

Uso de adaptatividade na modelagem de cursos para software educacional

W. J. Dizeró e J. J. Neto

Resumo — Este artigo apresenta uma proposta de aplicação da tecnologia adaptativa na modelagem e elaboração de cursos para softwares educacionais. A princípio, soluções adaptativas podem ser incorporadas a qualquer tipo de dispositivo guiado por regras. Neste trabalho, o dispositivo subjacente utilizado será Máquina de Moore, que é um autômato finito com saídas associadas aos seus estados. Assim, para cada estado pode-se associar o material didático a ser apresentado pela função de saída. Aplicando-se os conceitos de adaptatividade nesse transdutor, é possível elaborar cursos dinâmicos, que se auto-modifiquem com base em regras definidas pelo professor, e pelas experiências e nível de conhecimento individualizado dos alunos. Para complementar o trabalho, é apresentado um exemplo baseado em Máquina de Moore Adaptativa.

Palavras chaves — tecnologia adaptativa; Máquina de Moore Adaptativa; ensino por computador.

I. INTRODUÇÃO

Nos últimos anos, inúmeras ferramentas computacionais foram propostas e desenvolvidas para ajudar no ensino nas mais diferentes áreas do conhecimento. É fato que o uso de tecnologias computacionais tem sido importante no auxílio do aprendizado para inúmeras modalidades de ensino, seja quando aplicada à educação infantil ou de nível superior, em programas regulares ou de educação continuada, no ensino presencial ou a distância.

É verdade, também, que diversos grupos de pesquisa vêm explorando o uso do computador no ensino e atingindo um significativo grau de sucesso. Porém, as metodologias para o desenvolvimento de tais sistemas costumam empregar processos de desenvolvimento informais e muitas vezes inflexíveis. Processos informais podem levar a criação de sistemas com falhas de projetado, enquanto sistemas inflexíveis podem desmotivar seu uso. Ambas as situações podem ser cruciais no sucesso de um sistema de ensino auxiliado por computador.

Neste contexto, propõe-se um modelo baseado em máquina de Moore adaptativa, com o objetivo de definir formalmente cursos e, ainda, permitir que os cursos possam ser dinâmicos. Assim, enquanto o mecanismo formal do transdutor permite que se especifique o sistema através de uma notação

matemática rigorosa, o potencial da tecnologia adaptativa [22, 23, 32] permite ao aluno uma experiência de aprendizagem individualizada, apresentando o material didático e seu roteiro de estudo adaptado ao nível de conhecimento e preferências.

Assim, a descrição deste artigo envolve, basicamente, a exposição dos conceitos acerca de máquina de Moore adaptativa, além da apresentação de um pequeno exemplo da especificação um curso baseado no modelo proposto. No capítulo 2 é feita uma breve explanação sobre o uso de computadores na educação. O capítulo 3 apresenta alguns conceitos sobre tecnologia adaptativa, a definição de dispositivo adaptativo dirigido por regras, e, apresenta o conceito de Máquina de Moore Adaptativa, cujo dispositivo subjacente guiado por regras é um transdutor com saídas associadas aos estados. No capítulo 4, encontra-se um esquema do funcionamento de um curso projetado com base no modelo proposto. Por fim, a seção 5 traz algumas considerações finais e propostas de trabalhos futuros para enriquecer este artigo. Ao final do texto, encontram-se as referências bibliográficas utilizadas.

II. COMPUTADORES NA EDUCAÇÃO

De forma geral, os computadores têm tido um papel de destaque no processo educativo. Ao contrário de outras mídias utilizadas na educação, o computador, tem a característica de processar e manipular a informação que recebe. Ele pode transformar, traduzir, usar em cálculos, ordenar, arrumar e mesmo fazer inferências a partir de uma informação fornecida.

Na educação, o computador tem sido utilizado tanto para ensinar sobre computação, como para ensinar praticamente qualquer assunto. No ensino de computação, o computador é usado como objeto de estudo. Ou seja, o aluno usa o computador para adquirir conceitos computacionais, como princípios de funcionamento do computador, noções de programação e implicações sociais do computador na sociedade. Já no ensino mediado por computador, ele pode ser usado como máquina de ensinar ou como ferramenta pedagógica.

Sendo a educação um processo de exploração, descoberta, observação e construção do conhecimento, é importante destacar que cada pessoa prefere aprender de maneira

diferente. Algumas preferem através de recursos visuais, outras através de métodos verbais, algumas preferem explorar, outras deduzir. Ou seja, qualquer ferramenta pedagógica deve considerar as diferenças individuais de cada aprendiz.

Mas, ao considerar a aplicação do computador ou qualquer produto tecnológico na educação é preciso ter claro, e em destaque, que a aprendizagem - a aquisição de um conhecimento novo - só ocorre com o engajamento pessoal do aprendiz. Nenhuma máquina é capaz de colocar conhecimento em uma pessoa. Ela pode ser usada, para ampliar as condições do aprendiz de descobrir e desenvolver suas próprias potencialidades.

Não há dúvida de que atrás de qualquer tecnologia utilizada para sistemas de ensino existem pessoas, que preparam os materiais e os disponibilizam. Assim, as tecnologias não mudam necessariamente a relação pedagógica. As tecnologias tanto servem para reforçar uma visão conservadora, como uma visão progressista. Aqui, não serão discutidos assuntos relativos a questões pedagógicas, sendo apenas apresentados conceitos e aplicações de soluções relacionadas ao uso dos computadores e da informática na educação.

III. TECNOLOGIA ADAPTATIVA

O termo tecnologia é empregado para designar o uso do conhecimento científico na resolução de problemas práticos. Para a palavra adaptatividade e outras similares, adotou-se, como definição, a propriedade que um modelo tem de alterar seu próprio comportamento, de forma espontânea, sem auxílio externo. Logo, Tecnologia Adaptativa (TA) refere-se às técnicas, métodos e disciplinas que estudam as aplicações práticas da adaptatividade.

Dessa forma, TA compreende qualquer modelo de representação formal que tenha capacidade de mudar dinamicamente seu próprio comportamento, em resposta direta a um estímulo de entrada, sem qualquer intervenção externa. Num sistema definido por um conjunto de regras, a incorporação de algum mecanismo, através do qual possa se auto-modificar durante sua execução, o torna adaptativo.

Portanto, a TA pode ser identificada em qualquer modelo que possua a propriedade de alterar seu próprio comportamento, de maneira independente, sem a interferência de qualquer agente externo. Assim, se um sistema for definido por um conjunto de regras, pode-se conferir-lhe adaptatividade pela incorporação de algum mecanismo através do qual ele possa se auto-modificar durante sua operação, alterando o próprio conjunto de regras que define seu comportamento.

De acordo com Neto [23], a principal propriedade apresentada pela Tecnologia Adaptativa consiste no fato dela constituir um formalismo com regras dinamicamente variáveis, em contraste com a maioria dos modelos formais tradicionais, cujas regras, uma vez estabelecidas, permanecem

imutáveis durante toda a sua operação. É importante destacar, também, seu poder de expressão computacional, que é equivalentemente ao das máquinas de Turing [33].

A adaptatividade pode ser incorporada a qualquer tipo de dispositivo guiado por regras. O termo dispositivo é empregado aqui como alguma abstração formal, na qual o comportamento do dispositivo é conduzido por um conjunto finito e explícito de regras. Tais regras especificam, para cada situação em que se encontra o dispositivo, sua nova situação. Quando se cria uma camada adaptativa num dispositivo subjacente qualquer, tem-se, então, um dispositivo adaptativo guiado por regras, ou simplesmente, um dispositivo adaptativo [22]. Um dispositivo adaptativo é, portanto, uma técnica adaptativa, ou seja, uma forma particular de aplicação da TA.

O formalismo geral que caracteriza os dispositivos adaptativos foi apresentado pela primeira vez à comunidade científica em [22], embora em formulações mais restritas venha sendo utilizado a mais tempo. Tal formulação fundamenta-se em um núcleo constituído por um dispositivo não-adaptativo dirigido por regras, acrescido de uma camada adaptativa que provê os recursos de mecanismos adaptativos ao mesmo. Uma característica fundamental dos dispositivos adaptativos é a possibilidade de reaproveitar integralmente os formalismos consolidados, com aumento de seu poder de representação, ao custo de um pequeno acréscimo na sua complexidade formal [32]. A teoria dos dispositivos adaptativos surgiu da busca por um formalismo simples de usar, mas capaz de representar problemas complexos, envolvendo linguagens não-regulares e até mesmo dependentes de contexto. Outras informações sobre linguagens e técnicas adaptativas, artigos relacionados e projetos desenvolvidos podem ser encontrados em [15].

Nas seções a seguir, são apresentados: uma formulação geral para dispositivos adaptativos; e, a formulação para Máquina de Moore Adaptativa.

A. Dispositivos Adaptativos Dirigidos por Regras

Historicamente, dispositivos adaptativos dirigidos por regras, ou simplesmente dispositivos adaptativos, emergem do campo de linguagens formais e autômatos. Os métodos formais permitem que se especifique, desenvolva e verifique um sistema baseado em computadores, através da aplicação de uma notação matemática rigorosa. A utilização de uma linguagem de especificação formal proporciona meios para especificar um sistema de tal modo que sejam obtidas a consistência, a completeza e a correção desejadas [1].

Um dispositivo não-adaptativo dirigido por regras, pode vir a ser qualquer máquina formal cujo comportamento dependa exclusivamente de um conjunto finito de regras, que determinem, para cada possível configuração corrente do dispositivo, a sua próxima configuração. Já, um dispositivo formal é dito adaptativo sempre que seu comportamento puder

alterar-se dinamicamente, como uma resposta espontânea aos estímulos de entrada que o alimentam. Essa alteração deve se dar sem que haja qualquer interferência de agentes externos, inclusive de usuário. Para que isso possa acontecer, os dispositivos adaptativos devem ser, portanto, automodificáveis.

Na construção de um dispositivo adaptativo dirigido por regras [22], conceitualmente é possível separar de forma clara dois componentes importantes: um dispositivo subjacente, tipicamente não-adaptativo, e um mecanismo adaptativo, responsável pela incorporação da adaptatividade.

O mecanismo adaptativo tem dois elementos: as declarações das funções adaptativas e as associações de chamadas de funções adaptativas às regras que definem o dispositivo subjacente (tais chamadas denominam-se ações adaptativas).

A primeira parte consiste de um conjunto de declarações de funções adaptativas paramétricas, as quais especificam, de uma forma semelhante ao que ocorre com as declarações de sub-rotinas e funções em uma linguagem de programação usual, as modificações a serem impostas ao conjunto de regras, na ocasião em que a função adaptativa correspondente for ativada.

Programam-se as modificações a serem impostas ao conjunto de regras através de uma lista de ações adaptativas elementares. Uma ação adaptativa elementar indica três tipos de primitivas de edição: consultas, inclusões e exclusões.

- ✓ As ações adaptativas elementares de consulta permitem inspecionar o conjunto de regras que definem o dispositivo, em busca de regras que sejam aderentes a algum um padrão fornecido.
- ✓ Ações elementares de exclusão permitem remover do conjunto de regras qualquer regra aderente ao padrão fornecido.
- ✓ Ações de inclusão permitem especificar a adição de uma nova regra, de acordo com um padrão fornecido.

Quando uma ação adaptativa é executada, todas as ações elementares de consulta e exclusão são executadas em primeiro lugar, e por último são efetuadas todas as inclusões.

A segunda parte refere-se à associação de, no máximo, duas ações adaptativas a regras selecionadas do dispositivo subjacente. Uma dessas ações adaptativas é especificada para ser executada antes que a regra à qual está associada seja aplicada. A segunda é executada após a aplicação da regra. Ambas são opcionais, portanto regras que não se estejam associadas a ações adaptativas comportam-se como simples regras não-adaptativas.

A seguir, é apresentada uma figura com o esquema geral para Dispositivos Adaptativos. Na figura, é possível ver uma

cadeia de entrada e o comportamento inicial do dispositivo. Para se consumir um símbolo de entrada da cadeia, a camada adaptativa do dispositivo é acionada, realizando, opcionalmente, ações adaptativas antes de executar as regras do dispositivo subjacente. Em seguida, são execuções as regras do próprio dispositivo subjacente. Após a execução das regras do dispositivo subjacente, ações adaptativas posteriores podem ser executadas. E, finalmente, tem-se o novo comportamento do dispositivo.



Figure 1: Esquema de um dispositivo adaptativo.

Em outros termos, um dispositivo adaptativo pode ser obtido pela incorporação de ações adaptativas às regras da formulação subjacente, de tal modo que, sempre que alguma delas for aplicada, a ação adaptativa associada é acionada, causando as devidas alterações correspondentes no conjunto de regras do dispositivo não-adaptativo subjacente. Porém, quaisquer possíveis alterações no comportamento do dispositivo devem ser plenamente conhecidas a priori, em quaisquer etapas de sua operação em que tais alterações de comportamento devam se efetivar.

A principal característica dessa formulação é que ela apresenta a desejável propriedade de preservar a natureza do formalismo não-adaptativo, de tal forma que o dispositivo adaptativo resultante possa ser facilmente compreendido por todos os que tiverem familiaridade com o formalismo não-adaptativo subjacente original.

Em [15] é possível encontrar diversos tipos de dispositivos adaptativos já propostos, sendo alguns citados a seguir: Autômatos Adaptativos, Statecharts Adaptativos, Redes de Markov Adaptativas, Tabelas de Decisão Adaptativas, Árvores de Decisão Adaptativas, Redes de Petri Adaptativas, ISDL Adaptativos, entre outros.

B. Aplicações de Dispositivos Adaptativos

Inúmeras aplicações potenciais existem para a tecnologia derivada da adaptatividade, incluindo: inferência [21, 18], arte usando computador [20, 2], processamento de linguagem natural [19, 39], síntese de voz [32], reconhecimento de padrões [9, 32], tomada de decisão [32, 26, 40], linguagens de programação adaptativas [34; 11, 28], otimização de código [16], meta-modelagem [5], computação evolutiva [31, 3], engenharia de software [35], robótica [9, 37] segurança (security) [27, 7, 8] e, ensino por computador [10], entre outras.

C. Classificação de Dispositivos Adaptativos

Os formalismos subjacentes dos dispositivos adaptativos podem ser classificados em variadas categorias, conforme sua forma de operação. Entre outras, têm-se as seguintes, definidas em [24]:

- ✓ dispositivos de reconhecimento, da classe dos autômatos, baseados na sucessão de mudanças de estados;
- ✓ dispositivos de processamento, como as linguagens de programação adaptativas, que permitem descrever a lógica de programas com código automodificável;
- ✓ dispositivos de geração, da classe das gramáticas, baseados na aplicação sucessiva de regras de substituição;
- ✓ dispositivos para a representação de sistemas assíncronos, tais como os statecharts, que incorporam mecanismos responsáveis pela representação de fenômenos de sincronização;
- ✓ dispositivos estocásticos, como as redes de Markov, capazes de representar fenômenos de caráter aleatório;
- ✓ dispositivos de auxílio à tomada de decisões, representados principalmente pelas tabelas de decisão e pelas árvores de decisão;

D. Máquina de Moore (não-adaptativa)

Uma Máquina de Moore [12, 13, 14] é um autômato finito determinístico modificado, que possui saídas associadas aos estados. Tal máquina possui uma função que gera uma palavra de saída para cada estado da máquina, podendo essa ser uma palavra vazia.

É representada por uma héptupla:
 $ND = (C, NR, S, c_0, A, NA, RS)$, onde:

- ✓ ND é uma Máquina de Moore (não-adaptativa), cuja operação é determinada pelo conjunto de regras NR .
- ✓ C é o conjunto de todas as suas possíveis configurações, e $c_0 \in C$ é sua configuração inicial.
- ✓ S é o conjunto (finito) de todos os possíveis eventos que são estímulos de entrada válidos para ND , com $\varepsilon \in S$.
- ✓ $A \subseteq C$, (respectivamente, $F = C - A$) é o subconjunto de todas as suas configurações de aceitação (respectivamente, configurações de rejeição).
- ✓ ε denota “vazio”, e representa o elemento nulo de conjunto ao qual ele pertence.
- ✓ $w = w_1 w_2 \dots w_n$ é uma cadeia de estímulos de entrada, onde $w_k \in S - \{\varepsilon\}$, $k = 1, \dots, n$, com $n \geq 0$.
- ✓ NA é um conjunto (finito), com $\varepsilon \in NA$, de todos os possíveis símbolos a serem emitidos como saídas por ND .
 - ✓ RS é a função de saída por meio da relação $RS \subseteq C \times NA^*$ a qual é uma função total que determina a geração de uma palavra de saída para cada estado.
- ✓ NR é o conjunto de regras que definem ND por meio da relação $NR \subseteq C \times S \times C$. As regras $r \in NR$ apresentam-se na forma $r = (c_i, s, c_j)$, indicando que, em resposta a um estímulo de entrada qualquer $s \in S$, r

modifica a configuração corrente de c_i para a nova configuração c_j , e consome s .

Uma regra $r = (c_i, s, c_j)$, com $r \in NR$; $c_i, c_j \in C$; $s \in S$, é dita compatível com a configuração corrente c se e apenas se $c_i = c$, desde que s seja vazio ou igual ao estímulo de entrada corrente do dispositivo. Neste caso, a aplicação de uma regra compatível move o dispositivo para a configuração c_j (denotada por $c_i \Rightarrow^s c_j$). Note-se que s pode ser vazio.

Um seqüência opcional de movimentos em vazio, finalizada por um movimento que consuma o símbolo w_k , é denotada por: $c_i \square \sim c_m$, $m \geq c_i \Rightarrow^{\sim} c_m$, $m \geq 0$, e abrevia $c_i \Rightarrow^{\varepsilon} c_1 \Rightarrow^{\varepsilon} c_2 \Rightarrow^{\varepsilon} \dots \Rightarrow^{\varepsilon} c_m$.

Um seqüência opcional de movimentos em vazio, finalizada por um movimento que consuma o símbolo w_k , é denotada por: $c_i \Rightarrow^{\sim} c_j$ e representa $c_i \Rightarrow^{\sim} c_m \Rightarrow^{w_k} c_j$.

Diz-se que uma cadeia de entrada $w = w_1 w_2 \dots w_n$ é aceita por ND quando $c_0 \Rightarrow^{\sim w_1} c_1 \Rightarrow^{\sim w_2} \dots \Rightarrow^{\sim w_n} c_n \Rightarrow^{\sim} c$ (abreviadamente denotada como $c_0 \Rightarrow^w c$, com $c \in A$). De forma complementar, diz que w é rejeitada por ND quando $c \in F$.

A linguagem definida por ND é o conjunto: $L(ND) = \{w \in S^* \mid c_0 \Rightarrow^w c, c \in A\}$ de todas as cadeias $w \in S^*$ que são aceitas por ND .

O processamento de uma Máquina de Moore para uma dada entrada w consiste na sucessiva aplicação da função programa para cada símbolo de w (da esquerda para a direita), até ocorrer uma condição de parada. A palavra vazia como saída da função programa indica que nenhuma gravação é realizada e, portanto, a cabeça da fita de saída não se move. Se todos os estados geram saída vazia, então a Máquina de Moore se comporta como se fosse um autômato finito.

Outro tipo de transdutor é conhecido como máquina de Mealy, que também é um autômato finito modificado, mas que possui as palavras de saída associadas com as transições entre os estados. Neste tipo de máquina, as palavras de saída dependem do estado atual e do valor das entradas. Essa máquina não será abordada aqui.

E. Máquina de Moore Adaptativa

Uma Máquina de Moore Adaptativa (MMA) é um caso particular de aplicação do conceito de dispositivo guiado por regras adaptativo [22, 32], no qual o mecanismo subjacente utilizado é o da Máquina de Moore (descrito na seção anterior).

Complementarmente a um autômato adaptativo, uma Máquina de Moore Adaptativa pode adicionar ou remover as saídas associadas a cada estado, além de ter funções adaptativas para adicionar ou remover estados e transições.

Defina-se um contador T , que é iniciado com o valor *zero*, e automaticamente incrementado de uma unidade sempre que uma ação adaptativa não-vazia for executada. Cada valor k assumido por T pode ser usado para indexar os nomes de conjuntos variantes no tempo. Neste caso, tal indexação seleciona o passo k de operação de AD .

Um dispositivo $AD = (ND_0, AM)$ é dito adaptativo sempre que, para qualquer passo de operação $k \geq 0$, AD seguir o comportamento de ND_k , até que a execução de alguma ação adaptativa não-vazia inicie seu $(k+1)$ -ésimo passo ao modificar seu próprio conjunto de regras.

Em qualquer passo $k \geq 0$ de operação de AD , sendo ND_k o correspondente dispositivo subjacente definido por R_k , a execução de qualquer ação adaptativa não-vazia provoca a evolução de R_k para R_{k+1} . Assim, AD começa a executar seu $(k+1)$ -ésimo passo de operação já com o conjunto R_{k+1} , resultante das alterações impostas, no passo anterior, ao conjunto R_k . Daí em diante, AD seguirá o comportamento de ND_{k+1} até que alguma outra ação adaptativa não-vazia provoque o início de um novo passo de operação. Esse procedimento se repete até que a cadeia de entrada tenha sido integralmente processada ou que algum erro interrompa a operação de AD .

O dispositivo adaptativo AD inicia sua operação em c_0 , em sua forma inicial $AD_0 = (C_0, R_0, S, c_0, A, NA, BA, AA)$. No passo $k \geq 0$, um estímulo de entrada move AD para sua próxima configuração e então AD inicia seu $(k+1)$ -ésimo passo de operação se e somente se uma nova ação adaptativa não-vazia for executada. Dessa forma, estando AD em seu passo k , e apresentando-se na forma $AD_k = (C_k, R_k, S, c_k, A, NA, BA, AA)$, a execução de uma ação adaptativa não-vazia deve conduzi-lo a $AD_{k+1} = (C_{k+1}, R_{k+1}, S, c_{k+1}, A, NA, BA, AA)$.

Nessa formulação, tem-se:

- ✓ $AD = (ND_0, AM)$ representa algum dispositivo adaptativo, dado por um dispositivo subjacente iniciado por ND_0 e um mecanismo adaptativo AM .
- ✓ ND_k é o dispositivo não-adaptativo subjacente de AD em algum passo de operação k . ND_0 é o dispositivo não-adaptativo subjacente de AD , definido pelo conjunto inicial de regras não-adaptativas R_0 . Por definição, quaisquer regras não-adaptativas, em qualquer R_k , espelham as regras adaptativas correspondentes em R_k .
- ✓ C_k é o conjunto de todas as possíveis configurações que ND pode assumir no seu $(k+1)$ -ésimo passo de operação, e $c_k \in C_k$ é a configuração em que inicia tal passo de operação. Para $k = 0$, tem-se, respectivamente, C_0 , o conjunto inicial de configurações válidas, e $c_0 \in C_0$, a configuração inicial, tanto para ND_0 como para ND .
- ✓ ε denota "vazio", no contexto em que for empregado, a ausência de qualquer outro elemento válido do conjunto correspondente.
- ✓ S é o conjunto (finito, invariável) de todos os possíveis

eventos que são possíveis estímulos de entrada para AD ($\varepsilon \in S$).

- ✓ $A \subseteq C$, (respectivamente, $F = C - A$) é o subconjunto de todas as suas configurações de aceitação (respectivamente, de rejeição).
- ✓ BA e AA são os conjuntos de ações adaptativas, ambos contendo a ação adaptativa vazia ($\varepsilon \in BA \cap AA$).
- ✓ $w = w_1 w_2 \dots w_n$, $k = 1, \dots, n$ é uma cadeia de estímulos não-vazios de entrada, ou seja, $w_k \in S - \{\varepsilon\}$.
- ✓ NA , com $\varepsilon \in NA$, é um conjunto (finito, invariável) de todos os possíveis símbolos passíveis de serem gerados por AD como efeitos colaterais da aplicação de regras adaptativas.
- ✓ AR_k é um conjunto de regras adaptativas, dadas por uma relação $AR_k \subseteq BA \times C \times S \times C \times NA \times AA$. Em particular, AR_0 define o comportamento inicial de AD . Ações adaptativas mapeiam o conjunto corrente de regras adaptativas AR_k de AD em um novo conjunto AR_{k+1} , através da aplicação de um conjunto de operações de inclusão e/ou remoção de regras adaptativas AR_k . Regras adaptativas $ar \in AR_k$ apresentam-se na forma $ar = (ba, c_i, s, c_j, z)$, significando que, em resposta a algum estímulo de entrada $ar \in AR_k$, ar executa inicialmente a ação adaptativa $ba \in BA$; a execução de ba é descontinuada caso venha a eliminar ar de AR_k ; caso contrário, aplica-se a regra não adaptativa subjacente $nr = (c_i, s, c_j, z)$, conforme descrito anteriormente para o dispositivo não-adaptativo; finalmente, executa-se a ação adaptativa $aa \in AA$.
- ✓ Defina-se AR como o conjunto de todas as possíveis regras adaptativas para o dispositivo AD .
- ✓ Defina-se R como o conjunto de todas as possíveis regras não-adaptativas para ND , o dispositivo subjacente de AD .
- ✓ $AM \subseteq BA \times R \times AA$, definido para um determinado dispositivo adaptativo AD , é o mecanismo adaptativo de AD que deve ser aplicado, em qualquer passo k de operação, a cada regra $R_k \subseteq R$, e deve ser tal que opere como função, quando aplicado a qualquer subdomínio $R_k \subseteq NR$. Isso determina um único para de ações adaptativas a serem associadas a cada regra não-adaptativa.

Note-se que a definição de AR_k como subconjunto do produto cartesiano $BA \times C \times \dots \times AA$ pode ser refinada, à luz da definição do mecanismo adaptativo AM . Assim, pode-se reinterpretar AR_k como subconjunto do produto cartesiano $BA \times AR_k \times AA$.

O conjunto $AR_k \subseteq AR$ pode ser construído como a coleção de todas as regras adaptativas que possam ser obtidas pela associação de pares de ações adaptativas com as regras não-adaptativas em AR_k . Equivalentemente, como as regras de NR_k espelham as de AR_k em cada momento, é possível construir AN_k simplesmente removendo-se, das regras de AR_k , todas as referências a ações adaptativas.

IV. EXEMPLO DE APLICAÇÃO

Primeiramente, é apresentada a estrutura geral do sistema, na qual depois de sua abertura, é apresentado um menu para que o aluno possa escolher o curso que deseja participar. De maneira genérica, o sistema prevê o oferecimento de diferentes cursos. Assim, após a escolha do curso desejado, é realizada uma chamada à sub-máquina correspondente, passando de "A" e retornando em "B" após a realização do curso. Ao término do curso, retorna-se ao menu de escolha de cursos. As figuras 2 e 3, representam os autômatos desse modelo.

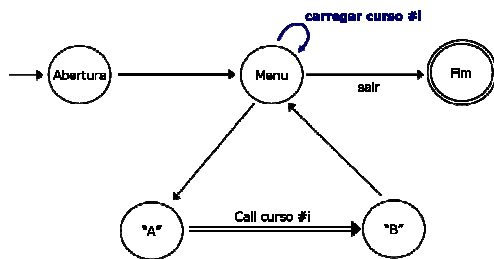


Figure 2: Autômato das chamadas aos cursos disponíveis.

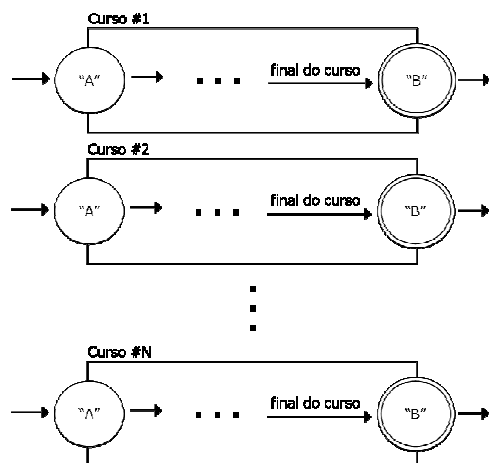


Figure 3: Representação das chamadas dos cursos.

A seguir, será, então, apresentado um exemplo demonstrando como será a representação de cada um dos cursos disponibilizados aos alunos. O exemplo será apresentado através de uma Máquina de Moore (não adaptiva). Posteriormente, serão apontadas as mudanças propostas para tornar o curso mais dinâmico, através do uso da adaptatividade.

A idéia central é elaborar um curso desmembrando-o seu material didático em pequenos blocos chamados de objetos de aprendizagem [41]. Assim, o curso pode ser montado com o mesmo material, de diferentes formas, ou ainda, ser modificado dinamicamente.

Cada curso é definido como um autômato [17] e referencia um conjunto de objetos de aprendizagem independentes, tal que o roteiro de estudos do curso permanece separado do material didático. Cada estado representa um tópico de estudo

dentro do curso. As transições compõem o roteiro das aulas. A função de saída funciona como a ligação lógica dos estados com os objetos de aprendizagem.

A. Curso com Máquina de Moore

Em [17] é apresentado uma forma de modelagem de cursos para Web utilizando autômatos finitos determinísticos com saída, cuja principal finalidade é a criação de material hipermídia independente do esquema de navegação. Na proposta, um curso pode ser representado tanto por Máquina de Moore ou quanto por Máquina de Mearly, tendo o material didático vinculado, respectivamente, aos estados ou às transições do transdutor. Em ambos os casos, esse autômato com saída fornece uma máquina abstrata para o controle da navegação em hipertextos.

Segundo esse modelo, cada autômato constitui um curso definido sobre um conjunto de hiperdocumentos (unidades de informação) independentes. As n-uplas dos autômatos de cursos apresentam uma correspondência às estruturas de hiperdocumentos na Web. O alfabeto de símbolos de entrada é um conjunto de nomes que identificam as âncoras de navegação no hipertexto. As saídas estão relacionadas a unidades de informação constituídas por hiperdocumentos e as palavras de saída, presentes na função de saída, são hipertextos apresentados como uma única página Web no navegador do usuário. As transições, definidas na função programa, funcionam como ligações lógicas (e não ligações físicas no documento) entre os conteúdos dos cursos e definem possíveis links a serem selecionados durante a navegação pelo hipertexto.

Uma das vantagens da solução utilizada é que a estrutura dos autômatos com saída permite a criação do material hipermídia de forma independente do autômato em si, possibilitando a programação de seqüências de estudo com objetivos e enfoques específicos, reuso de parte ou íntegra das páginas Web em diversos cursos, eliminando a redundância na criação de páginas, bem como a modularização que facilita a expansibilidade do sistema.

A estrutura do sistema de hipertexto que não utiliza links diretos permite a estruturação e edição do conjunto de materiais hipermídia sem a necessidade do autor dos documentos se preocupar com a inserção de links diretamente nos textos, facilitando a tarefa de edição dos documentos. Um benefício adicional é que, como páginas de um curso podem ser constituídas por outras sub-páginas concatenadas, o autor pode construir funções de saída/programa diferentes que contenham acesso a páginas de mesmo conteúdo, evitando-se redundância na criação dos documentos Html. Por exemplo, pode existir uma função que gere uma página contendo uma "definição" e um "exemplo", enquanto outra página pode ter o mesmo "exemplo" como "dica" para um exercício.

Tomando como exemplo um mini-curso para ensino de

algoritmos, usando Máquina de Moore, tem-se:

Alfabeto de entrada: { próxima, anterior, exercício, resumos, saída }
Alfabeto de saída: { A, B, C, D, E, F, G, H, I }
A – Introdução a Algoritmos
B – Definição de Desvio Condicional
C – Exemplo de Desvio Condicional
D – Exercícios sobre Desvio Condicional
E – Definição de Laço de Repetição
F – Exemplo de Laço de Repetição
G – Exercícios sobre Laço de Repetição
H – Conclusões
I – Fim

A figura a seguir ilustra uma possível configuração de Máquina de Moore, com suas regras de transição e regras de saída para a representação de um mini-curso sobre algoritmos.

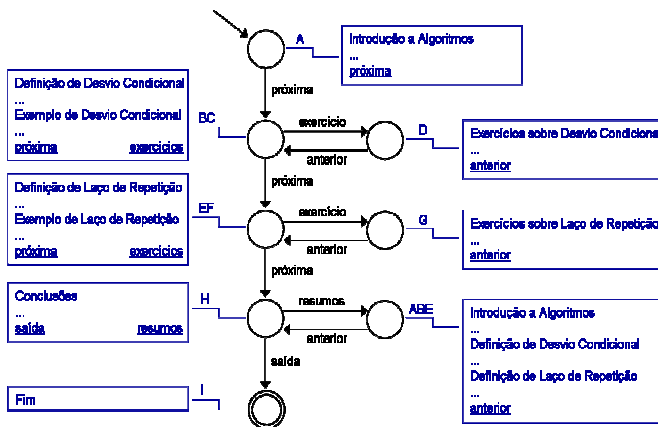


Figure 4: Curso de Algoritmos representado com Máquina de Moore.

É possível notar que num mesmo estado podem existir mais de uma unidade de informação associada, uma vez que a função de saída pode gerar palavras a partir do fecho de Kleen do alfabeto de símbolos de saída. É possível, também, notar o reuso dessas unidades de informação, que podem aparecer por mais de uma vez durante o curso. Da mesma maneira, é também verdade que esse material didático possa ser reaproveitado em outros cursos.

B. Curso com Máquina de Moore Adaptativa

Para a modelagem de um curso estático, o funcionamento de uma Máquina de Moore é suficiente para se conseguir especificar formalmente todo o modelo. Contudo, um enriquecimento nesse tipo de máquina, através da camada adaptativa, permite maior poder de representação, podendo tornar o acompanhamento do curso mais flexível e dinâmico, podendo tratar questões dependentes de contexto. Desta forma, um sistema adaptativo tenta antecipar as necessidades e desejos dos alunos a partir de modelos que representam o seu perfil, nível de conhecimento e preferências.

Com base nos modelos de Sistemas Hiperídia

Adaptativos [4, 25], é proposta a aplicação de ações adaptativas em três níveis:

- Adaptatividade na navegação
- Adaptatividade na apresentação
- Adaptatividade no conteúdo

A adaptatividade na navegação tem por objetivo demonstrar um caminho ideal para que um aluno em particular alcance seus objetivos, evitando que o mesmo se disperse, diante de um amplo conjunto de opções e sintaxe desorientado diante da grande quantidade de material oferecido. A adaptação na navegação é obtida acrescentando-se novas transições no modelo do curso ou, ainda, removendo-se algumas das transições existentes. Uma vez que as transições definem o roteiro de aulas do curso, pode-se manter o mesmo conteúdo programático, porém oferecer seqüências alternativas para seguir do curso. De acordo com o perfil de cada aluno, pode-se ofertar um número maior de caminhos a serem seguidos ou restringir essas possibilidades, tornando o curso mais linear e assim limitar o espaço de busca. As regras adaptativas nesse nível incidem exclusivamente sobre a criação e remoção de transições.

A adaptatividade na apresentação diz respeito à forma como as informações serão exibidas para os alunos, dependendo do perfil de cada aluno. Ou seja, um mesmo conteúdo pode ser apresentado utilizando-se diferentes layouts ou mesmo através de diferentes mídias, conforme as preferências do usuário. Nota-se, também, que o material a ser apresentado ao aluno não está necessariamente pronto e pode ser gerado em tempo real. Essencialmente, a adaptatividade na apresentação implica em se modificar as palavras de saída associadas aos estados.

A adaptatividade no conteúdo refere-se a possibilidade de se substituir um objeto de aprendizagem por outro mais elaborado, ou mesmo mais simplificado, dependendo da situação. Com isso, o curso pode ser atualizado trocando-se as palavras de saída ou até mesmo criando-se novos estados, com tópicos complementares não contidos no curso original.

As regras para o acompanhamento do curso podem ser definidas pelo professor, sendo aplicadas de acordo com a disciplina a ser abordada ou a linha pedagógica do professor.

A seguir é apresentado um simples exemplo no qual o professor define que o aluno precisa obter conhecimentos teóricos para somente depois poder fazer os exercícios sobre o assunto abordado. O aluno poderá retornar ao material teórico, caso necessite, e retomar o desenvolvimento dos exercícios. Contudo, somente após realizar os exercícios, o aluno poderá fazer a avaliação sobre o conteúdo abordado. No entanto, a avaliação poderá ser realizada uma única vez. Para tratar essa dependência de contexto, é criada uma ação adaptativa disparada após o aluno terminar sua avaliação. Essa ação adaptativa irá remover a transição entre os exercícios e a avaliação, impedindo que o aluno volte a fazer a avaliação.



Figure 5: Configuração inicial, antes do aluno realizar a avaliação.



Figure 6: Configuração do curso após a avaliação.

Uma questão importante a ser pensada diz respeito a onde e como aplicar regras adaptativas. A princípio, essas regras devem ser criadas pelo professor responsável pela elaboração do curso. Contudo, a critério do próprio professor, essas regras podem ser baseadas na própria inteligência coletiva do grupo de alunos participantes. Ou seja, a participação ativa dos alunos através de sugestões de novos roteiros de aula ou novos objetos de aprendizagem pode contribuir e definir as novas configurações do curso para os novos aprendizes.

V. CONCLUSÕES

O uso de um modelo baseado em Máquina de Moore Adaptativa para projetar sistemas de ensino assistidos por computador apresenta um grande potencial para se criar ambientes de ensino flexíveis, que se ajustem ao perfil de cada estudante.

Uma das características importantes, que ficou evidenciada na proposta apresentada, é a possibilidade de se criar o material didático de forma independente do autômato. A independência do material didático em relação ao roteiro das aulas permite, por exemplo: reutilização do material instrucional em diversos cursos. Outra característica importante é a possibilidade de se criar, para um mesmo curso, diferentes roteiros de aulas, com enfoques diferenciados, pois um curso pode ter seu autômato modificado dinamicamente.

O projeto encontra-se em fase de desenvolvimento. Já existe disponível um protótipo capaz de executar o modelo proposto de Máquina de Moore Adaptativa, representando cursos sobre quaisquer assuntos, obedecendo ao roteiro de aulas previstas e controlando, por exemplo, questões como pré-requisitos entre módulos. O protótipo está sendo desenvolvido para uso em ambiente Web. Dessa forma, basta apenas um programa navegador da Internet para poder utilizar o ambiente, independente de plataforma ou sistema operacional utilizado.

Como trabalhos futuros, pode-se: aplicar o conceito de máquina de Moore adaptativa para a elaboração de provas adaptativas; monitorar o comportamento de diversos alunos e, através dos conceitos de Inteligência Coletiva, permitir que os

materiais didáticos do curso e/ou roteiros de estudos do curso sejam modificados; expandir o conceito usado para curso, para que o mesmo possa ser aplicado, por exemplo, numa grade curricular completa, e, assim, trabalhar com adaptatividade multi-nível; criar uma ferramenta de autoria, que ofereça uma interface gráfica amigável para que o professor possa montar o roteiro do curso, sem a necessidade de ser treinado.

REFERÊNCIAS

- [1] ALMEIDA Jr, J. R.; *STAD – Uma Ferramenta para Representação e Simulação de Sistemas Através de Statecharts Adaptativos*; Tese de Doutorado; Escola Politécnica; Universidade de São Paulo; São Paulo; 202p;1995.
- [2] BASSETO, B. A.; *Um sistema de composição musical automatizada, baseado em gramáticas sensíveis ao contexto, implementado com formalismos adaptativos*; Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2000.
- [3] BRAVO, C. et al. *Towards an Adaptive Implementation of Genetic Algorithms* INBI 2007, XXXIII CLEI – Conferencia Latinoamericana de Informática, San José, Costa Rica, 2007.
- [4] BRUSILOVSKY, P. *Methods and Techniques of Adaptive Hypermedia. User Modeling and User Adapted Interaction. Special issue on adaptive hypertext and hypermedia*. Pittsburgh, 1996.
- [5] CAMOLESI, A. R.; *Proposta de um Gerador de Ambientes para a Modelagem de Aplicações usando Tecnologia Adaptativa*; Tese de Doutorado; Escola Politécnica; Universidade de São Paulo; São Paulo; 2007.
- [6] CAMOLESI, Almir Rogério; JOSÉ NETO, João. *Modelagem AMBER-Adp de um ambiente para Gerenciamento de Ensino a Distância*. Anais do XIII Simpósio Brasileiro de Informática na Educação. São Leopoldo-RS, 2002. v. 1, p. 401-409.
- [7] CEREDA, P.R.M. & ZORZO, S.D.; *Access Control Model Formalism using Adaptive Automaton*. IEEE Latin America Transactions. Volume 6, Issue 5, ISSN: 1548-0992, September 2008.
- [8] CEREDA, P.R.M.; *Modelo de Controle de Acesso Adaptativo*. Dissertação de Mestrado, UFSCAR, São Carlos/SP, 2008.
- [9] COSTA, E. R., HIRAKAWA, A. R., NETO, J. J.; *An Adaptive Alternative for Syntactic Pattern Recognition*. Proceeding of 3rd International Symposium on Robotics and Automation, ISRA 2002, pp. 409-413. Toluca, Mexico, 2002.
- [10] DIZERÓ, W. J.; *Uso de Máquina de Moore Adaptativa na Modelagem de Cursos*; 3º Workshop de Tecnologia Adaptativa – WTA'2009; São Paulo-SP, Brasil, 2009.
- [11] FREITAS, A.V. & NETO, J. J.; *Adaptive Languages and a New Programming Style* 6th WSEAS International Conference on Applied Computer Science (ACS'06) – Tenerife, Canary Islands, Spain, December, 2006.
- [12] HARRISON, Michael A.; *Introduction to Formal Language Theory*, Ed. Addison-Wesley; 1a edição; Califórnia – USA (1978)
- [13] HOPCROFT, J. E., Ullman, J. D.; *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley (1979).
- [14] LEWIS, Harry R. & Papadimitriou, Christos H.; *Elementos de Teoria da Computação*. Ed. Bookman, 2ª edição, (2000)
- [15] LTA - Laboratório de Linguagens e Tecnologias Adaptativas; Poli-USP; São Paulo; <http://www.pcs.usp.br/~lta/>; site visitado em Novembro/2009.
- [16] LUZ, J.C.; *Tecnologia Adaptativa Aplicada à Otimização de Código em Compiladores*; Dissertação de Mestrado, EPUSP, São Paulo, 2004.

- [17] MACHADO, Júlio P. et al.; *Autômatos Finitos: um Formalismo para Cursos na Web*; XIII Simpósio Brasileiro de Engenharia de Software 1999, SBES'99, Florianópolis, Brasil.
- [18] MATSUNO, I.P.; *Um Estudo do Processo de Inferência de Gramáticas Regulares e Livres de Contexto Baseados em Modelos Adaptativos*. Dissertação de Mestrado, USP, São Paulo, 2006.
- [19] MENEZES, C.E.D.; *Um Método para a Construção de Analisadores Morfológicos, Aplicado à Língua Portuguesa, Baseado em Autômatos Adaptativos*. Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2000.
- [20] NETO, J. J. & BASSETO, B. A.; *A Stochastic Musical Composer Based on Adaptive Algorithms*; Proceedings of the 6th Brazilian Symposium on Computer Music - SBC&M99, Rio de Janeiro, 1999.
- [21] NETO, J. J. & IWAI, M.K.; *Adaptive Automata for Syntax Learning*. CLEI 98 – XXIV Conferencia Latinoamericana de Informática, MEMORIAS. pp. 135-149, Quito, Equador, 1998.
- [22] NETO, J. J., *Adaptive Rule-Driven Devices - General Formulation and Case Study*. Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Vol. 2494, Pretoria, South Africa, July 23-25, Springer-Verlag, 2001, pp. 234-250.
- [23] NETO, J. J., *Contribuições à Metodologia de Construção de Compiladores*, Thesis (Livre Docência) – Escola Politécnica da Universidade de São Paulo, São Paulo, 1993.
- [24] NETO, J. J.; *Um Levantamento da Evolução da Adaptatividade e da Tecnologia Adaptativa*; **IEEE Latin America Transactions**; vol. 5 n° 7; novembro 2007.
- [25] PALAZZO, L. A. M.; *Sistemas de Hipermídia Adaptativa*, In Anais do XXII Congresso da Sociedade Brasileira de Computação (SBC), Florianópolis, 2002
- [26] PEDRAZZI, T., TCHEMRA, A. H., ROCHA, R. L. A.; *Adaptive Decision Tables - a Case Study of their Application to Decision-Taking Problems*; Proceedings of International Conference on Adaptive and Natural Computing Algorithms - ICANNGA 2005, Coimbra, March 21-23, 2005.
- [27] PELEGRINI, E. J. & NETO, J. J.; *Applying Adaptive Technology in Data Security*. Proceedings of the 6th Peruvian Computer Week - JPC 2007, Trujillo, Peru, Novembro 5-10 (pp. 31-40), 2007.
- [28] PELEGRINI, E. J.; *Códigos Adaptativos e Linguagem para Programação Adaptativa: Conceitos e Tecnologia*. Dissertação de Mestrado, EPUSP, São Paulo, 2009.
- [29] PIMENTA, Pedro e BAPTISTA, Ana Alice. *Das plataformas de E-learning aos objetos de aprendizagem*. In. DIAS, Ana Augusta Silva e GOMES, Maria João. Elearning para e-formadores. Minho, TecMinho, 2004, p. 97-109.
- [30] PISTORI, H. et al. *Defect Detection in Raw Hide and Wet Blue Leather CompIMAGE – Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, Coimbra, 2006.
- [31] PISTORI, H., MARTINS, P.S., CASTRO Jr., A.A.; *Adaptive Finite State Automata and Genetic Algorithms – Merging Individual Adaptation and Population Evolution*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms – ICANNGA 2005, Coimbra, March, 2005.
- [32] PISTORI, H.. *Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações*. Tese (Doutorado) - Universidade de São Paulo, 2003.
- [33] ROCHA, R. L. A.& NETO, J. J.; *Autômato adaptativo, limites e complexidade em comparação com máquina de turing*; In: Proceedings of the second Congress of Logic Applied to Technology - LAPTEC'2000. São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia: [s.n.], 2000.
- [34] ROCHA, R. L. A.& NETO, J. J.; *Uma proposta de linguagem de programação funcional com características adaptativas*. IX Congreso Argentino de Ciencias de la Computación. La Plata, Argentina, Octubre de 2003.
- [35] SILVA, P. S. M. & NETO, J. J. *An Adaptive Framework for the Design of Software Specification Languages*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms - ICANNGA 2005, Coimbra, March, 2005.
- [36] SILVA, P. S. M. & NETO, J. J. *An Adaptive Framework for the Design of Software Specification Languages*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms - ICANNGA 2005, Coimbra, March, 2005.
- [37] SLONNEGER, K. & KURTZ, B.L. *Formal Syntax and Semantics of Programming Languages – a Laboratory Based Approach* Addison Wesley, 1995.
- [38] SOUZA, M. A. A. & HIRAKAWA, A. H.; *Robotic Mapping and Navigation in Unknown Environments Using Adaptive Automata*. Proceedings of International Conference on Adaptive and Natural Computing Algorithms – ICANNGA 2005, Coimbra, March, 2005.
- [39] STANDISH T. A.; *Extensibility in Programming Language Design* ACM SIGPLAN Notices, p. 18-21, vol 10 n. 7, July, 1975.
- [40] TANIWAKI, C. Y. O.; *Formalismos Adaptativos na Análise Sintática de Linguagem Natural* Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2001.
- [41] TCHEMRA, A. H.; *Tabela de Decisão Adaptativa na Tomada de Decisão Multicritério*. Tese de Doutorado, Escola Politécnica da USP, São Paulo, 2009.
- [42] WILEY, D. A. *Connecting learning objects to instructional theory: A definition, a metaphor and a taxonomy*. The Instructional Use of Learning Objects. Wiley, D. (Ed.) 2001. Disponível na URL: <http://www.reusability.org/read/chapters/wiley.doc>. 2001. Acesso em 20/03/2005.

Wagner José Dizeró nasceu em Piracicaba-SP, Brasil, em 29 de Agosto de 1975. Se graduou na Faculdade de Tecnologia de Americana (1996) e defendeu mestrado em Ciência da Computação pela Universidade Federal de São Carlos (1999). Atualmente, é aluno de doutorado do departamento de Engenharia da Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo. Profissionalmente, é Professor no Centro Universitário de Lins a 9 anos, onde também é coordenador dos cursos de Bacharelado em Sistemas de Informação e Pós-Graduação em Gestão de Sistemas de Informação. Entre seus campos de interesse encontram-se: tecnologias adaptativas, engenharia de software, sistemas para Internet, informática na educação e educação a distância.

João José Neto é graduado em Engenharia de Eletricidade (1971), mestre em Engenharia Elétrica (1975), doutor em Engenharia Elétrica (1980) e livre-docente (1993) pela Escola Politécnica da Universidade de São Paulo. Atualmente, é professor associado da Escola Politécnica da Universidade de São Paulo e coordena o LTA – Laboratório de Linguagens e Tecnologia Adaptativa do PCS – Departamento de Engenharia de Computação e Sistemas Digitais da EPUSP. Tem experiência na área de Ciência da Computação, com ênfase nos Fundamentos da Engenharia da Computação, atuando principalmente nos seguintes temas: dispositivos adaptativos, tecnologia adaptativa, autômatos adaptativos, e em suas aplicações à Engenharia de Computação, particularmente em sistemas de tomada de decisão adaptativa, análise e processamento de linguagens naturais, construção de compiladores, robótica, ensino assistido por computador, modelagem de sistemas inteligentes, processos de aprendizagem automática e inferências baseadas em tecnologia adaptativa.