se que ela pode ser útil na concepção de ambientes de aprendizagem. Nossa premissa inicial é que a TP se aplica à modelagem de estratégias didáticas sobre quando o sistema deve intervir durante a resolução de problemas. Para esclarecer os pontos relevantes do modelo natural, elaborou-se uma metodologia fundamentada na Teoria do Perigo de Matzinger e na concepção de sistemas de diagnóstico distribuído baseados em multiagentes. A metodologia proposta compõe-se de 3 etapas : a) modelagem conceitual, b) modelagem multiagentes, c) implementação, testes e avaliação.

3.1. Primeira Etapa: Modelagem Conceitual

Na etapa de modelagem conceitual deve-se definir o objetivo do sistema de diagnóstico, o que inclui a identificação de situações do sistema que caracterizam os estados de seu funcionamento. Em geral, utiliza-se variáveis que armazenam os dados observáveis do ambiente. Estados anormais de funcionamento do sistema são representados por valores de variáveis que não correspondem a escopos aceitáveis. Todo valor inesperado para uma variável deve ser associado a uma causa (i.e, mal funcionamento de um componente), a um sinal de perigo e a amplitude deste sinal. Isto porque o modelo

permite que se delimite uma zona de perigo para cada sinal.

A delimitação de zonas de perigo é uma ferramenta útil e que diferencia este modelo dos demais. De fato, o que pode caracterizar uma situação de perigo em determinado momento (funcionamento anormal), pode ser tratado como normal em outros períodos. Esta consideração pode ser facilmente observada em problemas dinâmicos e não determinísticos onde seja necessário antecipar situações de perigo ligadas ao tráfego em uma auto-estrada ou ainda a troca de pacotes em uma rede de computadores. Por exemplo, Maxion (1990) analisou fluxos de pacotes em redes de computadores visando o diagnóstico de problemas na entrega dos pacotes. Buscando padrões em vários dias e horários da semana, ele detectou que as taxas de fluxo que podem ser consideradas normais em certos horários e dias da semana, passam a ser anormais quando acontecem em outros dias e horários. Portanto, ele identificou que não seria possível uma representação única de padrões normal e anormal para a rede tratada e que de fato outras informações precisam ser avaliadas para que situações anormais pudessem ser identificadas.

Nesta primeira etapa, deve-se definir as variáveis observáveis, os escopos, as zonas de perigo para cada variável e cada escopo. Supondo duas variáveis temperatura (tipo numérico) e estação do ano (primavera, verão, outono, inverno), um alerta de perigo pode ocorrer quando uma temperatura excede um valor de limiar esperado para ela naquela estação do ano.

3.2. Segunda etapa : modelagem multiagentes

Um sistema multiagentes é composto por um conjunto de agentes que encapsulam as funcionalidades do sistema, e um ambiente onde os agentes estão situados. Os agentes possuem habilidades de interação e organização, que permitem que resolvam problemas de forma coletiva, seja cooperando ou colaborando entre si.

Diversas metodologias já foram desenvolvidas para a concepção de sistemas multiagentes. Este tema não será desenvolvido neste artigo. As metodologias têm em comum a perspectiva de um sistema multiagentes deve ser composto por quatro elementos: agentes, ambiente, interação e organização. Os agentes constituem os componentes principais do sistema, sendo responsáveis pela modelagem do conhecimento do sistema, bem como pela tomada de decisão que em geral deve emergir das ações coordenadas dos agentes. O ambiente compreende o contexto ou o escopo onde os agentes se situam. As interações permitem que os agentes se comuniquem e desta maneira resolvam problemas coletivamente. A organização é um fator relevante, pois estabelece um modelo de funcionamento do sistema, em geral com uma inspiração social ou biológica. De fato, cada aplicação irá requerer uma configuração multiagentes particular.

O modelo de diagnóstico proposto integra características dos sistemas multiagentes e do modelo imunológico da Teoria do Perigo. Sendo assim, a modelagem do sistema inclui:

- agentes com funcionalidades variadas para varredura, detecção de situações anormais e sinalização de perigo;
- ambiente com características da aplicação fim e do próprio modelo imunológico artificial;
- protocolos de interação entre agentes que correspondem ao processo coletivo de diagnóstico;

- o modelo de organização é uma adequação dos princípios da Teoria do Perigo.

Como resultado desta etapa tem-se a definição dos componentes do sistema, suas funcionalidades e modelo de interação e organização dos agentes em um processo de diagnóstico.

3.3. Terceira etapa : implementação, testes e avaliação

A etapa da implementação pode ser realizada com o auxílio de uma plataforma multiagentes ou qualquer linguagem de programação. Em nossos experimentos já trabalhamos com as plataformas JADE, FIPA-OS, e JATLite, além de desenvolver sistemas multiagentes mais simples fazendo uso da linguagem Java e pacotes de comunicação para trocas de mensagens. A escolha da linguagem e da plataforma está principalmente associada à plataforma onde o sistema de diagnóstico irá atuar, aos componentes de software e hardware que necessitará acessar, as demandas do usuário do sistema, bem como da necessidade de integração com sistemas legados. Havendo hoje uma grande variedade de linguagens e plataformas de trabalho, não nos estenderemos neste aspecto da metodologia, deixando tal tarefa a cargo do desenvolvedor do sistema.

Os testes preliminares do sistema de diagnóstico podem ser feitos considerando-se cenários, simulações de situações e usuários fictícios. Tais testes servirão para avaliar o comportamento do sistema e coletar dados de diagnósticos produzidos em situações de uso simulado. A realização dos testes pode ser feita seguindo-se protocolos de uso que correspondam ao uso real, valendo-se de dados reais previamente coletados. Os cenários de teste devem ser suficientemente variados a fim de prever sempre que possível as diversas situações sobre as quais o sistema de diagnóstico deverá atuar. Os especialistas do domínio são as pessoas mais capacitadas para definir tais cenários, tendo papel importante em todas as etapas do desenvolvimento do sistema de diagnóstico.

Para a avaliação dos resultados obtidos pelo diagnóstico do sistema, recomenda-se o uso de comparações entre diagnósticos de um ou mais especialistas e do sistema. Deve-se levar em conta que a taxa de convergência dos diagnósticos elaborados por diferentes especialistas, o que torna muitas vezes a tarefa de validação do sistema complexa. Mesmo assim deve ser possível estabelecer uma margem de acerto do sistema. Nas comparações em geral pode-se observar situações de convergência total (entre o sistema e os especialistas), convergência parcial (o sistema converge com pelo menos um especialista) e divergências (o diagnóstico do sistema diverge do diagnóstico convergente dos especialistas, o diagnóstico do sistema e os especialistas divergem todos entre si). A margem de erro para aceitação do sistema de diagnóstico depende do domínio de aplicação, da sua complexidade e precisão de diagnóstico necessária.

4. Sistema Multiagentes de Diagnóstico

Seguindo a abordagem metodológica descrita na seção 3, implementou-se um sistema multiagentes de diagnóstico do aluno integrado a um ambiente de aprendizagem na área de programação de computadores. A metodologia imune se mostrou adaptada ao processo de diagnóstico do aluno, pois identificou-se diversas similaridades entre o conceito de sinal de perigo do SIN e o sinal de erro na aprendizagem que um ambiente de ensino deve emitir.

4.1. Sistema Multiagentes

O sistema multiagentes desenvolvido é composto por três classes de agentes (Webber, 2008). Primeiramente, o agente macrófago analisa e decompõe a string de entrada (algoritmo do aluno). Agentes semanticamente distribuídos fazem o papel de linfócitos e são responsáveis por procurar por padrões de perigo na string decomposta, emitindo sinal de perigo quando necessário. Agentes de diagnóstico coletam os sinais de perigo e agrupam aqueles que forem relacionados. Este processo de diagnóstico acontece em 4 etapas: 1) fagocitose, 2) resposta imune adaptativa, 3) sensibilização dos linfócitos B, e 4) diagnóstico final. Estas etapas são descritas a seguir.

Etapa 1 - Fagocitose

Esta primeira etapa corresponde à resposta imune inata observada no sistema imunológico humano, que provê defesa imediata. O primeiro agente inicializado é o macrófago. Este agente fagocita o antígeno representado pelo algoritmo do aluno e o divide em peptídeos (tokens da linguagem) para os linfócitos T. Ocorre então a criação de uma nova estrutura de dados, onde o algoritmo é separado e classificado pelos tokens contidos nele. Destes peptídeos separados, os únicos que sensibilizam os linfócitos T são os peptídeos que representam a linguagem algorítmica, como ``se-então", ``para", ``enquanto", ``repita-até", ``caso", ``<-" (atribuição), ``declare" e ``escreva". Estes peptídeos ficam na superfície do macrófago, prontos para serrem capturados por uma célula T.

Etapa 2 - Resposta imune adaptativa

Nesta etapa, os linfócitos T vasculham a superfície do macrófago buscando peptídeos com afinidade aos seus receptores. Estes receptores reconhecem os seguintes peptídeos: ``caso", ``enquanto", ``escreva", ``declare", ``para", ``repita-até", ``se-então" e ``<-" (atribuição). Quando o linfócito T reconhece um destes peptídeos, ele ativa o agente semântico correspondente. Se o agente semântico reconhecer um peptídeo com erro, então o linfócito T secreta uma citocina no ambiente. No SIN, citocina é uma proteína que afeta o comportamento de outras células sensíveis a ela. Uma citocina, para o SDA, é uma mensagem trocada entre os agentes. As mensagens utilizam uma arquitetura do tipo quadro negro para se comunicar. As mensagens deixadas no quadro negro estão disponíveis para todos os agentes.

Dependendo do tipo de erro encontrado, um tipo de citocina é secretado. Abaixo estão relacionados os tipos de citocina e seus respectivos comandos:

- expressão: erros de expressões nos comandos : *caso*, *enquanto*, *escreva*, *para*, *repita-até* e *se-então*;
- inicialização: erros de inicialização de variáveis;
- verificação de contadores: erro no teste de contados em laços de repetição;
- tipo: erros de atribuição entre tipos (<-);
- unicidade: erros de variáveis duplamente declaradas;
- declaração: erros de variáveis não declaradas.

A mensagem que representa a citocina contém 3 informações: o nome do agente que secretou a citocina, o tipo da citocina e o resultado ou a descrição do erro encontrado. Quando os linfócitos T concluem a análise de todos os peptídeos disponíveis na superfície do macrófago, eles secretam uma citocina do tipo “alerta”, que irá indicar

aos demais agentes do SIA o fim do antígeno no sistema (todos os peptídeos que compõem o antígeno foram analisados). Ao detectar o fim do antígeno, os linfócitos T encerram suas tarefas.

Etapa 3 - Sensibilização dos linfócitos B comuns

Existem 4 agentes do tipo linfócito B comum. Cada um é sensível a apenas um tipo de citocina secretada pelos linfócitos T (expressão, tipo, unicidade ou erro de declaração), além da citocina de alerta. Quando o linfócito B comum encontra uma citocina que o sensibilize, ele armazena o resultado desta citocina e continua a monitorar o ambiente, até que encontre a citocina de ``alerta", que indica o fim do antígeno. Quando o alerta é encontrado, o linfócito B comum secreta uma citocina do tipo “diagnostico” no ambiente. Esta citocina é acompanhada do resultado de todas as citocinas recolhidas anteriormente pelo agente. Desta forma, os agentes conseguem formar grupos de erros relacionados a cada categoria. Depois deste processo o linfócito B comum pode entrar em estado de repouso, voltando mais tarde a verificar o ambiente. O agente repete este processo de transição de estado algumas vezes e depois encerra as suas atividades.

Etapa 4 - Diagnóstico final

Este último passo conclui a tarefa de diagnóstico. A citocina de diagnóstico que foi secretada no ambiente sensibiliza o agente linfócito B de diagnóstico. Ao encontrar uma citocina de diagnóstico, o linfócito B de diagnóstico armazena seu conteúdo e continua a procurar outras citocinas de diagnóstico. Se ele encontrar uma citocina de alerta, ele pára de vasculhar o ambiente. Ao final da análise do antígeno, o agente linfócito B de diagnóstico terá o resultado global do trabalho de todos os agentes sobre suas respectivas partes do domínio. O agente Linfócito B de diagnóstico captura as mensagens parciais de diagnóstico, organiza e integra estas mensagens, para construir um diagnóstico final. O diagnóstico final compreende erros sintáticos e semânticos, agrupados em classes de erros.

4.2. Cenário Ilustrativo de Diagnóstico

Esta seção apresenta um processo de diagnóstico do aluno. Considere que a aprendizagem de programação é organizada em três áreas de conhecimento (AC) progressivas (área 1, área 2 e área 3) descritas na tabela 1.

Tabela 1 - Descrição das áreas de conhecimento

AC 1	AC 2	AC 3
Variáveis: declaração e tipos Expressões Operador de atribuição Algoritmos sequenciais Comandos Condicionais (se-então, se-então-senão, caso) Instruções de entrada e saída	Comandos de repetição (para, repita, enquanto) Condições de parada: - No início do bloco (enquanto) - No final do bloco (repita) Fluxo de controle Comandos aninhados	Estruturas de dados: - Vetores - Matrizes Manipulação de Índices

Para a tarefa de diagnóstico do aluno, estabeleceu-se que as três ACs correspondem à três zonas de perigo. Cada AC delimita uma zona de perigo onde os observáveis para o diagnóstico possuem pesos distintos à medida que o aluno avança em sua aprendizagem. Para que o acompanhamento da evolução da aprendizagem possa ocorrer, associou-se a cada AC uma variável de limiar (t) que corresponde a um número

a partir do qual o sinal de perigo será disparado. Ele corresponde ao número máximo de erros que podem ser encontrados na resolução de um problema para que o sistema emita um alerta de perigo, o que sinaliza problemas de aprendizagem. Este limiar deve ser configurado pelo professor antes do uso do sistema, podendo ser alterado posteriormente. A tabela 2 permite visualizar, para cada situação possível, como erros são convertidos em sinal de perigo; *ns* indica que não há sinal de perigo e *na* indica que nenhum sinal se aplica para esta área.

Tabela 2. Sinal de perigo em cada área de conhecimento

	AC 1		**AC 2**		**AC 3**	
AC corrente do aluno	$< t_{AC1}$	$>= t_{AC1}$	$< t_{AC2}$	$>= t_{AC2}$	$< t_{AC3}$	$>= t_{AC3}$
1	*fraco*	*forte*	*na*		*na*	
2	*ns*	*fraco*	*fraco*	*forte*	*na*	
3	*ns*	*fraco*	*ns*	*fraco*	*fraco*	*forte*

Para compreender a tabela 2, suponha um aluno no início da disciplina de algoritmos (primeira área de conhecimento). Durante a resolução de problemas é normal que o aluno erre e corrija seus erros várias vezes. Cada erro na AC 1 gera um sinal de perigo fraco. Se a quantidade de erros exceder ou igualar o limiar estabelecido t_{AC1}, então um sinal de perigo forte será disparado.

Quando o aluno alcança a segunda área de conhecimento, os erros da primeira área de conhecimento somente geram um sinal de perigo fraco se excedem ou igualam o valor do limiar t_{AC1} . Dessa forma prevê-se que o aluno poderia, por esquecimento ou erro de digitação, cometer poucos erros de conceitos ou comandos estudados na primeira área. Considerando que a meta de aprendizagem corrente é a segunda área, assume-se para efeitos de diagnóstico que erros da primeira AC somente indicariam que o aluno precisa revisar conceitos já estudados e corrigir a sua solução. Para erros da segunda AC, o tratamento segue o mesmo conceito da primeira AC, visto anteriormente. Se o aluno cometer uma quantidade de erros que não atinja o limiar t_{AC2}, um sinal de perigo fraco será disparado. Porém, se o aluno comete um número de erros maior ou igual ao que o limiar t_{AC2} permite, uma sinal de perigo forte para a segunda área de conhecimento deve ser disparado. O mesmo ocorre para o tratamento da terceira área de conhecimento.

4.3. Interface de Testes

A fim de realizar testes de diagnóstico desenvolveu-se uma interface simplificada (SIA-TP) onde é possível inserir soluções produzidas pelos alunos e observar o comportamento dos agentes e obter o diagnóstico. Para este estudo foram analisadas 200 resoluções de alunos cursando a disciplina introdutória de programação no ano de 2008. Solicitou-se aos alunos que resolvessem alguns problemas usando papel e lápis. Posteriormente, estas soluções foram inseridas através da interface de resolução de problemas (figura 1): o aluno digita o algoritmo e solicita o diagnóstico, podendo observar as mensagens enviadas entre os agentes e o resultado do diagnóstico. Na parte inferior a direita, três botões sinalizam situações de perigo (área 1, área 2, área 3) de acordo com uma escala de cores. A cor cinza indica que não há sinal de perigo; amarelo indica sinal fraco de perigo; e o vermelho indica sinal forte de perigo.

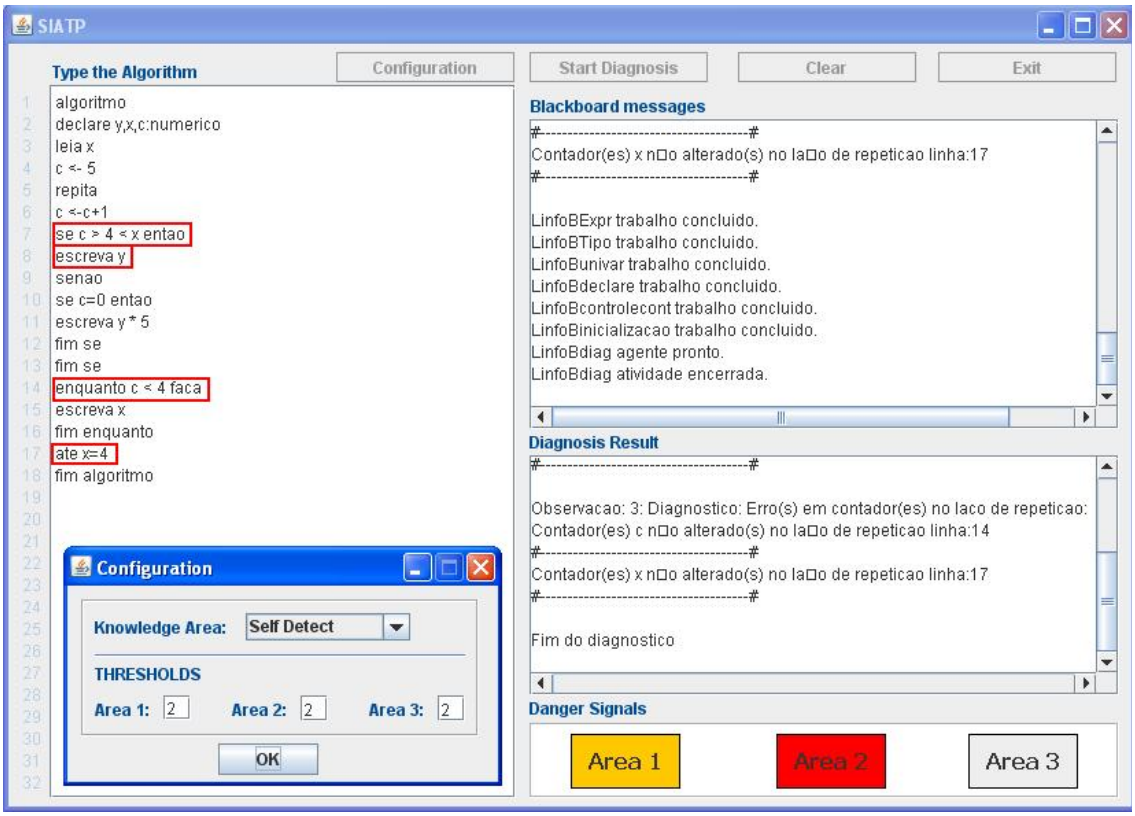

Figura 1 – Teste do algoritmo de um aluno na interface do SIA-TP

Para obter um diagnóstico apropriado, deve-se previamente configurar a área de conhecimento e o valores de limiar para cada área (janela *configuration*). Na opção *self-detect* o SIA-TP extrai a área de conhecimento de cada algoritmo pois detecta quais os comandos e estruturas de dados empregadas pelo estudante.

4.4. Análise do exemplo de diagnóstico do aluno

A fim de ilustrar um processo de diagnóstico, considere o mesmo algoritmo construído por um aluno (figura 1). Para a análise deste exemplo selecionou-se a opção *self-detect* (auto-detectar). O sistema identificou a solução do aluno como pertencendo a problemas da segunda área de conhecimento (área 2). No diagnóstico o SDA identificou 4 erros:

- expressão inválida no comando *se*, linha 7 (*área de conhecimento* 1);
- variável y não foi inicializada, linha 8 (*área de conhecimento* 1);
- variável c não muda de valor no laço *enquanto*, linha 14 (*área de conhecimento* 2);
- variável x não muda de valor no laço *repita*, linha 17 (*área de conhecimento* 2).

A tabela 3 sintetiza os resultados do SDA na avaliação do aluno neste exemplo.

Tabela 3 – Sinais de perigo para o exemplo de diagnóstico

Área de conhecimento do aluno	AC	t	Erros	Sinal de Perigo
2	1	2	2	Fraco
	2	2	2	Forte
	3	2	0	Nenhum

Sinais de perigo são utilizados para alertar o aluno e o seu professor sobre o processo de aprendizagem. Observe que o número de erros na primeira e segunda áreas

atinge os valores de limiar (t) e, portanto, sinais de perigo devem ser ativados. Para a primeira área será ativado um sinal fraco, e para a segunda um sinal forte.

5. Conclusão

Este artigo apresentou uma abordagem de diagnóstico baseada na teoria imunológica do perigo. Segundo esta teoria, somente quando um tecido se encontra ameaçado ou em situação de estresse é que uma resposta imunológica irá se produzir. Este modelo traz como vantagem a possibilidade de se lidar de forma diferente com situações anormais que podem ou não caracterizar situações de perigo ao longo da vida de um sistema.

Como pesquisadores em IA, buscamos em outros campos do conhecimento métodos que se apliquem a resolução de problemas. Os sistemas imunológicos constituem uma inspiração a partir da qual diversos métodos computacionais têm sido desenvolvidos. A sua aplicação na área do diagnóstico tem se mostrado promissora, embora um estudo comparativo ainda precise ser desenvolvido. Neste sentido, protocolos para teste se encontram em desenvolvimento e deverão ser aplicados em breve. Além da validação deste modelo, eles irão buscar posicionar esta abordagem em relação às já tradicionais abordagens de diagnóstico.

References

Aickelin, U., Cayser, S. (2002) “The danger theory and its application to AIS”. In Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002), pp.141-148.

Burnet F.M, (1959) “The Clonal Selection Theory of Acquired Immunity”, Cambridge Univ. Press.

Console, L., Theseider Dupré, D., Torasso, P. (1989) “A theory of diagnosis for incomplete causal models”, Proceedings of the 10th IJCAI, USA, pp.1311-1317.

Dasgupta, D. (2006) Advances in Artificial Immune Systems. IEEE Computational Intelligence Magazine. pp.40-49.

De Castro, L. N. & Von Zuben, F. J. (1999) "Artificial Immune Systems: Part I – Basic Theory and Application", Tech.Report, Unicamp 01/1999, p.65.

De Castro, L. N. & Von Zuben, F. J. (2000) "Artificial Immune Systems: Part II – A Survey of Applications", Tech.Report, Unicamp 02/2000, p.65.

Farmer, J., Packard, N., Perelson, A. (1986) The immune system, adaptation and machine learning. Physica D, 22, pp.187-204.

Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.(1994) “Self-Nonself Discrimination in a Computer”, In Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Computer Society Press.

Frohlich, P. M'ora, I.A., Nejdl, W., Schroeder, M. (1997) “Diagnostic agents for distributed systems”, Proceedings of ModelAge97, Sienna, Italy.

Fukuda, T., Mori, K., Tsukiyama, M. (1993) Immune networks using genetic algorithm for adaptive production scheduling. In : Proceeding of the 15th IFAC World Congress, volume 3, pp.57-60.

Janeway, C., Travers, P., Capra, J.D., Walport, M.J. (2000) Immunobiology: The Immune System in Health and Disease, Garland Pub, pp.635.

Matzinger, P. (2002) The Danger Model: A renewed sense of self. Science, 296 (5566), pp.301-305.

Maxion, R.A. (1990) Toward diagnosis as an emergent behavior in a network ecosystem. In:. Physica D 42, pp.66–84.

Reiter, R.(1987)“A theory of diagnosis from first principles”,*Artificial intelligence* , 32 (1), pp.57-96.

Ross, N., Teije, A., Witteveen, C. (2003) “A Protocol for Multi-Agent Diagnosis with Spatially Distributed Knowledge”, Austrália, *ACM Press*, pp.655-661.

Webber, C.G., Silva, J.L.T. (2008) “Self and Non-self Discrimination Agents”, In Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08), ACM, New York, 1987-1988.

Weinand, R.G. (1990) Somatic mutation, affinity maturation and antibody repertoire: A computer model. Journal of the Theoretical Biology, 143, pp.343-382.