

Hand skill programming: Using fuzzy sets to program topobo kinectis memory devices

Mateus Madail Santin, Silvia da Silva Costa Botelho
Programa de pós-graduação em Ciências
Universidade Federal do Rio Grande, FURG
Av. Itália km 8, Carreiros, 96201-900
Rio Grande, RS - Brazil
Email: mateus.santin@furg.br, silviacb@furg.br

Graçaliz Pereira Dimuro, Celso Luiz Lopes Rodrigues
Centro de Ciências Computacionais
Universidade Federal do Rio Grande - FURG
Av. Itália km 8, Carreiros, 96201-900
Rio Grande, RS - Brasil
Email: gracaliz@gmail.com, dmtcllr@furg.br

Abstract—Unlike other sciences such as Mathematics and Biology, for example, computation is presented to students at a later learning stage. Even though they are able to establish sets of procedures, it is alleged that in their first years of elementary school, children do not yet present cognitive structures capable of representing symbolically, through existing computer programming languages, the algorithms associated with such procedures. Within this context, there has appeared recently a robotic kit named Topobo, capable of capturing manual movements carried out in their blocks. This article presents the first result associated with the use of Topobo as a language of manual programming. The study will lead to a decompiler able to furnish, with its control flux and its structure of adequate data, the program which results from the manual manipulation of the system by the child. In a more precise form, this work anticipates the utilization of fuzzy concepts for the representation of knowledge generated from the registers of manual programming of the Topobo elements. The utilization of the fuzzy formalism will allow a qualitative and diffuse description of the knowledge, in a manner very similar to the “intuitive” and “little precise” way which human beings handle information (mainly children), leading to an adequate structure and representation of the data which are being manipulated. In addition to the formalism adopted, the article presents a 3D interface which will be helpful in the performance of the experiments. In possession of the decompiled programs, we intend to evaluate the structure of data and control fluxes which will emerge in order to identify the mental structures utilized in the construction of algorithms through manual programming by children.

Keywords—Topobo; logic fuzzy; skill programming; learning environment;

I. INTRODUÇÃO

O ensino de tópicos relacionados com as ciências, especialmente física e matemática, tem sido apoiado pela exploração didática de “kits” de componentes para montagem de robôs educacionais. Há vários fabricantes destes mas o exemplo mais conhecido é o dos “kits” Lego®. Em classes de ensino básico, os estudantes montam estruturas diversas, como, por exemplo, carrinhos com sensores, que seguem trilhas no terreno (mesa, assoalho etc.), ou símiles de animais como (cães, aranhas e outros). Tais procedimentos

propiciam ambiente para uma experiência de construção intuitiva de elementos conceituais diversos das áreas envolvidas. Ao projetar e montar os símiles de animais ou de máquinas conhecidas, o aprendiz irá construindo o conhecimento acerca do fenômeno simulado, passando a compreender como os animais andam ou dançam, por exemplo, ou como um veículo pode operar em um terreno ou outro. Há uma renovação do grau de interesse dos alunos pelos temas associados, na medida em que verificam as possibilidades dos dispositivos. Os “kits” também permitem descobrir afinidades entre cultura e aprendizagem tecnológica. Isto ocorre pela constatação, pelos alunos, de que há um mundo fora da escola em que os conceitos abordados pelas matérias escolares são de utilidade, percebendo alguma correlação deles com o que fazem os cientistas e técnicos. Desta forma, além de ser um meio de experimentação para a constatação, verificação e análise dos fenômenos estudados, o manuseio destes “kits” pode vir a despertar o interesse por carreiras tecnológicas a serem seguidas no futuro. Além de permitir o exercício da criatividade através das etapas de projeto e montagem, o potencial de uso pedagógico dos “kits” abrange a possibilidade de programação computacional dos sistemas desenvolvidos. Os aprendizes podem construir programas que, adequadamente, os guiem em seus movimentos, de modo a atuarem conforme o modelo que se pretende representar. Assim, para obterem o funcionamento adequado dos objetos, os estudantes são obrigados à especificação (formalizada em algum grau) dos fenômenos observados. Isto tem sido um obstáculo para crianças menores, pois envolve a tarefa de estabelecer algum tipo de especificação formal (seja através de um texto, seja por interação com uma interface gráfica), o que está em geral fora de seu alcance. Independente disso, os “kits” geram, sim, interesse aos estudantes, que, no entanto, podem desanimar-se pela complexidade do processo de programação. Com isso recentemente tem surgido os Topobos [8], uma nova geração de “kits” de desenvolvimento robótico, os quais são dotados do recurso tecnológico denominado “memória cinética”. Tal recurso permite o registro e gravação dos movimentos físicos

de manipulação realizados sobre as peças e componentes quando atuados por um agente externo (como uma pessoa que o manipule), e possibilidade de reprodução a posteriori. Este recurso tem como potencial permitir a estruturação de objetos ao ser incorporado a um “kit” de montagem que “aprende” os movimentos necessários pela reação às atuações dos usuários, através da memória cinética. Assim, um animal, por exemplo, um “cão”, com torso, cabeça e membros, pode ser montado tendo os componentes com memória cinética convenientemente dispostos, de modo que o “animal” “aprende a andar” pelo fato de repetir os movimentos que lhe são impostos pelo seu construtor, que lhe “ensina a andar” (ou “dançar”), como a uma criança pequena, a quem se força manualmente os movimentos dos membros. A experiência com crianças, principalmente, no uso dos “kits” de robótica educacional, é de que elas são mais refratárias à atividade de programação. Isto se dá, naturalmente, pela idade, por não terem ainda a maturidade lógico-formal. No entanto, operam bem com os dispositivos com memória cinética, pelo próprio caráter mais lúdico dos mesmos, bem como, pela característica de programação manual de suas atividades, sem qualquer necessidade de uso de linguagem formal de manipulação para descrever tais “algoritmos”. Atualmente desenvolve-se um hardware capaz de registrar em arquivos-texto as operações derivadas das ações manuais de “treinamento” efetuadas pelos usuários dos modelos construídos. Estes registros capturam o conjunto de procedimentos manualmente realizados pelo aprendiz, que decorrem da materialização dos algoritmos por eles desenvolvidos. Com base nesses registros, esta pesquisa tem como objetivo buscar padrões reconhecíveis de estruturas e fluxos de controle usadas na especificação destes algoritmos. Presume-se que elas existam, visto que, em princípio, no mínimo, seqüências simples de atuações diversas são necessárias para implementação de comportamentos básicos, além de repetições para locomoção, por exemplo, ou alternativas baseadas em condições, quando é necessário reequilíbrio. A apropriada decompilação de tais registros poderá conduzir a extração dos elementos básicos, fluxo de controle e estrutura de dados abstratas utilizadas pelas crianças durante suas ações de “programação manual” das peças. Busca-se definir uma linguagem de programação capaz de representar as estruturas de controle existentes, bem como as informações que estão sendo manipuladas e decompiladas pelo sistema. Primeiramente pretende-se definir um modelo formal capaz de descrever a informação manipulada resultante da decompilação. Tal modelo deve ser capaz de, o mais fidedigno possível, representar a informação que está sendo tratada pelas crianças, uma vez que o resultado da decompilação será utilizado para possível entendimento do processo cognitivo estabelecido na construção de algoritmos. Considerando que o formalismo fuzzy permite a descrição qualitativa e difusa do conhecimento, muito semelhante a maneira “intuitiva” e “pouco precisa” que seres

humanos tratam as informações (principalmente as crianças) este artigo apresenta os primeiros resultados associados a utilização de conceitos fuzzy para representação do conhecimento advindo dos registros de programação manual dos elementos Topobo. No processo de decompilação, as informações envolvidas nos registros da memória cinética serão fuzzificadas e utilizadas para descrever as variáveis dos programas resultantes. Na próxima seção apresenta-se a ferramenta para a extração e registro da programação manual. É apresentado o kit Topobo, sua memória cinética e a interface de captura de registros. A seção 3 apresenta os principais conceitos da lógica Fuzzy, necessários para a modelagem da informação. Na seção 4 apresenta-se uma proposta de formalização fuzzy para as variáveis manipuladas, da decompilação a qual é validada através de testes apresentados na seção 5. Conclusões e perspectivas futuras finalizam o artigo.

A. Decompilando Programas Manuais

Nesta seção será apresentada uma ferramenta para registro de programação manual passível de ser realizada utilizando kits robóticos do tipo Topobo. Serão apresentados o kit robótico e sua memória cinética, bem como uma proposta de decompilador e sua interface de registro desenvolvida.

1) *O Kit TOPOBO*: O nome Topobo é derivado das palavras topology, botanical, e robot, o qual se constitui em um sistema de modelagem em 3D que permite ser configurado de diferentes formas e programado para fisicamente se mover. A capacidade de representação 3D do Topobo permite que o aluno contextualize e crie objetos relacionados com a sua percepção. Utilizado como ferramenta pedagógica, o Topobo permite que o professor proponha situações-problemas que desafiem os alunos a resolvê-las com respostas buscadas em sua bagagem cultural e principalmente, no ato de suas ações. O diálogo, a discussão, como manifestação do ser humano dá-se em duas dimensões: “ação e reflexão” [3]. As interações do aluno com a ferramenta, propiciadas pela programação e reprodução dos movimentos estabelece um interessante ambiente de aprendizado e experimentação. Segundo Hodson, qualquer método didático que faça com que o aprendiz seja ativo, mais do que passivo, está de acordo com a ideia de que os alunos aprendem melhor pela experiência direta [4]. O Topobo é composto de 10 dez blocos primitivos básicos os quais se conectam em diversas orientações. Nove destas primitivas são chamadas “passivas” pois formam ligações estáticas, sendo que quatro das formas geométricas ocorrem em duas escalas. A décima primitiva denominada “ativa”, é constituída de um motor associado a componentes eletrônicos constituindo uma tecnologia embarcada. Quando estas peças são unidas, criam uma rede peer-to-peer descentralizada. Na Figura 1. são mostradas as diversas partes do Topobo. No centro encontra-se a primitiva ativa (motor) o qual tem memória cinética e é o responsável por dar movimento a



Figure 1. Conjunto de primitivas ativas e passivas que formam o Topobo

estrutura. Ao seu redor os blocos passivos (estáticos), que combinados formam várias estruturas, possuem coloração diferenciada, a interconexão dos blocos estáticos é realizada através de conectores transparente que se fixam tanto nas peças passivas, quanto na peça ativa. O componente ativo constituído do motor principal e de circuitos elétricos que se encarregam da distribuição de energia, memória, processamento e da comunicação multicanal serial, é construído dentro de um invólucro plástico de alta resistência e possui um único botão responsável pela programação. Ambas as peças ativas e passivas são interligadas através da técnica mecânica de plugues, semelhante aquela utilizada pelo LEGO.

2) *A memória cinética: programando robôs “com as mãos”*: Além dos componentes passivos com função meramente estrutural, o Topobo apresenta o conceito de memória cinética associada aos componentes ativos, a qual, por meio de um botão liga/desliga, permite que a manipulação de tais juntas possam ser registradas. O usuário pode programar manualmente a movimentação de uma junta através do uso da movimentação de tal tipo de junta, que armazenará o movimento realizado. Outra tecnologia disponível para a manipulação dos elementos é um tipo especial de dispositivo denominado *Queens*. Este permite que o sistema replique às outras juntas um movimento realizado, transmitindo a estas o movimento executado na *Queen*. Todos os movimentos associados às peças ativas podem ser registrados. Para isso o Topobo disponibiliza uma interface de registro (botões de liga/desliga) capaz de gravar e reproduzir movimentos manualmente executados. Na Figura 2. mostra-se a forma de programação de uma peça ativa, a qual possui uma estrutura simples passiva conectada na saída do motor.

3) *Um Ambiente Misto de Aprendizado utilizando o Topobo.*: O ambiente misto de aprendizado é composto por um hardware e um software. O hardware em fase de desenvolvimento, interliga a peça ativa do Topobo ao microcontrolador Arduino conectados através de comunicação

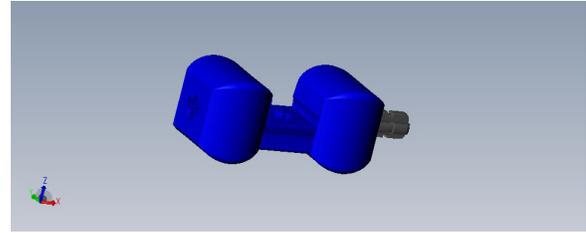


Figure 3. Detalhe de uma peça passiva no ambiente 3D

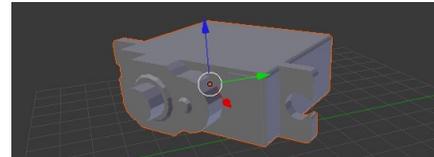


Figure 4. Representação da peça ativa do Topobo

serial *uart* com o controlador atmega 644 do Topobo. O software é um conjunto de bibliotecas desenvolvidas na linguagem Python que recebem os dados do Arduino e os grava em arquivos texto. Os registros são manipulados por este conjunto de bibliotecas e permite através destas, fornecer informações para o simulador físico. O simulador físico tem a representação das peças ativas e passivas em 3D, modeladas no *Blender*. O simulador físico tem como funcionalidade possibilitar a experimentação modelada em 3D da estrutura real, e dar carga a estrutura física através da recuperação dos registros tratados pela biblioteca em Python.

A Figura 3. mostra uma representação de uma peça passiva a qual foi inspirada para ser incorporada ao ambiente experimental em 3D.

O ambiente experimental incorpora as 9 nove peças passivas do Topobo. Na Figura 4. tem-se a representação da peça ativa que foi base para a forma desenvolvida para o ambiente.

O sistema conta com uma tela inicial, onde o usuário tem disponíveis as nove peças passivas que compõem o kit robótico Topobo assim como é permitido integrar a estrutura a peça ativa, a qual tem como característica principal a memória cinética. O usuário pode então através do conjunto de entrada mouse/teclado arrastar as peças para a área de construção das estruturas centrada na tela do ambiente. As peças passam a representar uma estrutura bastante semelhante a formada pelo Kit Topobo real. A inserção da peça ativa, o motor, vai dar o movimento ao sistema, possibilitando a rotação da estrutura virtual no ambiente, represente o mesmo conjunto de movimentos das peças reais. O sistema fica restrito a movimento rotacional de 170 graus em função da constituição eletro mecânica da peça. No momento em que se está construindo uma estrutura no ambiente, é possível altera-la através da remoção de peças. A utilização de funcionalidades como mover para cima, mover para baixo, mover para esquerda e direita



Conecta a energia

Pressiona o botão para gravar

Alterar o movimento do eixo

Pressione o botão Para reproduzir o movimento

Figure 2. Programação manual do Topobo

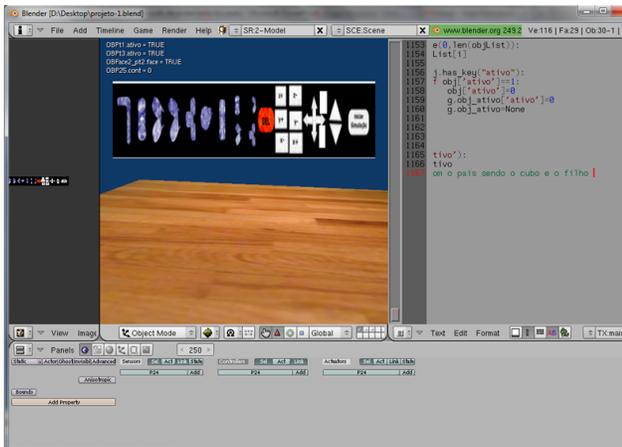


Figure 5. Ambiente para desenvolvimento de estruturas Topobo 3D

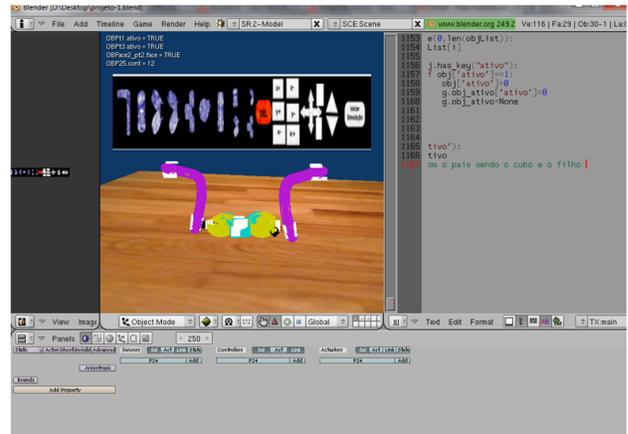


Figure 6. Ambiente com uma representação Topobo 3D

também estão disponíveis. Quando esta funcionalidade é requerida o sistema estrutural inteiro precisa responder ao comando, fazendo com que as conexões definidas pelo usuário também se alteram ao novo movimento. “A ascensão das mídias digitais na última década tem mudado a prática de design, fornecendo ferramentas para a tomada de objetos e construções que lembram as criaturas vivas, modelados com curvas complexas e formas, mantendo-se nitidamente artificial. Superfícies adquiriram profundidade, tornando-se densas substâncias complexas equipadas com suas próprias identidades e comportamentos, membranas flexíveis são encaixadas com redes digitais e mecânicas”. [6]. A Figura 5. mostra a tela inicial com as funções descritas. O ambiente ainda em fase de desenvolvimento incorporará novas funções quando integrados com a biblioteca que faz a conexão com o Topobo Virtual como Topobo Real. É mostrada na Figura 6. do ambiente uma construção do Topobo Virtualizada.

II. A LÓGICA FUZZY

A lógica Fuzzy foi desenvolvida por Zadeh em 1965 para representar conhecimento incerto e impreciso [10]. Esta lógica fornece uma maneira aproximada, mas efetiva de descrever o comportamento de sistemas que são muito complexos, mas definidos ou de difícil análise matemática. Zadeh tinha como objetivo flexibilizar a pertinência de elementos em conjuntos que não possuíam fronteiras bem definidas. Para isso, ele criou o conceito de grau de pertinência. Dessa forma, um elemento poderia pertencer parcialmente a um dado conjunto. A ideia básica em controle fuzzy é modelar as ações a partir de conhecimento especialista, ao invés de, necessariamente, modelar o processo em si. Isso nos leva a uma abordagem diferente dos métodos convencionais de controle de processos, onde os mesmos são desenvolvidos via modelagem matemática dos processos de modo a derivar as ações de controle como função do estado do processo. A motivação para esta nova abordagem veio de casos onde o conhecimento especialista de controle era disponível, seja por meio de operadores ou de projetistas,

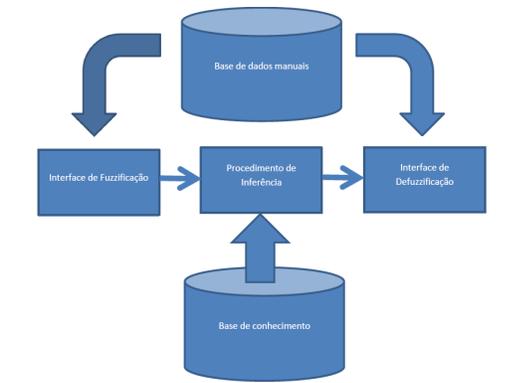


Figure 7. Estrutura básica de um controlador Fuzzy

e os modelos matemáticos envolvidos eram muito custosos, ou muito complicados para serem desenvolvidos. A estrutura de um processo controlado por um controlador fuzzy é mostrada na Figura 7. Enfatizando-se seus componentes básicos: a interface de fuzzição, a base de conhecimento, a base de dados, o procedimento de inferência e a interface de defuzzição.

A interface de fuzzição toma os valores das variáveis de entrada, faz um escalonamento para condicionar os valores a universos de discurso normalizados e fuzzifica os valores, transformando números em conjuntos fuzzy, de modo que possam se tornar instâncias de variáveis linguísticas. A base de conhecimento consiste de uma base de regras, caracterizando a estratégia de controle e suas metas. A base de dados armazena as definições necessárias sobre discretizações e normalizações dos universos de discurso, as partições fuzzy dos espaços de entrada e saída e as definições das funções de pertinência. O procedimento de inferência processa os dados fuzzy de entrada, junto com as regras, de modo a inferir as ações de controle fuzzy, aplicando o operador de implicação fuzzy e as regras de inferência da lógica fuzzy. A interface de defuzzição transforma as ações de controle fuzzy inferidas em ações de controle não-fuzzy. Em seguida, efetua um escalamento, de modo a compatibilizar os valores normalizados vindos do passo anterior com os valores dos universos de discurso reais das variáveis. Após a inferência da ação de controle fuzzy, é necessária a determinação de uma ação de controle não fuzzy que melhor represente a decisão fuzzy, para ser efetivamente enviada ao controle. Apesar de não haver nenhum procedimento sistemático para a escolha da estratégia de defuzzição, as mais comuns incluem: o critério do máximo (MAX), que escolhe o ponto onde a função inferida tem seu máximo, a média dos máximos (MDM), que representa o valor médio dentre todos pontos de máximo quando existe mais de um máximo, e o método do centro de área (CDA), que retorna o centro de área da função inferida [2].

4) *Os Conjuntos Fuzzy*: Segundo Russell, a teoria dos conjuntos Fuzzy é uma maneira de especificar o quanto um objeto satisfaz a uma descrição vaga[9]. Em [1] afirma que a força da Lógica Fuzzy deriva da sua habilidade em inferir conclusões e gerar respostas baseadas em informações vagas, ambíguas, incompletas e imprecisas. Neste aspecto, os sistemas de base Fuzzy têm habilidade de “interpretar” mesmo possuindo informações incompletas sobre o ambiente onde o agente está inserido. Seu comportamento é representado de maneira muito simples e natural, levando à construção de sistemas compreensíveis e de fácil manutenção. Por meio desta característica, torna-se possível a representação da imprecisão e da incerteza em valores linguísticos [5]. Na programação manual foram detectadas as variáveis deslocamento e velocidade as quais foram fuzziçadas em dois conjuntos. Com relação ao conjunto da variável deslocamento tem-se o seguinte: $Da(i) =$ muito perto, perto, médio, longe, muito longe onde $Da(i)$ é o deslocamento do ativo i é uma variável que trata do deslocamento ao qual o programador infere na peça ativa do Topobo, e a qual refletirá no movimento da estrutura projetada. A variável velocidade também foi fuzziçada e trata da velocidade com que o eixo da peça vai sem movimentar, esta variável esta relacionada com variação angular em função do tempo. $Va(i) =$ lento, muito lento, parado, rápido, muito rápido A variável $Va(i)$ está fuzziçada para representar a intenção do programador em relação a velocidade do deslocamento da estrutura. Estes conjuntos são formados por Variáveis Linguísticas que possuem valor expresso qualitativamente por um termo linguístico e quantitativamente por uma função de pertinência [7]. Estas variáveis podem conter modificadores que alteram seu valor. Esses modificadores podem amenizar ou intensificar o seu valor, alguns exemplos mais comuns são: “muito”, “pouco”, “não muito”, “mais ou menos”. Então, para descrever certos fenômenos relacionados ao mundo sensível, temos utilizado graus que representam qualidades ou verdades parciais ou ainda padrões do melhor (na linguagem sofista). Caso em que podemos exemplificar nos conceitos de rápido, alto, fumante, infeccioso, presa, etc.

Para expressar tais conceitos é muito comum o uso de elementos qualitativos ao invés de valores quantitativos. Outros elementos típicos incluem “mais ou menos”, “alto”, “não muitos”, “médio”, etc. Estas ideias são capturadas pela definição de variável linguística. Uma variável linguística tem por característica assumir valores dentro de um conjunto de termos linguísticos, ou seja, palavras ou frases. Desta forma, podemos assumir instâncias numéricas, uma variável linguística assume então instâncias linguísticas. No sistema Topobo por exemplo, uma variável linguística Deslocamento poderá assumir como valor um dos membros do conjunto esquerda ,centro, direita. Para se atribuir um significado aos termos linguísticos, associa-se cada um destes a um conjunto fuzzy definido sobre um universo de discurso comum veja Figura 8. É comum expressar o conhecimento por meio de

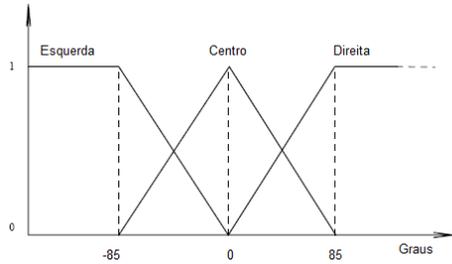


Figure 8. Variável linguística deslocamento

regras do tipo condição x ação. Nestas, um conjunto de condições descrevendo uma parcela observável das saídas do processo é associado com uma ação de controle que irá manter ou levar o processo às condições de operação desejadas. Tipicamente, uma condição é uma proposição linguística, envolvendo variáveis linguísticas, sobre o valor de alguma das variáveis de entrada. De modo análogo, uma típica ação de controle é uma descrição linguística, como por exemplo gire mais para esquerda. A ideia geral aqui é se representar o conhecimento por meio de um conjunto de regras nas quais as condições são dadas a partir de um conjunto de termos linguísticos associados às variáveis de saída x entrada do processo[2] As ações de manipulação ou as saídas movimentos da estrutura Topobo são expressadas de modo similar para cada variável de controle. Regras do tipo se-então são frequentemente chamadas de declarações condicionais fuzzy ou simplesmente regras fuzzy. Dependendo de o propósito ser controle ou modelagem, podem ser chamadas ainda de regras de controle fuzzy ou regras de modelagem fuzzy.

A. IV. FORMALISMO PARA REPRESENTAÇÃO DO CONHECIMENTO NO PROCESSO DE RECOMPILAÇÃO

Quando se está programando a peça ativa das estruturas Topobo, algumas variáveis são setadas de modo intuitivo. Exemplos de variáveis que o programador acaba "inicializando" são velocidade e distância, e de forma implícita pode-se ter a aceleração da estrutura. No processo de programação manual do Topobo identifica-se a variável velocidade e a direção do movimento como sendo as duas principais variáveis a serem tratadas no ambiente, pois elas são as responsáveis pela estrutura do Topobo em representar o movimento. Na programação manual, similar a programação verificou-se o surgimento de estruturas lógicas como AND, OR e NOT, quando se tem uma estrutura Topobo montada com múltiplos ativos. A relação AND pode ser verificada quando se tem que mover o ativo 1 e AND ativo 2. Relações do tipo OR são verificadas na programação manual quando existem dois ativos e para dar um movimento pode-se iniciar tanto com o ativo 1 ou OR ativo. Uma estrutura lógica do tipo NOT na programação manual descreve por exemplo

uma restrição de movimento onde determinada ação não pode ser representada em função de restrições físicas (NOT ESQUERDA). O registro representa que para a esquerda o ativo não pode ser movimentado. Um exemplo do processo é dado no seguinte exemplo em que a peça ativa tem restrição de movimento em 170° graus no processo de recompilação, a partir do centro do eixo que se move 85° , determinados pelo conjunto fuzzy Da(i) o qual dá a direção do movimento do eixo.

B. CONCLUSÕES

Este artigo apresentou uma ferramenta para a extração e registro da programação manual, o kit Topobo e sua memória cinética. Na sequência foi mostrado a interface de captura de registros que são extraídos das peças ativas do Topobo. Apresentou os principais conceitos da lógica Fuzzy, os quais direcionam a modelagem da informação neste trabalho. E por fim a proposta de formalização fuzzy para as variáveis manipuladas da decompilação a qual foi validada através de testes.

C. PERSPECTIVAS FUTURAS

O trabalho visa interligar a biblioteca de comunicação em desenvolvimento para então iniciar os testes de fuzzificação das variáveis deslocamento e velocidade da peça ativa, validando-a em situações reais de uso da ferramenta. Fazer o refinamento das peças modeladas dentro do ambiente 3D permitindo uma melhor visualização através da aplicação de texturas e novas funções ao ambiente. E por fim, fazer a análise das variáveis fuzzificadas no contexto pedagógico, investigando as relações cognitivas das crianças quando submetidas a tarefas que necessitam de um raciocínio lógico-formal, permitindo assim fazer um registro de como as crianças "pensam".

AGRADECIMENTOS

Agradeço ao Programa de Pós Graduação em Educação em Ciências: Química da Vida e Saúde, pelo apoio no desenvolvimento deste trabalho.

REFERENCES

- [1] FABRI, J. A. ; RISSOLI, V. R. V. ; CAMARGO, H. A. *Um Sistema Especialista Fuzzy Aplicado a Classificação de Arquitetura de Computadores*, In: VIII Semana de Informática da Universidade Federal da Bahia, 2000, Salvador - BA. SEMINFO 2000 - VIII Semana de Informática da UFBA, 2000. v. 1. p. 334-340.
- [2] GOMIDE, F. A. C.; GUDWIN, R. R. *MODELAGEM, CONTROLE, SISTEMAS E LÓGICA FUZZY*, 1994, In: SBA Controle & Automação Vol.4 n°3 setembro outubro 1994.
- [3] FREIRE, P. *Pedagogia do Oprimido*, 17rd ed. Rio de Janeiro, Paz e Terra, 1987.

- [4] HODSON, D. *Is this really what scientists do? Seeking a more authentic science in and beyond the school laboratory*, In Wellington, J. (Ed.). *Practical work in school science: Which way now?*. Londres: Routledge, pp. 93-108, 1998.
- [5] KLIR, J. G.; YUAN, B. *Fuzzy Sets and Fuzzy Logic Theory and Applications*. P. Hall, 1995.
- [6] LUPTON, Ellen, *Skin: Surface, Substance + Design*, New York: Princeton Architectural Press, 2002.
- [7] ORTEGA, N.R.S. *Aplicação da teoria de conjuntos fuzzy a problemas da biomedicina*, 2001. 152 p. Tese (Doutorado em Ciências) - Universidade de São Paulo, São Paulo, 2001.
- [8] PARKES, A. J. *Topobo: A Gestural Design Tool with Kinetic Memory*, *Master of Science Thesis in Media Arts and Sciences at the Massachusetts Institute of Technology*, MIT Media Lab, Boston, USA, September 2004.
- [9] RUSSELL, S.; NORVIG, P. *Inteligência Artificial: uma abordagem moderna*, Ed. Campus, 2nd ed Edição. São Paulo, 2003.
- [10] ZADEH L.A. (1975) *The concept of a linguistic variable and its application to approximate reasoning*, *Information Sciences*, Part I : 8, 199-249 ; Part II : 8, 301-357 ; Part III : 9, 43-80.