



COBENGE 2005

XXXIII - Congresso Brasileiro de Ensino de Engenharia

"Promovendo e valorizando a engenharia em um cenário de constantes mudanças"

12 a 15 de setembro - Campina Grande - Pb

Promoção/Organização: ABENGE/UFPG-UFPE

UM AMBIENTE PARA A MODELAGEM, SIMULAÇÃO E SUPERVISÃO DE SISTEMAS A EVENTOS DISCRETOS COM OBJETIVOS DIDÁTICOS*

Angelo Perkusich – perkusich@dee.ufcg.edu.br

Carlos Wagner Barros Araújo – karlos_barros@yahoo.com.br

Tiago Alves Moraes – tiagoeng@hotmail.com

Vinicius Eduardo Ferraz da Nóbrega – viniuciusnobrega@gmail.com

Departamento de Engenharia Elétrica, Universidade Federal de Campina Grande.

Rua Aprígio Veloso, Bodocongó

CEP – Campina Grande – PB

Resumo. *No contexto deste trabalho é apresentado um ambiente para a modelagem, supervisão e simulação de sistemas a eventos discretos (SEDs) com a utilização da plataforma de desenvolvimento do kit Lego Robotics Invention System™. Este ambiente foi desenvolvido com o objetivo de disponibilizar uma ferramenta didática para auxílio ao ensino de SEDs no Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande. O arcabouço teórico para modelagem e controle é a teoria de linguagens formais e autômatos. Além do arcabouço de simulação, foram integradas ao ambiente um conjunto de ferramentas para modelagem e síntese de controladores bem como uma interface de programação para os sistemas simulados. Tais ferramentas possibilitam obter esqueletos de códigos em uma linguagem de programação similar a linguagem C que devidamente alterados são utilizados para controlar sistemas implementados com o kit Lego Robotics Invention System.*

Palavras-chave: automação de sistemas, sistemas a eventos discretos, ensino de engenharia.

1. INTRODUÇÃO

O foco no contexto deste trabalho é apresentar um ambiente para modelagem, análise e simulação de sistemas a eventos discretos utilizando o o kit Lego Robotics Invention System™, e um conjunto de ferramentas computacionais. Tal kit tem sido extensivamente utilizado como uma ferramenta eficiente para ensino de engenharia segundo KLASSNER (2003), GAWTHROP (2004) e WEINBERG (2003). O objetivo central de tal sistema é disponibilizar uma ferramenta didática para auxílio ao ensino de sistemas a eventos discretos no Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande. O arcabouço teórico para modelagem e controle de sistemas a eventos discretos é a teoria de linguagens formais e autômatos como apresentado em CASSANDRAS e LAFORTUNE (1999). Além do arcabouço de simulação, foram integradas ao ambiente um conjunto de ferramentas para modelagem e síntese de controladores bem como uma interface de programação para os sistemas simulados. Tais ferramentas possibilitam obter esqueletos de códigos em uma linguagem de programação similar a linguagem C que, devidamente alterados, são utilizados para controlar dois sistemas implementados com o kit Lego Robotics Invention System.

O restante deste trabalho está organizado da seguinte forma: na Seção 2 são apresentados alguns conceitos básicos relevantes sobre sistemas a eventos discretos e autômatos. Na Seção 3 descreve-se a plataforma LEGO utilizada. Nas Seções 4 e 5 são apresentados dois sistemas simulados com o kit LEGO, seus modelos, e como pode-se supervisionar estes sistemas. Na Seção 6 apresenta-se uma discussão sobre metodologia de ensino. Por fim, na Seção 7 são apresentadas conclusões e perspectivas futuras.

2. SISTEMAS A EVENTOS DISCRETOS E AUTÔMATOS

Para síntese ou análise de um sistema é necessária a concepção de um modelo e ferramentas de análise de tal forma que seja ao mesmo tempo prático e que fique de acordo com o comportamento de tal sistema, sendo assim, precisamos definir funções matemáticas que relacionem a entrada com a saída ao longo do tempo, podendo determinar os parâmetros de funcionamento do sistema, ou seja, seu estado. O estado de um sistema pode ser contínuo ao longo do tempo, ou seja, pode assumir infinitos valores, e por isso ser chamado sistema de estado contínuo, como também pode assumir apenas um conjunto finito de valores e então ser um sistema de estados discretos. O evento é um conceito bastante intuitivo, sua ocorrência ocasiona mudanças no o estado atual de sistema. Um sistema dirigido a eventos funciona de tal forma que a transição de seus estados, ou seja, a evolução de seu comportamento é determinada pela ocorrência de eventos possíveis para cada estado e o conjunto de todos os possíveis estados alcançáveis, no caso de sistemas finitos. Observe que tais eventos podem ocorrer de forma assíncrona a qualquer instante de tempo, e podem ou não ser sincronizados com um relógio.

Um autômato é um dispositivo que é capaz de representar uma linguagem de acordo com regras bem definidas. Segundo CASSANDRAS e LAFORTUNE (1999), podemos considerar um conjunto de eventos E como um alfabeto, seqüências de eventos podem então formar palavras ou cadeias, e uma linguagem é um conjunto de tais cadeias. Um autômato G é definido por conjunto de eventos E , um conjunto de estados X , uma função de transição f , o estado inicial x_0 e pelo conjunto de estados marcados X_m . Os eventos em E levam à transição de estados em G . Todas as cadeias de eventos que podem ser obtidas pelas seqüências de transições de estados de x_0 para qualquer estado de G constitui a linguagem gerada pelo autômato; dentre estas cadeias, aquelas que levam de x_0 até um estado em X_m constituem a linguagem marcada por G . Na Figura 1 apresentamos o exemplo de um autômato, onde O conjunto de eventos $\{a, b\}$ e o de estados $\{q_0, q_1\}$ são explicitados nesta representação gráfica. O estado inicial é o estado q_0 , ele também é um estado marcado. Estados são marcados quando se é desejado atrelar algum significado especial a eles. Por exemplo, para marcar o mesmo como um “estado final”.

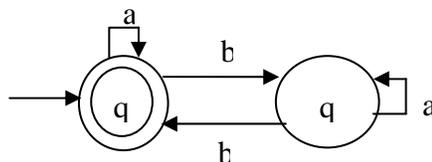


Figura 1 – Autômato exemplo.

O *Desco* (DESCO) é uma ferramenta gráfica que permite editar modelos descritos por autômatos ou redes de Petri, e permite exportar o modelo através da geração de um arquivo no formato *xml*. Este arquivo é utilizado como entrada para a ferramenta *Supremica* (SUPREMICA) que é uma ferramenta para efetuar a composição de autômatos (paralela e pelo produto), sintetizar um supervisor para sistemas a eventos discretos e gerar “esqueleto de código” executável a partir do modelo descrito. Este esqueleto é definido como um conjunto de instruções e relações lógicas em uma linguagem de programação alvo, no nosso caso NQC, com base no comportamento descrito pelo projetista através de um autômato, onde, de forma simples, variáveis denotam estado e funções eventos.

3. PLATAFORMA UTILIZADA

A LEGO, visando obter uma unidade que integra a simplicidade das peças encaixáveis ao projeto de obter uma plataforma de desenvolvimento de robótica fundiu estes propósitos em um kit chamado LEGO Mindstorms. Foi pesquisado durante 15 anos até sua fabricação, através de colaboração entre a LEGO e o MIT (Instituto de Tecnologia de *Massachusetts*) originando o Lego Robotics Invention System™. Possui um tijolo que mascara um microcontrolador, motores, sensores e um conjunto de software que permite criar programas a serem transmitidos a esse tijolo (RCX) via infravermelho. Também inclui fichas de montagens nas quais são explicados os conceitos de lógica e design através de ilustrações. A programação padrão do robô é feita no PC usando o programa RCX Code ou o ROBOLAB.

O RCX executa os programas quando ativado. Até 3 dispositivos de entradas podem ser ligados ao RCX para reagir ao ambiente para, em seguida, ativar as 3 saídas que podem ser ligadas a motores, lâmpadas, buzinas, etc. O RCX usa um microcontrolador de 8 bits da Hitachi (H8/3297) com *clock* de 16MHz. Tem 16Kbytes de ROM, 512 bytes de SRAM (para o *firmware*) e 32Kbytes de SRAM externo para seus programas. O RCX também aloja um compartimento para bateria. Uma visão do microcontrolador presente no kit se encontra na Figura 2.

Do ponto de vista de implementação a linguagem utilizada é NQC (Not Quite C), específica para o sistema de robôs de Lego utilizado. É basicamente a linguagem C, com facilidades que permitem o desenvolvimento de programas para interface como sensores e atuadores do kit.

Utiliza-se como ambiente de programação padrão o *Brick Command Center* (*BricxCC*), que foi obtido pela internet através do site <http://bricxcc.sourceforge.net/> (01/04/2004), e trata-se de um ambiente integrado de desenvolvimento para programar o RCX. Ao iniciarmos o aplicativo temos um meio de programação usual com uma tela de edição de programas e ferramentas para compilação e *download* de aplicativos para o microcontrolador.

4. APLICAÇÃO I: BRAÇO MECÂNICO E VEÍCULO AUTÔNOMO

Inicialmente foi projetada uma unidade de distribuição de objetos na qual foram utilizados os sensores e atuadores do kit, visando obter um perfil do nível de detalhamento que se pode obter na simulação de linhas de montagem.



Figura 2 – Visão geral do RCX junto com os sensores e atuadores

O objetivo foi obter um classificador de bolas, cujo diagrama é apresentado na Figura 3, onde uma garra determinava o tamanho da bola e um veículo autônomo classificava a cor do objeto para, com a informação do tamanho (transmitida via infravermelho), depositar a bola num dado local pré-estabelecido. Na Figura 4 uma foto da montagem é apresentada.

4.1 Descrição do funcionamento e modelo utilizado

O braço mecânico espera até que um sensor, que informa a presença de bolas na entrada, seja pressionado por um usuário. Uma vez detectada uma bola, o braço mecânico inicia uma série de movimentos a partir da posição de repouso para então segurar a bola. O grau de fechamento da garra irá determinar o tamanho da bola. Após o processo de detecção, a bola é levada até o veículo autônomo. Este processo só ocorre se a unidade de controle do braço mecânico perceber que o veículo autônomo está na posição de recebimento, pois a rotina descrita só poderá ocorrer se houver um destino para depositar a bola.

Instantes depois de depositar a bola, a unidade de controle do braço envia uma mensagem pela porta de infravermelho informando o tamanho da bola e que esta, efetivamente, foi colocada no veículo que então realiza diversas leituras no sensor de luz que identifica a cor da bola. O sensor de luz só apresenta a capacidade de distinguir cores muito diferentes entre si, por isso a escolha de bolas pretas e brancas. Isso se deve a forte sensibilidade à luz ambiente para a distinção de mais de duas cores. Com estas duas informações é selecionado o local de depósito. O veículo detecta a posição na qual será deixada a bola pela contagem das marcas pretas fora do percurso preto que é percebido por outro sensor.

A modelagem por autômatos da unidade de controle se encontra na Figura 5. Neste modelo, descrito a partir do *Descó*, o estado inicial é definido corresponde ao indicado pela seta menor que não se origina em nenhum estado como na Figura 1, enquanto que o que se encontra dentro do círculo é o estado marcado considerado.

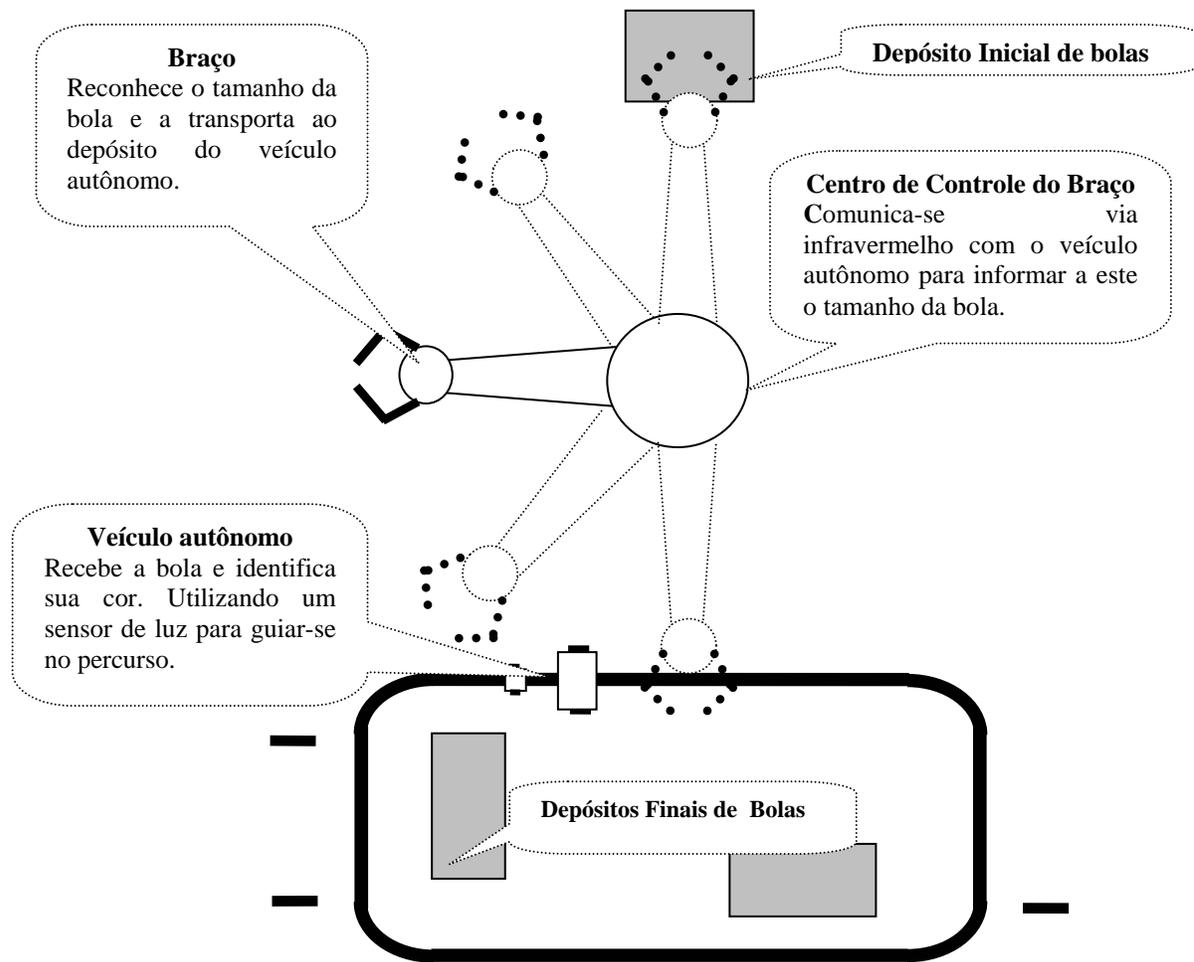


Figura 3: Diagrama para o sistema classificador de bolas

Neste modelo deve ser considerado, na notação dos estados que: *Besp* representa a garra na posição de espera e *Bcbo* significa braço com bola. Analogamente devemos considerar em relação ao carro autônomo *Cesp* e *Ccbo*. *EntSB* representa entrada sem bolas, enquanto que *EntCB*, o oposto. O estado de manutenção corresponde ao momento onde o sistema precisa de revisão de baterias.

Quanto aos eventos: *PressEnt*, significa que o sensor detector de presença de bolas foi pressionado; *PegarBola*, que o braço pegou uma bola da entrada; *DepBolanoCarro*, representa o depósito da bola pelo braço no carro; *BracoRetPosEspera*, que o braço retornou à posição de espera; *RetornodoCarro*, é um evento que se explica pela sua própria denominação; *CarrocomBateriaBaixa* e *BracomBateriaBaixa* representam que houve queda da bateria a um nível próximo da insuficiente ou no carro ou no braço. Esta aplicação constitui uma primeira estrutura que pode ser utilizada para testes de vários modelos além do mostrado.

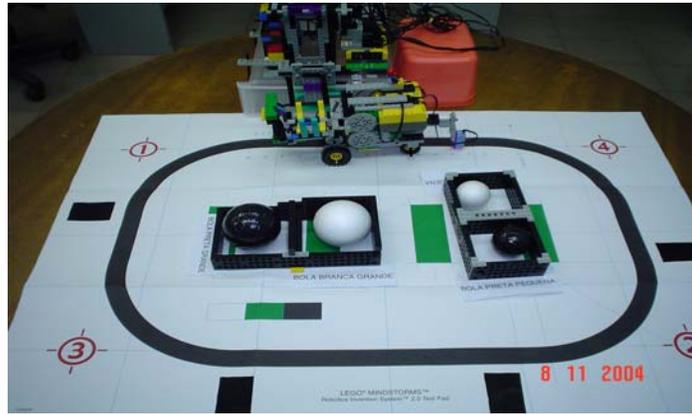


Figura 4: Foto da montagem do classificador de bolas

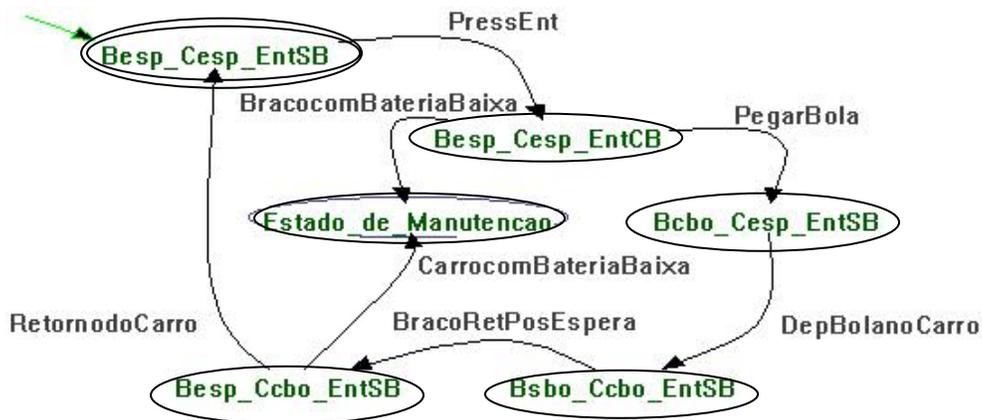


Figura 5: Autômato de controle da aplicação

5. APLICAÇÃO II: CÉLULA DE MANUFATURA

Para esta aplicação o objetivo é simular uma célula de manufatura para fins didáticos, utilizada no curso. Para isto foi construída uma estrutura, utilizando peças do kit, como mostrado no diagrama da Figura 8, que pode ser dividida nas seguintes partes: (i) detector de eixos: detecta a colocação de eixos sobre o trilho para então comandar a movimentação deste; (ii) trilho: transporta os eixos até as posições para colocação da carenagem e de prensa; (iii) prensa: fixa a carenagem aos eixos; (iv) braço: transporta a carenagem; (v) bandeja: posição de espera da carenagem, e; (vi) fornecedor de peças: posição de fornecimento da carenagem. O produto final da célula é um carro construído com peças do kit LEGO. O carro é constituído de dois eixos (conectados as rodas) e carenagem. O procedimento de montagem é realizado da seguinte forma: (i) o braço mecânico retira a peça do estoque e leva para a seção de pintura; (ii) a peça é depositada na caixa de pintura; (iii) a caixa de pintura é rotacionada para a área de pintura; (iv) é realizada a pintura, a caixa de pintura é rotacionada novamente retornando a sua posição inicial; (v) o braço mecânico retira a peça da caixa de pintura e leva para a esteira; (vi) o operador da máquina coloca os acessórios na peça pintada; (vii) a peça é finalmente prensada.

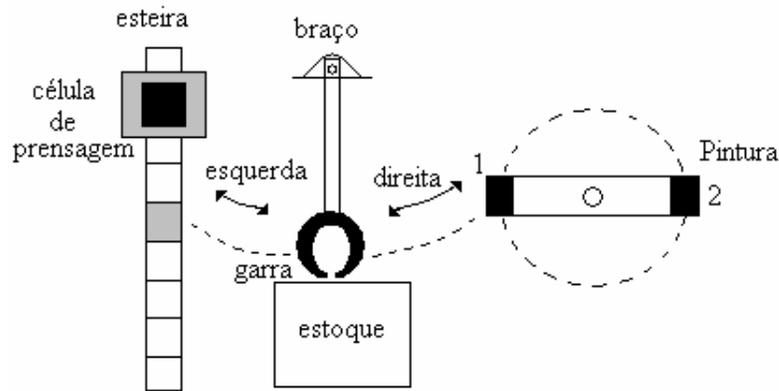


Figura 8 – Esquema da célula de manufatura

5.1 Modelagem da Célula de Manufatura

A modelagem do sistema foi obtida através da construção dos modelos para componentes individuais do sistema completo, os quais são compostos para obter o modelo global. A seguir são apresentados os modelos destes componentes, com exceção do modelo do estoque de entrada que assume-se ter capacidade infinita.

Modelo para a Bandeja de Pintura: A seção de pintura possui duas posições, uma onde se deposita a peça proveniente do estoque (*buffer1*) e a outra onde se realiza a pintura (*buffer2*).

Modelo para o Braço Mecânico: O braço possui três posições; *central*, *esquerda* e *direita*. Em cada posição, a garra pode estar aberta ou fechada.

Modelo para a Esteira: Como apresentado na Figura 11 a esteira possui três estados: parada, movendo para frente e movendo para trás.

Modelo para a Prensa: O modelo para a prensa é mostrado na Figura 12, a prensa pode ou estar em repouso ou comprimindo a peça.

Conjunto de Estados

P_i: peça do buffer *i* sendo processada

E: peça do buffer *i* na fila

0: sem processamento ou peça na fila

Conjunto de Eventos

eb1 – Entrada de peça no *buffer1*

eb2 – Entrada de peça no *buffer2*

Pb1p – Peça no *buffer1* pronta

Pb2p – Peça no *buffer2* pronta

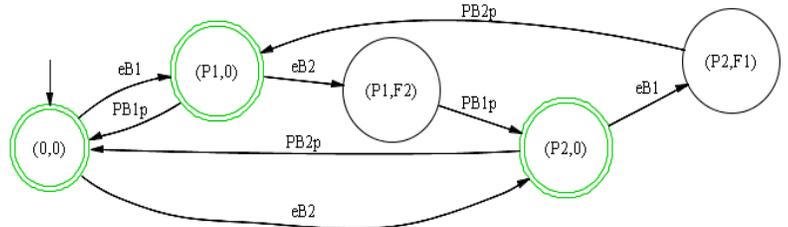


Figura 9: Modelo para a bandeja de pintura

Conjunto de estados:
C: posição central
E: posição esquerda
D: posição direita
A: garra aberta
F: garra fechada
Conjunto de Eventos
c: vai para o centro
e: vai para esquerda
d: vai para a direita
a: abre a garra
f: fecha a garra

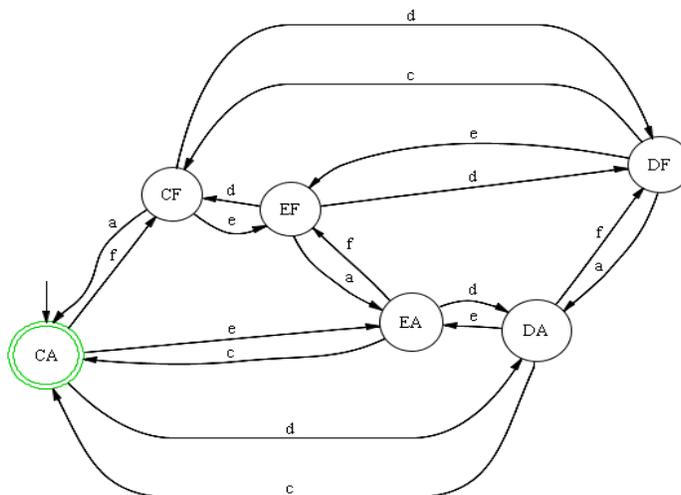


Figura 10: Modelo para o braço mecânico

Conjunto de Estados
P: esteira parada
F: esteira indo para frente
T: esteira indo para trás
Conjunto de Eventos
p: parar
f: mover para frente
t: mover para trás

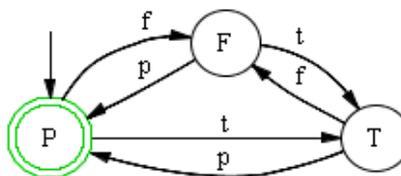


Figura 11: Modelo para a esteira

Conjunto de Estados
P: prensa na posição final (comprimindo)
R: prensa na posição inicial (retorna da prensa)
Conjunto de Eventos
c: comprimir
r: retornar

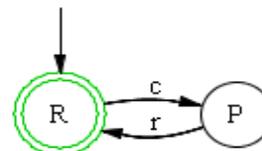


Figura 12: Modelo para a prensa

O modelo global do sistema de manufatura é obtido, utilizando a ferramenta *Supremica*, pela composição paralela dos modelos apresentados e foi omitido devido ao grande número de estados. A seguir apresentamos a implementação para o supervisor.

5.2 Implementação do Supervisor

A especificação do sistema a ser implementado deve satisfazer as seguintes restrições: (i) não tirar peça da esteira; (ii) não colocar peça no estoque; (iii) não abrir e fechar a garra, continuamente, numa mesma posição. A planta foi simplificada para incluir apenas a supervisão do braço mecânico e da garra. Os eventos relacionados ao braço mecânico abstraem as seguintes ações quando é realizado o controle desse componente: (i) direita (rotaciona para a direita – refere-se a posição da célula de pintura); (ii) esquerda (rotaciona para a esquerda – refere-se a

posição da esteira); (iii) centro (rotaciona para o centro – refere-se a posição do estoque). Os eventos relacionados à garra abstraem as seguintes ações quando é realizado o controle desse componente: (i) abrir garra (desce braço, abre garra, levanta braço); (ii) echar garra (desce braço, fecha garra, levanta braço);

Considera-se que a operação do braço é iniciada com o mesmo na posição central, com sua garra aberta, numa altura adequada na qual evite a colisão com as outras células de manufatura em caso de movimentação do braço. O braço deve, então, descer até determinada altura que garanta pegar a peça de forma segura, fechando-se a garra, elevando-se o braço novamente para assim deslocar a peça para a célula de pintura. Para depositar a peça na célula de pintura, o braço deve mais uma vez descer, abrir a garra e retornar a altura adequada. Enquanto o processo de pintura é realizado, o braço permanece parado com a garra aberta. Após ser pintada, o braço retira a peça do setor de pintura e encaminha-a para a esteira. Depositada a peça na esteira, finalmente, o braço retorna para a posição inicial.

A partir dos estados definidos, pode-se definir o autômato que realiza a especificação estabelecida como apresentado na Figura 13.

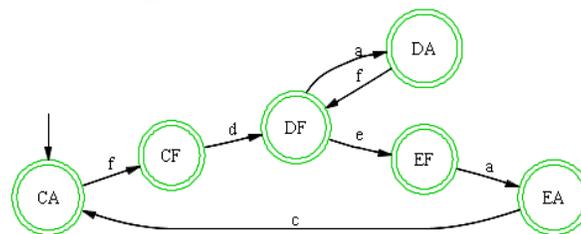


Figura 13. Especificação do Sistema.

O resultado da composição por produto entre o autômato do supervisor e o da planta será o próprio autômato do supervisor, sendo os estados marcados do autômato do sistema sob supervisão os estados marcados da planta como mostrado na Figura 14..

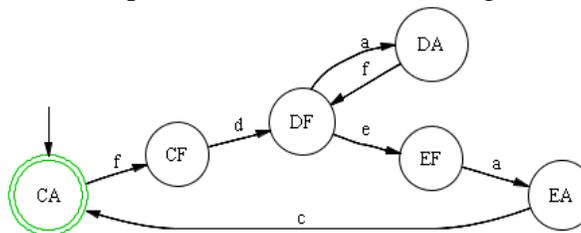


Figura 14. Autômato do sistema sob supervisão.

O autômato gerado pelo Descó, inserido no Supremica para geração do código é mostrado na Figura 15. Apenas a nomeação dos estados e eventos é diferente. A partir deste autômato um esqueleto de código em linguagem NQC é gerado. Este esqueleto, juntamente com uma interface de programação de aplicação dedicada ao sistema de manufatura e pequenas alterações no código é utilizado para controlar a planta de forma a forçar o comportamento descrito pelo autômato apresentado na Figura 15. O programa resultante foi omitido deste artigo por restrições de espaço.

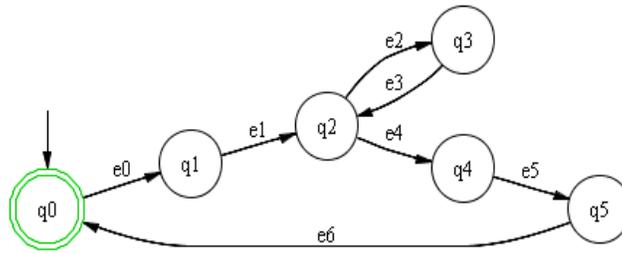


Figura 15. Autômato do sistema sob supervisão utilizado no Supremica.

6. METODOLOGIA DE ENSINO

Com o objetivo de definir um procedimento metodológico do ponto de vista didático, inicialmente definiu-se uma planta da célula de manufatura, projetada com informações essenciais quanto ao gerenciamento da estrutura. Junto com esta descrição, foi fornecida uma pequena API específica para programação da estrutura automática. Uma vez apresentados estes recursos, que foram de teor básico para aplicação das técnicas de projeto de sistemas a eventos discretos, foi definida como tarefa inicial a modelagem do sistema completo sem o supervisor, o que ocasionou a elaboração de modelos extensos. Logo após foram definidos os locais onde os microcontroladores ficariam distribuídos no sistema para a implementação dos supervisores como apresentado nas seções anteriores. Então foi definida, como segunda tarefa, a modelagem do supervisor do braço mecânico para aplicação das várias disposições de modelos de controle e, mais adiante, nas outras partes da célula de manufatura. Cada modelo foi aplicado considerando-se o modelo maior da planta descrito anteriormente, o que fez com que cada grupo utilizasse projetos diferentes, mas procurando obter o mesmo comportamento dinâmico para o sistema.

7. CONSIDERAÇÕES FINAIS

No contexto deste trabalho descreveu-se um ambiente para modelagem, simulação e supervisão de sistemas a eventos discretos utilizando um *kit Lego Robotics Invention System* e um conjunto de ferramentas computacionais de suporte. O foco central na implementação de tal ambiente foi disponibilizar um arcabouço com fins didáticos para utilização no ensino de sistemas a eventos discretos em cursos de graduação.

Foram apresentados os projetos executados utilizando o ambiente construído voltados para a concepção de técnicas de supervisão de sistemas a eventos discretos no que se refere à abordagem de problemas de automação industrial. A aplicação I foi catalogada em termos de formação da estrutura física para reconstituição em casos em que seja necessária a montagem para estudos de caso. A aplicação II foi também desenvolvida para este mesmo fim, mas foi tomada como referência para o desenvolvimento de um meio para utilização em atividades práticas de laboratório no contexto de ensino de sistemas a eventos discretos

Um aspecto importante que deve ser enfatizado é o fato de que programa executável obtido com o objetivo de controle ser efetivamente executado pelo microcontrolador do kit LEGO, o limita a velocidade de execução e o tamanho do código. Isso torna necessário a obtenção de um terminal de trabalho, onde é interpretado o modelo no computador, que é tomado como uma unidade central, para este gerenciar os sensores e atuadores da aplicação através de comunicações sem fio, para permitir mobilidade à aplicação. O infravermelho presente apresenta limitações que

serão melhoradas em trabalhos posteriores. Porém, para atividades de ensino de laboratório, foi possível constatar que o sistema disponibiliza um meio físico próprio para simulações a baixo custo.

Os experimentos aqui apresentados foram utilizados pelos alunos da disciplina Sistemas a Eventos Discretos do Curso de Graduação em Engenharia da Universidade Federal de Campina Grande, no período 2004.2. Os alunos puderam assim, além dos aspectos teóricos tradicionalmente considerados no curso, experimentar com a implementação, via síntese automática de esqueletos de código usando autômatos, de programas implementando a supervisão da célula de manufatura, como apresentada na Seção 5. A situação apresentada consistia na modelagem dos autômatos do braço mecânico, da esteira, da bandeja e da prensa. De posse das funções da linguagem NQC e dos aplicativos de software *Descos* e *Supremica*, os alunos foram orientados a modelar o autômato de cada componente da célula separadamente. Em seguida, executaram a composição paralela dos componentes para obter o modelo global do sistema ou da planta. A seguir os alunos projetaram e sintetizaram um autômato supervisor, definido a partir de uma especificação como mostrado na Seção 5, que controla a célula de manufatura. Este autômato supervisor foi implementado com o intuito de restringir o comportamento da célula, eliminando palavras indesejadas do funcionamento da planta completa. Por fim, os alunos testaram e depuraram os códigos no laboratório, sendo possível verificar na prática os conceitos teóricos.

AGRADECIMENTOS

Este trabalho tem o suporte parcial do CNPq através do projeto 504057/2004-9, da CAPES e Programa PIBIC da UFCG.

REFERÊNCIAS BIBLIOGRÁFICAS

CASSANDRAS, CHRISTOS G.; LAFORTUNE, STÉPHANE. **Introduction to Discrete Event Systems**. Boston, United States: Kluwer Academic Publishers, 1999.

NQC, <http://bricxcc.sourceforge.net/> (acessado em 01/04/2004)

DESCO, <http://www.s2.chalmers.se/software/desco/> (acessado em 01/08/2004)

SUPREMICA, <http://www.supremica.org/> (acessado em 01/08/2004)

KLASSNER, F.; ANDERSON, S.D., LEGO mindstorms: Not just for K-12 anymore, IEEE Robot. Automat. Mag., vol. 10, no. 2, pp. 12–18, 2003.

GAWTHROP, Peter J.; MCGOOKIN, Euan, A LEGO-Based Control Experiment, IEEE Control Systems Magazine, October 2004, 43-46

WEINBERG, J.B; YU, Xudong, Robotics in education: Low-cost platforms for teaching integrated systems,” IEEE Robot. Automat. Mag., vol. 10, no. 2, pp. 4–6, 2003.

AN ENVIRONMENT FOR MODELING, SIMULATION AND SUPERVISION OF DISCRETE EVENT SYSTEMS WITH DIDACTIC OBJECTIVES

Angelo Perkusich– perkusich@dee.ufcg.edu.br
Carlos Wagner Barros Araújo– karlos_barros@yahoo.com.br
Tiago Alves Moraes – tiagoeng@hotmail.com
Vinicius Eduardo Ferraz da Nóbrega – viniciusnobrega@gmail.com
Departamento de Engenharia Elétrica, Universidade Federal de Campina Grande.
Rua Aprígio Veloso, Bodocongó
CEP – Campina Grande – PB

Resumo. *In this paper an environment for modeling, simulation and supervision of discrete event systems using as a development platform the Lego Robotics Invention System™ kit is introduced. This environment was developed to be used as a didactic resource to help teach a discrete event systems course for the Undergraduate Course in Electrical Engineering from Federal University of Campina Grande. The theoretical framework for modeling and control of discrete event systems is the automata and formal languages theory. Besides, de simulation environment, a set of computational tools have been integrated to help students to build, simulate, an synthesize controllers, as well as a programming interface for the simulated systems. O arcabouço teórico para modelagem e contrle de sistemas a eventos discretos é a teoria de linguagens formais e autômatos. Such tools allowed the generation of C language like program skeletons, that could be changed and then used to control systems simulated using the Lego Robotics Invention System kit.*

Keywords: automation, discrete event systems, engineering teaching.