

AWARE – um ambiente para estudo e visualização de processos adaptativos

I.S. Oliveira, J.E. Kogler Jr, *Senior Member, IEEE* e E. Del Moral Hernandez, *Senior Member, IEEE*

Abstract— this paper presents AWARE, a software tool developed to aid research in the field of adaptive agents, consisting of a graphical user interface for description and execution of adaptive finite state machines (AFSM), and that enables the visualization of the topological changes produced by the application of the adaptive functions. It includes a simulator of a virtual agent behavior embedded in a virtual environment, under the control of the specified AFSM, which helps to understand the effects of the adaptation strategies before applying it to a real agent in a physical world. It also includes a module to handle physical robotic agents, by receiving their sensorial inputs and directing them the control actions resulting of the application of the controller rules, generated by the AFSM simulator.

Keywords— adaptive devices, modeling, simulation, virtual agents, robotic agents, software tools.

I. INTRODUÇÃO

ESTE trabalho apresenta uma ferramenta de software, à qual demos o nome AWARE (de *Adaptive SoftWARE*), voltada ao estudo da adaptatividade em sistemas de agentes robóticos imersos em ambientes dinâmicos, permitindo a monitoração dos comportamentos dos agentes e dos processos que os controlam. Sua principal característica consiste em permitir a visualização e documentação das mudanças estruturais decorrentes de adaptações no comportamento e controle dos agentes. O modelo de controle explorado baseia-se particularmente em um dispositivo formal adaptativo dirigido por regras. Na presente versão as regras são pré-especificadas em correspondência com as situações potencialmente causadoras de modificações. Todavia, sua formulação foi concebida de modo a poderem ser construídas através de outras regras adaptativas hierarquicamente mais abstratas e que possam ser expressas via métodos de decisão formais (traduzindo uma adaptatividade formal de ordem mais elevada) ou por decisões decorrentes de procedimentos de inferência indutiva tais como aprendizagem estatística e neurocomputação, entre outros.

Os agentes robóticos considerados são plataformas fisicamente constituídas, dotadas de hardware diretamente controlado pelo AWARE. A ferramenta aqui apresentada oferece também recursos para simular tais agentes de forma minimalista, dispondo de uma interface gráfica pela qual o

comportamento estimado do agente pode ser visualizado em correspondência com a apresentação visual das modificações em sua estrutura de controle acarretadas pela ação das adaptações. Isso permite que se estude passo a passo o processo de adaptatividade antes de aplicá-lo diretamente ao agente físico, possibilitando evitar perigos físicos ou prever situações de complexidade maior que requeiram acompanhamento mais cuidadoso. A figura 1 ilustra essas características da ferramenta.

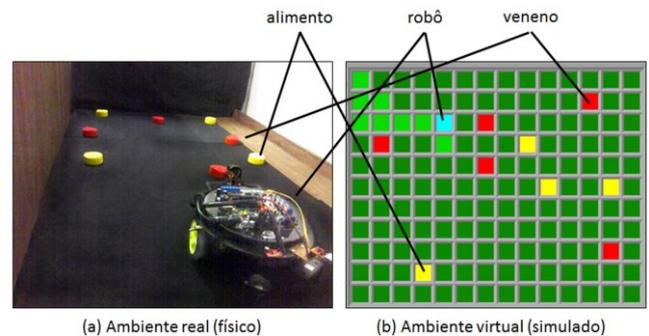


Figura 1. Agente robótico imerso no ambiente – (a) real; (b) virtual. (Note-se, todavia, que não são situações exatamente idênticas, mas são similares).

Na figura 1 apresentam-se lado a lado duas situações, uma em que um robô físico está imerso no ambiente real e, na outra um robô virtual encontra-se em um ambiente virtual similar. Ambos ambientes são dotados de dois tipos de objetos, que diferem apenas em duas propriedades: uma delas é diretamente observável (cor); a outra se manifesta pelo seu efeito sobre o agente (alimento versus veneno), que deve ser inferida. No ambiente virtual, o robô é abstraído por um quadrado azul e sua trajetória é indicada pelas posições que ocupou anteriormente, as quais estão marcadas com um tom verde mais claro que o do fundo. O AWARE proporciona através desse esquema um ambiente virtual de experimentação prévia, permitindo testar hipóteses em situações em que perturbações físicas (atrito, erros de posicionamento, incertezas sensoriais) não precisem ser levadas em consideração.

O ambiente de software que constitui o AWARE será apresentado nas próximas seções, mostrando-se sua concepção e os seus atuais componentes. Entretanto, esta ferramenta não se resume a sua atual realização, pois pode ser modificada para acomodar aplicações de maior complexidade, mantendo-se sua parte essencial que é a arquitetura de controle subjacente. Ela suporta quaisquer dispositivos adaptativos dirigidos por um conjunto finito de regras com formulação determinística. A estrutura básica do simulador do ambiente virtual também pode ser mudada sem modificar sua essência. E, quanto ao

I.S. Oliveira, Universidade de São Paulo, São Paulo, Brasil, ian.oliveira@usp.br

J.E. Kogler Jr., Universidade de São Paulo, São Paulo, Brasil, joao.kogler@ieee.org

E. Del Moral Hernandez, Universidade de São Paulo, São Paulo, Brasil, emilio_del_moral@ieee.org

agente físico, também é possível lidar com robôs ou equipamentos mais sofisticados, sem que para tanto seja necessário realizar mudanças fundamentais na arquitetura.

Discutiremos na próxima seção a concepção e a especificação do AWARE e aspectos de sua realização. Na seção seguinte apresentaremos a ferramenta tal como se encontra na versão atual e a seguir daremos alguns exemplos de testes do conceito. Finalmente concluiremos, discutindo as perspectivas de futuras versões e quanto à sua distribuição.

II. CONCEPÇÃO E ESPECIFICAÇÃO DA FERRAMENTA AWARE

A necessidade de uma ferramenta como o AWARE surgiu naturalmente como meio de se controlar e monitorar o comportamento de agentes cognitivos. Essencialmente um agente é uma abstração que se concentra nos mecanismos de produção de ações em resposta a estímulos. Agentes podem ser empregados como modelos para estudar-se o comportamento de organismos naturais (biológicos) e são úteis na produção de teorias que visem prever as reações de criaturas vivas submetidas a determinadas situações. A validade de cada teoria decorrerá de sua capacidade de acertar as previsões sobre as observações obtidas em situações experimentais. Agentes também podem ser utilizados para especificar organismos artificiais (robôs, máquinas e sistemas automáticos), que se comportem de modo análogo aos seres vivos, isto é, seguindo o princípio de associar as ações que produzem a respostas a determinadas entradas que recebem como estímulos.

Há diversas categorias de agentes, de acordo com sua capacidade de utilizar sua experiência pretérita. Há os que são puramente reativos e dotados de regras fixas que associam cada possível estímulo a uma resposta correspondente. Há também aqueles cujas regras contêm parâmetros que podem ser modificados de acordo com os estímulos, conferindo uma forma elementar de adaptação que, entretanto, não modifica a forma da regra. Há os que podem inferir regras novas a partir dos estímulos e/ou de regras anteriores, e os que usam conhecimento anterior para produzir as novas regras. Estes dois últimos tipos constituem a classe dos agentes cognitivos. Pode-se então conceber agentes cujo controle baseia-se em ingredientes de maior ou menor complexidade no mecanismo de elaboração das regras que definem seu comportamento, sendo estas dependentes ou não em relação aos estímulos e ao conhecimento anterior. Em resumo, a essência da ideia de agente está na forma de associação entre as suas ações e os estímulos que as precedem.

Há uma classe muito variada de modelos que descrevem essas associações, e entre eles encontram-se os modelos formais, que se caracterizam como dispositivos dirigidos por regras. É de grande interesse que se possam utilizar modelos formais, beneficiando-se de sua manipulação através de métodos algébricos e dedutivos, o que facilita sua construção, modificação e verificação. No caso de se trabalhar com agentes cognitivos, a dificuldade na formalização decorre de sua natureza intrinsecamente flexível, de poderem adaptar seus comportamentos em função do ambiente em que estão imersos,

o que se traduziria, em termos de modelos formais, em sistemas de regras que permitam modificar-se de acordo com as circunstâncias. É nesse cenário que os dispositivos adaptativos são de grande valia.

A introdução do conceito de dispositivos formais adaptativos como modelos de sistemas capazes de se alterarem dinamicamente em resposta espontânea aos estímulos de entrada a que estão submetidos [1], [2], trouxe consigo novas possibilidades na investigação da aprendizagem por agentes.

Trabalhar com tais modelos baseados em dispositivos adaptativos constitui um grande desafio, todavia. Usualmente a especificação de um modelo cresce em complexidade na medida em que se empregam construtos elementares mais simples. O arcabouço das teorias baseadas em métodos mais formais constitui-se de elementos construtivos geralmente muito simples, como é o caso das teorias de computação: tendem a basearem-se em operações elementares. A expressão de funcionalidades nesses modelos provém da estrutura que envolve essas operações elementares. No caso dos dispositivos adaptativos, o agravante extra de complexidade resulta da adaptatividade, que se expressa através de regras que permitem modificar a estrutura que traduz cada funcionalidade, isto é, da topologia que caracteriza as relações entre os estímulos e as respostas. Lidar com esse tipo de modelo eficientemente requer ferramentas que auxiliem a apreciação dos detalhes desse sistema complexo, das mudanças topológicas que essas adaptações acarretam. Tendo-se isso em consideração, decidimos que, antes de iniciar uma linha de investigação em agentes capazes de aprender empregando modelos formais, seria importante investir na busca de uma ferramenta de análise que facilitasse lidar com os dispositivos adaptativos, que permitisse o acompanhamento de suas modificações na medida em que se adaptam, colocando-as em correspondência com as modificações no comportamento do agente.

A especificação dessa ferramenta foi orientada para a concepção de uma plataforma básica contendo apenas as características essenciais mínimas para a realização de experimentos dessa categoria, mas de maneira tal que facilmente proporcionasse sua modificação para aplicações mais complexas. Os principais requisitos considerados nessa concepção mínima compreendem: capacidade de tratar um agente robótico controlado por um dispositivo adaptativo, imerso em um ambiente dotado de pelo menos dois tipos de objetos com os quais o agente possa interagir.

A expressão ‘agente robótico’ denota um agente autônomo dotado de sensores e efetores, capaz de alterar suas relações com o ambiente em resposta a estímulos que recebe do mesmo e sob a ação de processos internos que controlam seu estado. O agente considerado na versão atual consiste de uma plataforma robótica móvel capaz de se comunicar com um computador hospedeiro. Parte da arquitetura de controle encontra-se embarcada na plataforma, sendo esta dotada de um micro controlador destinado a gerenciar o acionamento dos motores das rodas, a aquisição de dados de sensores simples tais como acelerômetros e sensores ultrassônicos de distância e

a comunicação com o computador hospedeiro. Nomeadamente, utilizou-se para esse fim a micro arquitetura Arduino [6], embora esse detalhe não seja importante, uma vez que qualquer micro arquitetura embarcada poderia ter sido utilizada, desde que fosse capaz de lidar com todos os sinais e processos locais preconizados.

Em princípio, a emulação do dispositivo adaptativo que efetivamente controla o comportamento do agente poderia ser realizada tanto localmente no processador embarcado, quanto remotamente, em um computador hospedeiro em comunicação com o agente. É essencial que seja respeitada a temporização dos processos em resposta aos estímulos, permitindo que o comportamento do agente se desenvolva sem atrasos que possam comprometer sua interação com o ambiente e seu desempenho. No caso particular da atual implementação do agente robótico, optou-se pela distribuição do controle de modo a deixar embarcado neste apenas a parte relativa à aquisição de sinais de sensores simples e o acionamento dos efetores (rodas). Os dados sensoriais são processados remotamente e convertidos nos estímulos consumidos pelo dispositivo adaptativo. Este é emulado no computador hospedeiro e produz as saídas que serão direcionadas para o controlador embarcado e que servirão para acionar os efetores. O agente robótico foi também especificado para empregar visão computacional, necessária para permitir ao agente a identificação de situações baseada em informações visuais obtidas através do processamento de vídeo capturado com uma câmera. Esse fator reforçou a necessidade de apoiar-se em um hospedeiro remoto, haja vista que o processamento do vídeo pode ser bastante oneroso para ser executado em uma micro arquitetura embarcada.

Especificado o agente robótico e a arquitetura do sistema, que conseqüentemente tem o controle distribuído, passou-se à concepção da ferramenta de software para a simulação dos dispositivos formais adaptativos, que exercerão o controle do agente. Primeiramente buscou-se alguma ferramenta já existente com as características desejadas. Nomeadamente, tais características são:

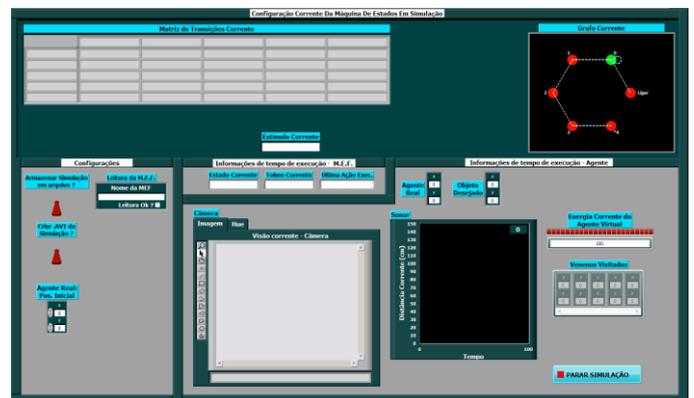
1. Capacidade de lidar com dados provenientes da aquisição de sinais de sensores analógicos de diferentes modalidades.
2. Capacidade de produzir saídas que serão convertidas para sinais analógicos que servirão para acionar efetores.
3. Ser compatível com o emprego de arquitetura de controle distribuída.

Além dessas características essenciais à configuração pretendida, a ferramenta a ser utilizada deveria oferecer flexibilidade suficiente para ser utilizável para configurações mais complexas, permitindo facilidade nas modificações e boa escalabilidade. Feita uma busca na literatura, encontrou-se apenas uma potencial ferramenta que se aproximasse do atendimento a esses requisitos. Os trabalhos de Camolesi e Neto [3] e [4], oferecem uma proposta de um metambiente, materializada por meio de uma ferramenta com interface

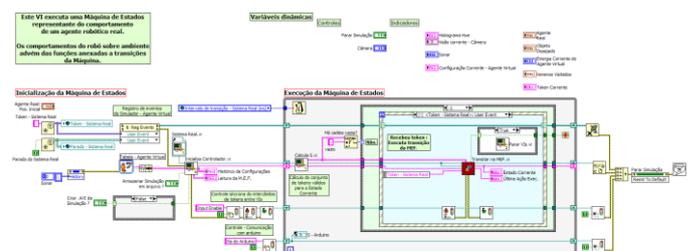
visual, destinada a oferecer ao usuário, meios para a construção automática de ambientes completos para a definição, formalização, simulação e ensaios de dispositivos formais, adaptativos ou não, especificados pelo próprio usuário [5]. Entretanto, tais trabalhos concentraram-se na definição e formalização do metambiente e não pareceram atender a todos os requisitos propostos. Conseqüentemente, decidimos por desenvolver uma ferramenta própria, adequada aos fins propostos.

III. CARACTERÍSTICAS DA FERRAMENTA DE SOFTWARE

Para se desenvolver rapidamente uma plataforma suficientemente flexível para ser aplicável a configurações experimentais bastante variadas, satisfazendo aos requisitos especificados optou-se pelo emprego do LabVIEW® , recurso disponível e adequado. O LabVIEW [7] consiste em um ambiente de desenvolvimento de programação visual e uma linguagem visual orientada pelo fluxo de dados. A linguagem do LabVIEW oferece expressividade suficiente para alcançar a flexibilidade requerida em lidar com a variedade de situações experimentais que se tem em mente e dispõe de uma biblioteca suficientemente rica de funções e construtos mais elaborados que, juntamente com os recursos de seu ambiente de desenvolvimento, permitem que se construa a ferramenta desejada em um prazo razoável. A figura 2 apresenta a interface gráfica do AWARE e o código visual em LabVIEW, o que oferece uma boa ideia de suas características.



(a)



(b)

Figura 2. Características do AWARE – (a) interface gráfica de usuário - GUI (painel frontal); (b) código do programa na linguagem visual do LabVIEW® (diagrama de blocos).

Na fig.2a pode-se apreciar os elementos essenciais do AWARE, que se encontram na parte superior do painel e consistem na matriz e no grafo que descrevem o dispositivo adaptativo representado por uma máquina de estados finitos. As linhas e colunas da matriz são indexadas, respectivamente, pelos estados de origem e de destino ligados pelas transições, e suas células contêm as regras na forma $ba / cj / s / aa$, denotando, respectivamente, a função adaptativa aplicada antes da ação (ba), o *token* corrente (cj), o estado atual (s) e a função adaptativa aplicada depois da ação (aa), seguindo a notação em [2]. A figura 3 apresenta um exemplo da matriz e do grafo correspondente em duas situações entre o consumo de um token e a aplicação de uma ação adaptativa de inserção.

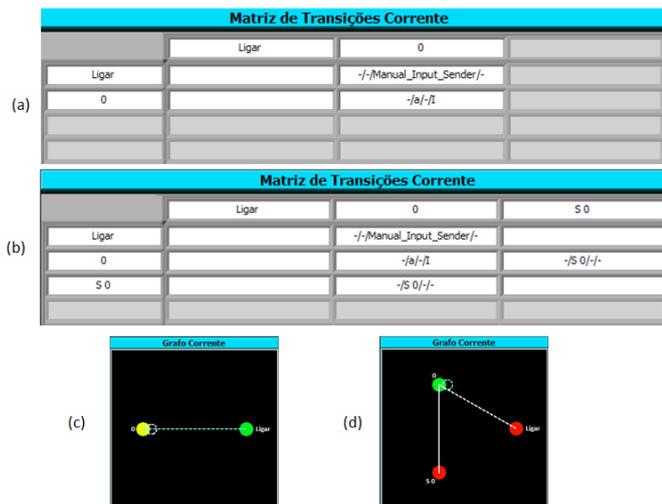


Figura 3. (a) e (c) – matriz e grafo para MEF com 2 estados: em verde o estado de origem e em amarelo o destino; (b) e (d) – MEF com 3 estados.

Nas figs. 3a e 3c apresentam-se a matriz e o grafo correspondentes a uma máquina de estados finitos (MEF) tal que, após consumir o token ‘a’, é aplicada uma função adaptativa de inserção, resultando na MEF indicada nas figs.3b e 3d. Nesta última, o círculo verde indica o estado atual e os vermelhos os demais estados. As linhas tracejadas indicam transições unidirecionais e as cheias transições bidirecionais. As linhas em cor ciano indicam a transição corrente.

A interface gráfica mostrada na fig.2ª apresenta também outras janelas, utilizadas para o monitoramento das diversas variáveis de interesse, cuja natureza depende do tipo de sensor ou efector que está sendo considerado. E também apresenta uma janela, mostrada na figura 1b, que no caso de se estar trabalhando com o agente virtual, exibe a configuração do ambiente virtual. Note-se que neste caso não há sensores a se monitorizar, portanto seu espaço no painel é ocupado por essa visualização do ambiente virtual.

IV. CONCLUSÕES.

O AWARE foi desenvolvido em um prazo de aproximadamente onze meses, dentro de um programa de iniciação científica. Os autores agradecem à Associação de Ex-

Alunos da Escola Politécnica da Universidade de São Paulo (AEP) pelo suporte financeiro. Diversos testes foram implementados para funções adaptativas de todos os tipos (inserção, remoção, consulta) para diversos exemplos de [2] e outros adicionais, como prova de conceito. Após estes, foram conduzidos mais testes com o ambiente virtual e alguns com o agente robótico físico, todos bem sucedidos. A continuidade do trabalho visa aplicar agora o AWARE à investigação de processos de segunda ordem, que deverão apoiar-se em inferências indutivas de natureza estatística realizadas sobre os resultados das interações entre o agente e os componentes do ambiente.

REFERENCIAS

- [1] Neto, J.J. (2002). Adaptive rule-driven devices-general formulation and case study. (B. Watson & D. Wood, Eds.)Implementation and Application of Automata, 2494, 466–470.
- [2] Ramos, M.V.M., Neto, J.J., Vega, I.S., Linguagens Formais, Bookman, São Paulo, 2009.
- [3] Camolesi, A.R., Neto, J.J. *Modelagem Adaptativa de Aplicações Complexas*. XXX Conferencia Latinoamericana de Informática – CLEI'04. Arequipa - Peru, Setiembre 27 - Octubre 1, 2004.
- [4] Camolesi, A.R. *Proposta de um Gerador de Ambientes para Modelagem de Aplicações Usando Tecnologia Adaptativa* – Tese de Doutorado, Escola Politécnica da USP, São Paulo, 2007.
- [5] Neto, J.J. Um Levantamento da Evolução da Adaptatividade e das Tecnologias Adaptativas. IEEE LATIN AMERICA TRANSACTIONS, v. 5, n. 7, pp. 496-505, NOV. 2007
- [6] Margolis, M. – Arduino Cookbook – O’Reilly, 2011
- [7] Travis, J., Kring, J. LabVIEW for Everyone, Prntice-Hall, Upper Saddle River, NJ, 3a ed., 2007.



Ian Silva Oliveira is undergrad student of Computer Engineering at Polytechnic School of Engineering of the University of Sao Paulo. His current research interests are applications of computational intelligence techniques and computer theoretic models to adaptive aspects of intelligent agents operating in changing environments.



João Eduardo Kogler Jr. is currently researcher and lecturer at Polytechnic School of University of Sao Paulo (USP). He received his PhD in computer vision and image processing and his MSc in biomedical engineering from USP. He is graduated in electrical engineering and physics respectively at Maua Institute of Technology and USP. He was a visiting scientist at Siemens Corporate Research Lab of Princeton, NJ and of the INRIA Sophia Antipolis, France. His current research interests are in computational intelligence, adaptive computational devices, cognitive robotics and computational models of cognition and perception..



Emilio Del Moral Hernandez is currently associated professor at Polytechnic School of University of Sao Paulo (USP). He received his PhD and MSc on electrical engineering at University of Pennsylvania, PA. He is graduated and received his MSc in electronic engineering at Polytechnic School of USP. He has research collaborations with several international institutions and universities, including the CINVESTAV-Mexico Institute, the Catholic University of Peru - PUC-Lima, the Universidad Autónoma de Barcelona and the Universidad Complutense de Madrid. His current research interests are theory and applications of neural networks, bio-inspired computational systems, electronic implementation of neural models, data mining, pattern recognition, decision support systems and applications to finance engineering.