MODELAGEM E ANÁLISE DO PADRÃO IEEE 1451 POR MEIO DE MÁQUINAS DE ESTADOS FINITOS

*Tiago da Silva Almeida [1], Alexandre César Rodrigues da Silva *[2], Silvano Renato Rossi[3]*

Abstract *— The incompatibility and inflexibility of the protocols employed in the industrial networks. It's made a new standard interface between transducer (sensor and actuators) connected in network, denominated IEEE 1451. That provides the interoperability and the way as it should operate the system fieldbus, through hardware and software specifications. IEEE 1451 is divided in committees, which each one specifies standards for different forms of communication of the transducers disposed in network. However as other electronic circuits, the specifications as much of hardware as of software of IEEE 1451 can be modeled, analyzed and synthesized in the finite state machine form, caning be graphically or through algorithms that use hardware description language. Therefore, the life cycle of the application processor in the standard IEEE 1451 was modeled and analyzed graphically as finite state machines described by the model of Mealy and Moore.*

Index Terms —Finite State Machine, IEEE 1451.

INTRODUÇÃO

As redes de transdutores adotam basicamente as mesmas características utilizadas nas redes de computadores pessoais. Comparando as redes industriais às redes de computadores, as redes industriais apresentam algumas restrições mais rigorosas, tais como a velocidade de transmissão de dados, confiabilidade e imunidade a ruídos.

O rápido desenvolvimento de microprocessadores e de novas tecnologias de rede tem tornado as redes de transdutores (sensores e atuadores) soluções atrativas economicamente para aplicações que exigem medição e controle.

Atualmente existem diversas opções de protocolos e tecnologias de redes industriais para os vários níveis de integração em uma empresa, onde a seleção de determinada rede deve ser fundamentada nas características quantitativas e qualitativas da rede, dos equipamentos e da filosofia adotada pela empresa. O que torna tal decisão complexa e difícil em função da quantidade de dados e características a serem consideradas. Essa complexidade e abrangência podem estar influenciando na disseminação de diferentes tecnologias, já que, em geral as empresas responsáveis pela implantação de redes industriais escolhem somente a tecnologia que lhe é conveniente conforme é apresentado em [1].

Com a existência de múltiplas incompatibilidades de protocolos de rede, o número de interfaces de transdutores e a quantidade de esforço necessária para suportar a variedade de protocolos são enormes para usuários e fabricantes de transdutores.

Uma possível solução poderia ser uma interface comum de transdutores para minimizar esses problemas. Em vista dessa situação, o comitê técnico de tecnologia de sensores do IEEE (*Institute of Electrical and Eletronics Enginneer*), em parceria com a NIST (*National Institute of Standards and Technology*), homologou um padrão de interfaceamento de transdutores conectados em rede, denominado IEEE std. 1451 [2].

As especificações do padrão IEEE 1451, como outros circuitos eletrônicos, podem ser modeladas, analisadas e sintetizadas, através de máquinas de estados finitos.

Máquinas de estados finitos ou máquinas sequenciais são representações abstratas de um circuito sequencial real. Seu comportamento pode ser descrito como uma sequência de eventos que acontecem em instantes de tempo discreto.

Máquinas de estados finitos são usadas em vários momentos do nosso dia a dia, como por exemplo, em controle de elevadores, semáforos, etc. Em muitos sistemas digitais existe a necessidade de armazenar e efetuar operações lógicas e/ou aritméticas, tais sistemas podem ser representados (modelados) como máquinas de estados finitos [3].

Para realizar a extração dessas máquinas de estados, torna-se necessário a utilização de ferramentas adicionais. Essas ferramentas já vem sendo desenvolvido em vários trabalhos ao longo dos anos como, em [4] e [5].

A partir da descrição de uma máquina de estados finitos é possível a implementação dessas especificações em qualquer dispositivo reprogramável, como FPGAs (*Field Programmable Gate Array*), microcontroladores e CPLDs (*Complex Programmable Logic Device*) utilizando sintetizadores disponíveis comercialmente, como o ambiente Quartus II da Altera.

A proposta deste trabalho é o estudo e análise da modelagem das especificações de *software* do padrão IEEE

1Tiago da Silva Almeida, DEE, FEIS, UNESP - Univ Estadual Paulista, Av. José Carlos Rossi, s/n – Campus III, Ilha Solteira, SP, tiagoalmeida@aluno.feis.unesp.br
2 Alexandre César Rodrigues da Silva, DEE, FEIS, UNESP - Univ Estadual Paulista, Av. José Carlos Rossi, s/n – Campus III, Ilha Solteira, SP, acrsilva@dee.feis.unesp.br
3 Silvano Renato Rossi, Facultad de Ingeniería, Universidad Nacional del Centro de la Pcia. de Buenos Aires, Av. Del Valle 5737, B7400JWI Olavarría, Argentina, srossi@fio.unicen.edu.ar

1451, através de máquinas de estados finitos. A fim de aumentar o desempenho de uma determinada rede industrial, por meio de uma futura implementação em dispositivos reprogramáveis.

O PADRÃO DE IEEE 1451

O objetivo do padrão IEEE 1451 é definir um conjunto de interfaces de comunicação comum para conectar transdutores baseados em sistemas micro processados e definir uma ou mais barramento de campo em um ambiente independente [2].

A família do padrão é basicamente divida em duas partes. A primeira define o conjunto de interfaces de *hardware* para conectar transdutores a um microprocessador ou um sistema de instrumentação. A segunda parte define um conjunto de interface de *software* para conectar transdutores a diferentes tecnologias de controle de rede.

O IEEE 1451 também prevê a interoperabilidade do sistema, e a capacidade de *plug and play* dos transdutores [2].

A estrutura básica de blocos do padrão pode ser descrita como ilustrado na Figura 1.

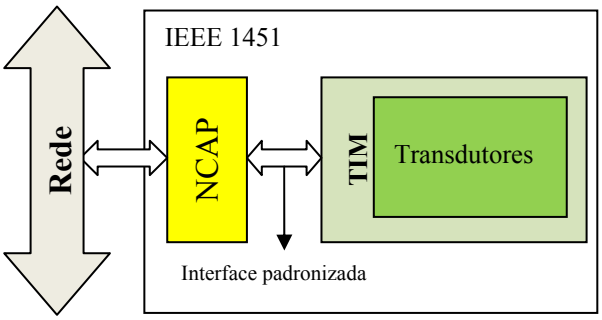

FIGURA. 1
ESTRUTURA DE BLOCOS DO PADRÃO IEEE 1451.

O IEEE 1451 é dividido em comitês, os quais cada um especifica um barramento de campo diferente para conectar transdutores dispostos em rede. A característica chave destes comitês é a definição das TEDs (*Transducer Eletronic Data Sheets*). As TEDs são dispositivos de memória ligados ao transdutor, que armazenam dados dos fabricantes, identificação, calibração, entre outros dados relacionados aos transdutores. De forma que um novo transdutor conectado no barramento de campo possa ser reconhecido automaticamente [2].

O padrão IEEE 1451 é formado pelos seguintes comitês:

- IEEE 1451.0 – define um conjunto de comandos e operações comuns as TEDs para a família do IEEE 1451, de forma independente a comunicação física utilizada no barramento de campo;
- IEEE 1451.1 – define um modelo de objeto comum que descreve o comportamento dos transdutores. Ele define ainda o modelo de comunicação utilizado pelo padrão por meio de um processador capaz de operar em rede, chamado NCAP (*Network Capable Application Processor*), que se comunica com os transdutores através de uma camada física padronizada. O NCAP deve ser capaz de se comunicar também com uma rede externa, como por exemplo, a rede *ethernet*;
- IEEE 1451.2 – define a interface para a comunicação dos transdutores com NCAP e as TEDs para uma configuração ponto a ponto. Os transdutores são partes de um TIM (*Transducer Interface Module*). O TIM contém todo o sistema de aquisição de dados, incluindo os transdutores e o condicionamento de sinal necessário. Em muitos trabalhos desenvolvidos na área é utilizado o padrão RS-232 como exemplo do comitê IEEE 1451.2;
- IEEE 1451.3 - define a interface para a comunicação dos transdutores com NCAP e as TEDs para uma configuração utilizando uma arquitetura de comunicação *multi-drop*. Ele permite vários transdutores dispostos em um mesmo nó de rede;
- IEEE 1451.4 – define uma interface de modo misto para transdutores analógicos com modos de operações digitais e analógicas. O modelo das TEDs também foi reduzido para armazenar o mínimo de dados em um dispositivo de memória;
- IEEE 1451.5 - define a interface para a comunicação dos transdutores com NCAP e as TEDs para uma configuração de transdutores sem fio. Alguns padrões de comunicação sem fio são considerados como uma mesmo interface física, como por exemplo, *WiFi*, *Bluetooth* e *ZigBee*;
- IEEE 1451.7 – esse comitê ainda esta sendo desenvolvido. Ele tenta propor métodos de comunicação de transdutores por meio de sistemas de comunicação RFID (*Radio Frequency Identification*).

O padrão não especifica o tipo de *hardware* ou tecnologia que deve ser utilizada, ele especifica apenas o interfaceamento, o protocolo comum e as especificações das TEDs, podendo ser empregado também em tecnologias futuras.

Ciclo de vida comum aos blocos do NCAP

O estudo do NCAP é de vital importância, já que é a única forma de aquisição e publicação dos dados provenientes dos diferentes TIMs. As especificações que definem o funcionamento do NCAP são previstas no comitê IEEE 1451.1.

Nesse trabalho foram estudados somente os blocos de software no qual o NCAP é divido de acordo com as suas especificações.

Recomenda-se que a programação dos *softwares* que devem compor o NCAP seja orientada a objeto. Dessa forma, é possível padronizar as classes e subclasses que compõem o NCAP, dividindo-as em blocos comuns a qualquer tipo de aplicação. Esses blocos são abordados de forma sistemática pelo padrão IEEE 1451.1 por meio de máquina de estados finitos.

As subclasses do NCAP podem ser subconjuntos de estados para fornecer algum sub-estado e deverão ser associadas as transições contidas dentro de estados definidos por alguma cláusula.

Os mecanismos para a criação e destruição de um bloco estão fora do escopo do comitê IEEE 1451.1.

Um exemplo desses blocos orientados a objetos é o processo de ciclo de vida comum a qualquer bloco que compõe o NCAP, como o apresentado na Figura 2 [6].

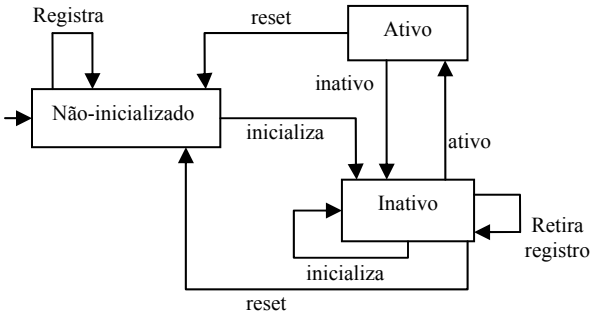

FIGURA. 2
DIAGRAMA FUNCIONAL DO CICLO DE VIDA DOS BLOCOS DO NCAP.

Na Figura 2, cada processo (nó) representa um bloco de classes, que constitui as operações de aquisição de dados, controle do barramento de campo, e outras operações cabíveis ao bloco. O bloco NÃO-INICIALIZADO é o bloco inicial do ciclo de vida, o qual representa as classes que devem ser executadas antes do bloco ser inicializado, o bloco INATIVO representa as classes utilizadas para iniciar as atividades do bloco o e ATIVO representam as classes que são empregadas enquanto o bloco está em funcionamento. Os fluxos denominados INICIALIZA, ATIVO, INATIVO, RESET, REGISTRA E RETIRA REGISTRO, representam os métodos, ou sinais, que controlam o funcionamento e as transições entre cada bloco [6].

Os blocos que são criados durante as operações do NCAP que necessitam ser visíveis na rede deverão ser registrado pelo uso do método REGISTRA no bloco do NCAP local. O registro deverá ser tirado usando o RETIRA REGISTRO no bloco do NCAP, antes o bloco é destruído por mecanismos que não são especificados no comitê IEEE 1451.1. Porém, não são feitas nenhuma exigência no padrão para que os blocos sejam feitos visíveis na rede.

O estado NÃO-INICIALIZADO deverá ser reservado para atividades locais para trazer o objeto do bloco em existência e executando qualquer preparação local necessária para o funcionamento do bloco.

O estado INATIVO deverá ser reservado para atividades como:

- Configuração das propriedades de comunicação do bloco com a rede externa e propriedades de objetos do bloco;
- Inicialização do bloco trazendo o objeto do bloco e objetos de suas propriedades em estado conhecido na preparação para a transição no estado ATIVO;
- Diagnóstico e manutenção do objeto do bloco.

O estado ATIVO deverá ser reservado para atividades relacionadas às funções de aplicação normal do bloco [6].

Comportamento do bloco NCAP

A classe do bloco do NCAP fornece recursos e operação a qual o processador NCAP é capaz de suportar blocos, serviços e manutenção de componentes. Este suporte inclui:

- Registro de um novo bloco;
- Cancelamento do registro de um bloco existente;
- Inicialização e *startup* de qualquer bloco;
- Desligar qualquer bloco.

O bloco NCAP dentro de um processo é a fonte fundamental de configuração de sistema e fornece informações de contabilidade sobre a propriedade do objeto visível na rede externa.

O fornecimento de informações é feito para especificar a identidade de uma ferramenta de configuração para o bloco do NCAP que deseja usar o servidor de nome ou tecnologia similar como partes da configuração do sistema.

A máquina de estados finitos para a classe do comportamento do NCAP é idêntica a classe do ciclo de vida comum a todos os blocos, exceto para o estado INATIVO que possui sub-estados, formando uma hierarquia. Dentro do estado INATIVO existem os estado INICIALIZADO e ERRO. A transição inicial do estado NÃO-INICIALIZADO para o estado INICIALIZADO deverá ser causada por mecanismos específicos dentro do bloco NCAP ou pelo sistema operacional que esteja operando no NCAP. Essa transição inicial não deverá ser invocada por meio da rede externa.

FALHA é uma transição gerada internamente que pode ser executada como uma entrada para o sub-estado ERRO.

Operações do bloco, como ATIVA também podem ser geradas internamente pelo próprio bloco do NCAP. A operação INATIVA deverá causar uma transição para o sub-estado ERRO do estado INATIVO se o bloco NCAP detectar condições de erro que invalide as especificações do bloco.

Na Figura 3 apresenta-se o diagrama funcional do bloco NCAP, conforme já comentado [6].

A modelagem das especificações do padrão pode ser feitos por meio das máquinas de estados propostas no próprio comitê IEEE 1451.1. Porém, para que seja feita uma implementação em *hardware* das especificações de *software* é necessária uma análise mais detalhada em baixo nível de abstração dessas determinadas máquinas de estados.

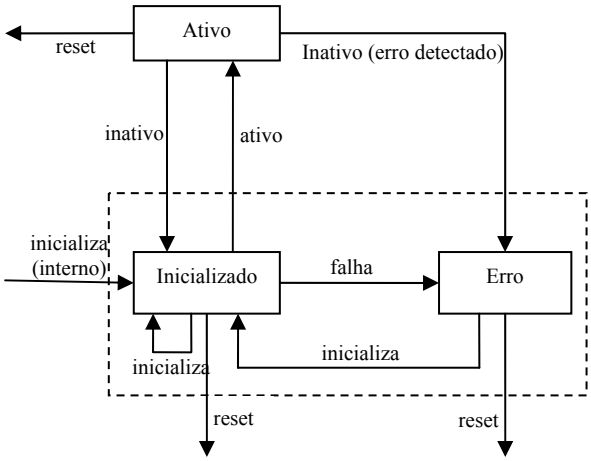

FIGURA. 3
DIAGRAMA FUNCIONAL DO BLOCO NCAP.

MODELAGEM POR MEIO DE MÁQUINAS DE ESTADOS FINITOS

O diagrama de estados é um grafo orientado, no qual cada estado representa um nó, e de cada nó emanam *p* arcos orientados correspondendo às transições de estado. Cada arco orientado é rotulado com a entrada que determina a transição e a saída gerada. As máquinas de estados finitos determinam o próximo estado K(t+1) somente com base no estado atual K(t) e na entrada atual N(t). As máquinas de estados podem ser representadas pela equação 1:

$$K(t+1) = f[K(t), N(t)] \qquad (1)$$

onde *f* é uma função de transição de estados. O valor da saída M(t) é obtido através de duas equações, 2 e 3:

$$M(t) = g[K(t)] \qquad (2)$$

$$M(t) = g[K(t), N(t)] \qquad (3)$$

onde *g* é uma função de saída.

Uma máquina com propriedades descritas nas equações 1 e 2 é chamada de *Máquina de Moore* e uma máquina descrita através das equações 1 e 3 é chamada de *Máquina de Mealy* [3].

Utilizando-se a teoria de máquinas de estados finitos pode-se construir o ciclo de vida comum aos blocos do NCAP apresentado na Figura 2 através do diagrama de estados finitos apresentado na Figura 4.

Esse diagrama representa a máquina de Mealy do bloco definido no comitê IEEE 1451.1. Assim os estado de Un_A até Un_F, representam o bloco NÃO-INICIALIZADO, os estados de In_A até In_F representam o bloco INATIVO e os estado de Ac_A até Ac_F, representa o bloco ATIVO, conforme o apresentado na Figura 2.

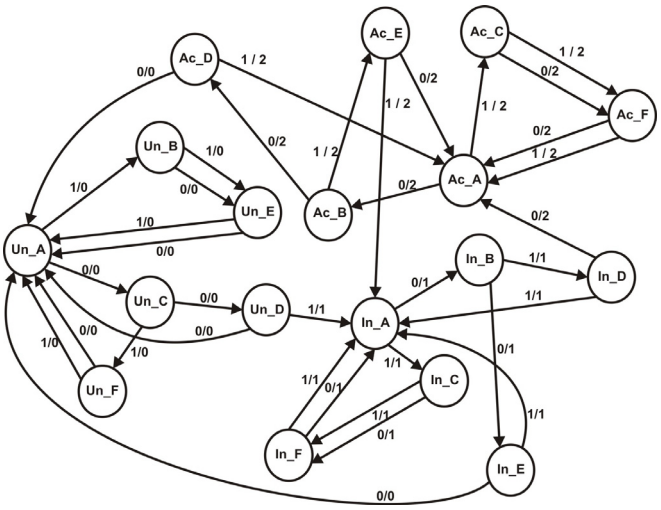

FIGURA. 4
DIAGRAMA DE TRANSIÇÃO REPRESENTANDO O CICLO DE VIDA DOS BLOCOS.

De acordo com o IEEE 1451.1 o bloco NÃO-INICIALIZADO deve ser enumerado como 0, o bloco INATIVO como 1 e o bloco ATIVO como 2, conforme o ilustrado na Figura 4.

A Tabela I ilustra a relação da enumeração utilizada pelo padrão IEEE 1451.1.

TABELA I
ENUMERAÇÃO DOS ESTADOS [6].

Estado	*Valor*
Não-Inicializado	0
Ativo	1
Inativo	2
Reservado	3 - 255

Porém, os fluxos de controle, como o INICIALIZA, o RESET, ATIVO, INATIVO, REGISTRA, RETIRA REGISTRO, não possuem uma enumeração definida no padrão, portanto convencionou-se em utilizar a enumeração em binário de acordo com a Tabela II.

TABELA II
ENUMERAÇÃO BINÁRIA UTILIZADA PARA OS FLUXOS NO DIAGRAMA DE TRANSIÇÃO DE ESTADOS.

Fluxo	*Valor binário*
Reset	000
Inicializa	001
Ativo	010
Inativo	011
Registra	100
Retira registro	101

A Figura 5 representa o funcionamento do bloco NCAP remodelado para uma implementação em *hardware*, por meio da máquina de Moore.

Os estados de In_A até In_F, representam o sub-bloco inicializado, os estados de Er_A até Er_E representam o sub-

bloco erro e os estado de Ac_A até Ac_F, representa o bloco ativo.

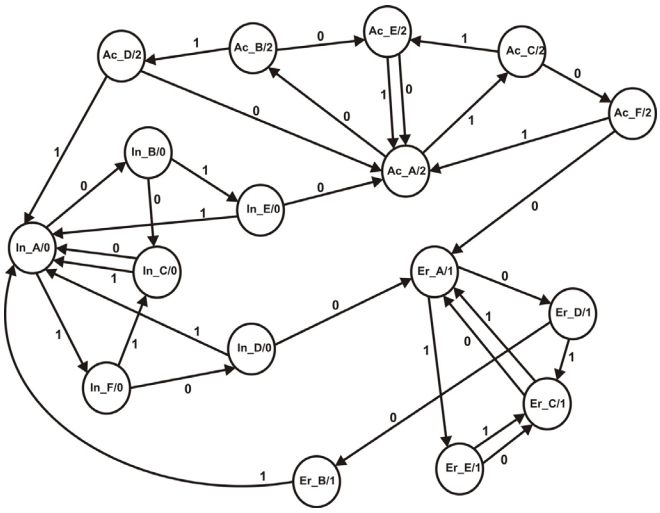

FIGURA. 5
DIAGRAMA DE TRANSIÇÃO REPRESENTANDO O BLOCO NCAP.

A enumeração adotada na modelagem das máquinas de estados finitos para o sub-estados e estados é descrita de acordo com a tabela III.

TABELA III
ENUMERAÇÃO DOS SUB-ESTADOS.

Estado	*Valor*
Inicializado	0
Erro	1
Ativo	2

O padrão não especifica uma enumeração para os fluxos de controle, dessa forma, convencionou-se em utilizar a enumeração em binária apresentada na tabela IV.

TABELA IV
ENUMERAÇÃO BINÁRIA UTILIZADA PARA OS FLUXOS NO DIAGRAMA DE TRANSIÇÃO DE ESTADOS.

Fluxo	*Valor binário*
Reset	000
Inicializa	001
Ativo	010
Inativo	011
Falha e Inativo com falha	100

CONCLUSÃO

Neste trabalho foram estudadas e analisadas as especificações do comitê IEEE 1451.1. O padrão IEEE 1451 vem para tentar sanar os problemas causados pela grande variedade de protocolos e tecnologias empregadas nas redes industriais. O IEEE 1451 fornece especificações tanto de *hardware* como de *software.* A implementação das especificações de *software* em dispositivos reprogramáveis pode aumentar consideravelmente o desempenho do sistema projetado, o que é conveniente em sistema que exigem grande velocidade, como em aplicações de tempo real.

Porém, constatou-se que essas especificações podem ser remodeladas em forma de máquinas de estados finitos, facilitando a sua implementação em *hardware*. Já que o próprio padrão é abordado por meio de máquinas de estados para especificar as classes e subclasses que envolvem a programação do NCAP. Contudo, se faz necessária uma análise mais detalhada do desempenho dessas máquinas de estados por meio de ferramentas de simulação. A implementação física, a fim de validar esta proposta de trabalho também se torna importante.

Outro fator evidenciado no desenvolvimento deste trabalho é a possibilidade de utilização de ferramentas de *software* que possa sintetizar o NCAP aumentando ainda mais o desempenho do sistema.

Nesse contexto, em trabalhos futuros, pretende-se utilizar métodos já desenvolvidos pelo grupo de pesquisa LPSSD (Laboratório de Processamento de Sinais e Sistemas Digitais) para otimização e implementação das máquinas de estados propostas neste trabalho, e analisar o seu desempenho em FPGA.

AGRADECIMENTOS

Os autores agradem a CAPES (Coordenação de aperfeiçoamento de Pessoal de Nível Superior), PPGEE (Programa de Pós Graduação em Engenharia Elétrica) e a FUNDUNESP.

REFERÊNCIAS

[1] CORRÊA, Marco Aurélio; HOUSELL, Marcelo da Silva. The Dissemination of Industrial Filedbuses. In: **CONFERÊNCIA INTERNACIONAL DE APLICAÇÕES INDUSTRIAIS**, 6., 2004, Joinville. INDUSCON. Joinville: 2004. p. 1 - 5.

[2] LEE, Kang. Sensor Networking and Interface Standardization. In: **IEEE INSTRUMENTATION AND MEASUREMENT TECNOLOGY CONFERENCE**, 2001, Budapest, Hungary. IMTC. Budapest, Hungary: 2001. p. 147 - 152.

[3] SILVA, Alexandre César Rodrigues Da. **Contribuição à minimização e simulação de circuitos lógicos**. 1989. 138 f. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, 1989.

[4] DOU, Chie. Integration of SDL and VHDL for HW/SW Codesign of Communication Systems. In: **EUROMICRO CONFERENCE**, 23., 1997, Budapest, Hungary. New Frontiers of Information Technology. Budapest, Hungary, 1997. p. 188 - 195.

[5] WANG, Tsu-Hua; EDSALL, Thomas. In: **VERILOG HDL CONFERENCE AND VHDL INTERNATIONAL USERS FORUM**, 1998, San Jose, USA. Practical FSM analysis for Verilog. San Jose, USA, 1998. 52-58 p.

[6] INSTITUTE OF ELECTRICAL AND ELETRONICS ENGINNEER (USA) (Org.). **IEEE Standard for a Smart Transducer Interface for Sensors andActuators-Network Capable Application Processor (NCAP) Information Model. D**isponível em: <http://ieeexplore.ieee.org/xpls/ abs_all.jsp?tp=&isnumber=4338160&arnumber=4338161&punumber =4338159?tag=1>. Acesso em: 10 ago. 09.