

ROBERTO LUIZ DA CUNHA BARROSO RAMOS

**MODELAGEM DE SISTEMAS DE CONTROLE INTEGRADOS
POR UMA ARQUITETURA BASEADA EM INTEGRONS**

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do título
de Mestre em Engenharia
Mecânica.

SÃO PAULO
2000

**MODELAGEM DE SISTEMAS DE CONTROLE INTEGRADOS
POR UMA ARQUITETURA BASEADA EM INTEGRONS**

ROBERTO LUIZ DA CUNHA BARROSO RAMOS

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do título
de Mestre em Engenharia
Mecânica.

Area de Concentração:
Engenharia Mecânica -
Mecatrônica

Orientador:
José Reinaldo Silva

SÃO PAULO
2000

8732 - 07

Universidade de São Paulo
Biblioteca de Engenharia

BC

A Monique e Carolina.
Juntos, formamos um
integro.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. José Reinaldo Silva, pelo apoio e pela liberdade que me concedeu no desenvolvimento deste trabalho.

Aos professores Paulo Miyagi, Ettore de Barros, Newton Maruyama e Diolino J.

Santos Filho pela atenção constante e pelas discussões proveitosas.

Ao Departamento de Engenharia Mecatrônica, à Universidade de São Paulo e ao

CNPq pela importância dada à pesquisa, como elemento fundamental para o desenvolvimento científico e tecnológico do nosso país e para a independência do nosso

povo.

Aos colegas professores do UNIFIEO Leonardo Pujatti, Marcelo Ramos Martins

e Antônio Guardado pela amizade e apoio em todos os momentos.

ERRATA

Na página 3, **ONDE ESTÁ ESCRITO:** (PETRI, 1962)
LEIA-SE: (MURATA, 1989)

Na página 24, **ONDE ESTÁ ESCRITO:** (REISIG, 1985)
LEIA-SE: (REISIG, 1983)

Na página 27, **ONDE ESTÁ ESCRITO:**
A matriz de incidência $A = [a_{ij}]$ é uma matriz $n \times m$ (lugares \times transições) de inteiros $\{-1, 0, 1\}$, onde cada coluna corresponde à modificação da marcação quando a transição associada é disparada. Portanto, a matriz de incidência reflete a estrutura da rede, definindo as relações de fluxo entre os seus elementos, sendo, portanto, independente da marcação.

LEIA-SE:

A matriz de incidência $A = [a_{ij}]$ é uma matriz $n \times m$ (lugares \times transições) de inteiros, onde cada coluna corresponde à modificação da marcação quando a transição associada é disparada. Portanto, a matriz de incidência reflete a estrutura da rede, definindo as relações de fluxo entre os seus elementos, sendo, portanto, independente da marcação.

Na página 34, **ONDE ESTÁ ESCRITO:** (ARKIN, 1998)
LEIA-SE: (ARKIN, 1990)

Na página 38, **ONDE ESTÁ ESCRITO:**

Na página 60, **ONDE ESTÁ ESCRITO:**
A presença de um componente ou módulo, dentro do modelo de integrons, pressupõe a existência de uma semântica e de um comportamento bem definidos, quase independentes da estrutura global, ou seja, o sistema integrado resulta de componentes individuais que já existiam previamente, com uma funcionalidade específica, segundo um determinado esquema "bottom-up".

LEIA-SE:
A presença de um componente ou módulo, dentro do modelo de integrons, pressupõe a existência de um comportamento bem definido, quase independente da estrutura global, ou seja, o sistema integrado resulta de componentes individuais que já existiam previamente, com uma funcionalidade específica, segundo um determinado esquema "bottom-up".

Na página 62, **ONDE ESTÁ ESCRITO:**

(SILVA; MIYAGI, 1996)

LEIA-SE: (SILVA; MIYAGI, 1996, 1995)

Na página 62, **ONDE ESTÁ ESCRITO:**

(MIYAGI, 1996, 1988, 1985)

LEIA-SE: (MIYAGI, 1996, 1985)

Na página 65, **ONDE ESTÁ ESCRITO:**

(PETERSON, 1981) e (CASSANDRAS, 1993)

LEIA-SE: (CASSANDRAS, 1993)

LEIA-SE:

A rede de integração representada pelos dois boxes e pelas linhas tracejadas da figura 6.6 é obtida a partir das especificações $x_1 + x_5 \leq 1$ e $x_3 + x_4 + x_5 \leq 1$, que restringem as marcações ou estados do integron.

Na página 93, **ONDE ESTÁ ESCRITO:**

Os estados p1 e p3 representam, respectivamente, as sub-redes de energia normal e energia baixa. Os estados p6 e p7 representam, respectivamente, os modos de guiagem segurança com economia de energia (menor profundidade) e seguindo o terreno.

LEIA-SE:

Os boxes p1 e p3 representam, respectivamente, as sub-redes de energia normal e energia baixa. Os boxes p6 e p7 representam, respectivamente, os modos de guiagem segurança com economia de energia (menor profundidade) e seguindo o terreno.

Na página 96, **ONDE ESTÁ ESCRITO:**

de BARROS, E.A. et al. Application of a Cooperative Control System to the Collision Avoidance Guidance of AVVs, Journal of The Society of Naval Architects of Japan, Vol. 178, 1996, pp.685-697.

LEIA-SE:

de BARROS, E.A. et al. Application of a Cooperative Control System to the Collision Avoidance Guidance of AVVs, Journal of The Society of Naval Architects of Japan, Vol. 178, pp. 685-697, 1995.

Na página 97, **ONDE ESTÁ ESCRITO:**

HOPFIELD, J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Natl. Acad. Sci. USA 79, pp. 2554-2558.

LEIA-SE:

HOPFIELD, J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Natl. Acad. Sci. USA 79, pp. 2554-2558, 1982.

SUMÁRIO

RESUMO
ABSTRACT

1. INTRODUÇÃO..... 1

1.1. Motivação..... 1

1.2. Sistemas Complexos..... 4

1.3. Aplicações..... 7

2. TEORIAS E MODELOS BÁSICOS..... 8

2.1. Teoria de Autômatos e Linguagens Formais..... 8

2.2. Teoria de Controle Supervisório (TCS)..... 15

2.3. Teoria de Redes..... 20

2.4. Conceitos da Teoria dos Grafos..... 32

3. A ARQUITETURA BASEADA EM INTEGRONS..... 33

3.1. Definições e Hipóteses..... 33

43

5.5. Interação entre redes com base em invariantes..... 74

5.6. Princípios para uma metodologia de projeto de sistemas de controle integrados..... 77

6. ESTUDO DE CASO: VEÍCULO AUTÔNOMO..... 79

7. CONCLUSÕES..... 95

REFERÊNCIAS BIBLIOGRÁFICAS..... 96

RESUMO

Este trabalho apresenta um modelo para uma classe de sistemas dinâmicos complexos artificiais, baseado em objetos matemáticos chamados *integrans*. Integrans são subsistemas abertos quase independentes (interagem entre si e com o ambiente), definidos recursivamente e organizados de modo "bottom-up" conforme uma arquitetura hierárquica e descentralizada, representada por um grafo do tipo árvore. O conceito de hierarquia é aqui utilizado no sentido de que cada integrans é constituído a partir da integração de alguns componentes e também, por sua vez, pode se integrar a outros componentes, gerando sistemas mais complexos. Em um sistema completamente integrado, um único integrans pode representar todo o sistema, condensando as propriedades e interações realizadas a partir de um conjunto de alguns tipos de componentes primitivos quase independentes.

Os integrans matemáticos emulam o comportamento complexo e autônomo de organismos biológicos multicelulares (integrans biológicos) e interagem entre si e com o ambiente através de restrições de controle e mecanismos de comunicação baseados na teoria de controle supervisorio de sistemas a eventos discretos.

A arquitetura baseada em integrans constitui uma base formal para o projeto de robôs autônomos, sistemas multirobôs, CIM ("Computer Integrated Manufacturing") e...

... Estes sistemas operam normalmente em ambientes não

A arquitetura baseada em integrans e a síntese de um sistema de controle integrado e de seus componentes são ilustradas pelo estudo do comportamento e da emergência de propriedades de um veículo autônomo interagindo com um ambiente complexo. Trabalhos anteriores mostraram o projeto baseado em integrans para sistemas de automação predial.

ABSTRACT

This work presents a model for a class of artificial complex dynamic systems, based on mathematical objects called *integrons*. Integrons are recursively defined open subsystems (interacting with the environment), organized bottom-up according to a tree-like hierarchical and decentralized architecture. The concept of hierarchy is here utilized in the sense of that each integron is constituted from the integration of some components and it can also be integrated with other components, yielding more complex systems. A single integron can represent a whole integrated system obtained from the interactions of some types of almost independent primitive components.

The mathematical integrons emulate the complex and autonomous behavior of multicellular biological organisms (biological integrons). They interact with each other and with the environment through control restrictions and communication mechanisms based on the supervisory control theory of discrete event systems.

The integron-based architecture constitutes a formal basis for the design of autonomous robots, multirobot systems, CIM (Computer Integrated Manufacturing) and intelligent buildings. These systems usually operate in unstructured environments and must possess a rich repertoire of properties and efficient mechanisms of reaction and interaction to accomplish their mission objectives.

Capítulo 1

INTRODUÇÃO

1.1 Motivação

Atualmente existe uma demanda crescente por máquinas e sistemas de automação complexos, com alto grau de autonomia, operando em ambientes pouco conhecidos, de difícil acesso ou perigosos para o ser humano, visando principalmente explorar a terra, o mar e o espaço. Em edifícios e indústrias, o uso desses sistemas visa proporcionar maior conforto e segurança aos usuários, economizar energia e produzir com maior qualidade e

complementação às técnicas utilizadas pela teoria de controle, baseadas nas equações diferenciais e a diferenças.

Considerando o sucesso dessa teoria na modelagem e controle local de sistemas a variáveis contínuas (SVC), amparado em diversas instalações de malhas de controle de

processos ao longo de muitos anos, seria razoável supor que alguns métodos poderiam simplesmente ser estendidos ou adaptados para o tratamento de sistemas mais complexos, utilizando a tecnologia digital hoje disponível.

No entanto, a crescente instalação de sistemas criados pelo homem, em projetos de robótica, manufatura e telecomunicações, mostrou a necessidade de se considerar uma nova classe de sistemas, em que o estado se modifica apenas em determinados instantes de tempo e não, de forma contínua. Estes sistemas são conhecidos como sistemas a eventos discretos (SED).

Além dos SEDs propriamente ditos, a modelagem baseada em eventos discretos

abordagem científica de uma classe de sistemas naturais desenvolvida pela Física, através de leis estabelecidas, e pela Matemática, através do cálculo diferencial e integral, o estudo de SEDs e sistemas complexos ainda se encontra em desenvolvimento, tanto no que diz respeito à base fenomenológica quanto ao formalismo matemático.

Na busca de modelos, formais ou não, para sistemas complexos, algumas

contribuições importantes foram realizadas, incluindo o paradigma da cibernética de Wiener (WIENER, 1948), baseado no controle por realimentação, a teoria de comunicações de Shannon (SHANNON; WEAVER, 1949), os conceitos de máquinas inteligentes e formação de padrões de Turing (TURING, 1936, 1952), a arquitetura da complexidade de Simon (SIMON, 1962), as estruturas dissipativas da termodinâmica de não-equilíbrio de Prigogine (PRIGOGINE, 1977), a sinérgica de H. Haken (HAKEN, 1978), a autopoiese de Varela e Maturana (VARELA; MATURANA; URIBE, 1974), os autômatos celulares de von Neumann, Ulam e Wolfram (VON NEUMANN, 1966) (WOLFRAM, 1984), o modelo de redes neurais de Hopfield (HOPFIELD, 1982), os algoritmos genéticos de Holland (HOLLAND, 1975), a teoria

Portanto, os trabalhos em sistemas complexos têm mobilizado cientistas de diversas áreas, como Física, Matemática, Engenharia Elétrica, Mecatrônica, Robótica, Biologia e Computação. De forma geral, o objetivo comum é tentar explicar, conceituar e modelar sistemas complexos naturais e artificiais, que apresentam, de forma geral, um comportamento inteligente. Infelizmente, conforme observa Wolfram (WOLFRAM,

especialmente a formação de estruturas multicelulares. A intenção é abstrair alguns princípios básicos e leis físicas que regem o comportamento complexo e autônomo dos organismos vivos, utilizando estes conceitos para representação da estrutura e do comportamento de uma classe de sistemas complexos artificiais.

Assim, o presente trabalho pretende dar uma contribuição para o

complexos não podem ser dissociados da realidade física, mas devem resultar dela. naturais e artificiais seja obtido. Objetos matemáticos associados a sistemas autônomos um entendimento genuíno das leis e princípios subjacentes aos sistemas complexos observar que sistemas reais não poderão ser projetados e construídos até que teoria, dificultando a verificação e a generalização dos modelos propostos. Cabe trabalhos compromete o método científico, baseado no ciclo teoria-experimentação-que o problema certamente exige. Por outro lado, a ausência de formalismo em muitos a formação original dos pesquisadores, não apresentando a abordagem multidisciplinar 1994), os trabalhos apresentados até o presente momento refletem, com raras exceções,

Essa classe abrange máquinas e sistemas autônomos, com aplicações em robôs móveis, veículos auto-guiados, CIM e edifícios inteligentes. Estes sistemas apresentam um comportamento complexo, resultante das diversas propriedades que devem possuir para a realização de seu objetivo ou missão e operam normalmente em ambientes não-estruturados, cujas características são conhecidas apenas parcialmente pelo projetista.

Exemplos de comportamento autônomo e complexo já são encontrados em sistemas biológicos, emergindo basicamente em estruturas multicelulares, como a formação de subunidades cooperativas em colônias de microorganismos *Dicystostelium discoideum* (MARÉE et al., 1999) (REIMANN, 1998) (NAGANO, 1998) (SAVILL, HOGEWEG, 1997), muito estudados como paradigma para a formação de oscilações e

primitivos capazes de gerar combinações aceitas pelo ambiente.

Estas características podem ser condensadas através do conceito de *integron biológico*, proposto pelo biólogo francês François Jacob, Prêmio Nobel de Medicina e Fisiologia de 1965, em sua famosa obra "La Logique du Vivant" (JACOB, 1970),

conhecimento apenas parcial do ambiente, talvez seja possível a identificação de alguns componentes primitivos abertos (com capacidade de interagir entre si, conforme as suas afinidades e com o ambiente) e de algumas propriedades locais que possam levar o sistema ao cumprimento de sua missão de forma aceitável. Cada nova propriedade deve resultar de propriedades anteriores que devem ser preservadas, ou seja, a geração de um

Na arquitetura proposta no presente trabalho para sistemas complexos feitos pelo homem, *integrons artificiais* modelados matematicamente por redes de Petri interagem entre si recursivamente para formar integrons de nível superior com novas propriedades, gerando um sistema global que realiza determinadas funções interagindo com o meio ambiente. A estrutura se forma segundo um esquema "bottom-up" em árvore, a partir de alguns tipos de componentes primitivos (folhas da árvore), até atingir o(s) integron(s)

dedicada ao estudo da organização complexa dos seres vivos. O conceito de integron está associado à organização biológica hierárquica em grupos quase independentes e aninhados, formados evolutivamente por afinidades químicas a partir de alguns tipos de elementos primitivos, que deu origem a células, órgãos, tecidos e organismos.

integron estável de nível superior não deve afetar a estabilidade de seus componentes. Cada propriedade local está associada à estabilidade de um integron e esta, por sua vez, corresponde a estados críticos não atingidos por restrições de controle.

A modelagem dos comportamentos locais que viabilizam o integron final é feita com o auxílio dos resultados da Teoria de Controle Supervisório de SEDs (TCS). O formalismo se apoia na teoria de autômatos e na teoria de redes, em particular nas redes de Petri hierárquicas, que são entes matemáticos que incorporam facilidades de distribuição de processamento, estruturação e abstração. Aspectos de composição e comunicação entre redes de Petri relevantes para a arquitetura de integrons também são considerados. A partir da arquitetura baseada em integrons, um método para o "design"

Integrated Manufacturing" (CIM) e automação predial integrada (edifícios inteligentes).

Capítulo 2

Teorias e Modelos Básicos

Este capítulo aborda as teorias e modelos cujos conceitos foram utilizados na formulação da arquitetura de integrons. Compreende a teoria de autómatos e linguagens formais, a teoria de controle supervisorio de SEDs, a teoria de redes e a teoria dos grafos.

2.1 Teoria de autómatos e linguagens formais

Os principais modelos utilizados pela teoria de autómatos são apresentados a seguir, sob a forma de tuplas (KELLEY, 1995) (MIKOLAJCZAK, 1991):

par (q, e) de estado atual e entrada.

- Autômato Finito Não-Determinístico (AFN)

$$M = (Q, E, q_0, F, d)$$

- Q : conjunto finito de estados.

- E : alfabeto de entrada.

- q_0 : estado inicial.

- F : conjunto de estados finais ou aceitos.

- d é uma relação sobre $(Q \times E) \times Q$ e é chamada relação de transição, sendo $d(q,$

$e) \subseteq Q$, refletindo que a partir de qualquer estado q pode-se ter zero ou mais

escolhas para o próximo estado, todas para o mesmo símbolo de entrada. Assim,

d mapeia um par (q, e) em um elemento do conjunto-potência de Q .

- Autômato Pushdown Não-Determinístico (APND)

$M = (Q, E, S, d, q_0, F, z)$

- Q : conjunto finito de estados.

- E : alfabeto de entrada.

$d(q, e, s)$ resulta em um conjunto de pares (p, w) , onde $p \in Q$ e w é o próximo estado e $w \in S^*$ é uma string colocada (ou retirada) na pilha no lugar do símbolo s que ali estava anteriormente. Um dos pares no conjunto é escolhido de forma não-determinística como resultado e o autômato pushdown é alterado para refletir essa escolha.

O autômato pushdown tem, portanto, um mecanismo que permite armazenamento não-limitado, operando como uma pilha, superando a limitação de "memória" finita apresentada pelo AFD e pelo AFN, associada unicamente ao conjunto de estados.

- Máquina de Turing

$$M = (Q, E, T, q_0, b, F, d)$$

- Q: conjunto finito de estados.

- E: alfabeto de entrada.

- T: alfabeto de fita.

- q_0 : estado inicial.

b - T: símbolo esnac

As máquinas de Turing superam a limitação da pilha dos autômatos pushdown, permitindo o acesso a dados abaixo do topo da pilha, o que adiciona um poder de modelagem substancial.

- Máquina de Mealy

$$M = (Q, X, Y, f, g, q_0)$$

- Q : conjunto finito de estados.
- X : alfabeto finito de eventos de entrada.
- Y : alfabeto finito de eventos de saída.
- $f: Q \times X \rightarrow Q$ é a função de transição de estados.
- $g: Q \times X \rightarrow Y$ é a função de saída; $g(q, x)$ define a saída associada com a transição do estado q a partir do evento x .
- q_0 : estado inicial.

As máquinas de Mealy utilizam dois alfabetos, um de eventos de entrada e outro

de eventos de saída, o que permite representar de forma adequada a interação de um

Uma linguagem L , definida sobre um alfabeto E , é um conjunto de seqüências

formado a partir dos símbolos ou eventos em E .

Um autômato pode ser visto como um dispositivo que gera uma linguagem através da manipulação do alfabeto (eventos) de acordo com um conjunto especificado de regras.

Assim, o problema de projeto ou análise de SEDs pode ser visto como a construção de um sistema que fala uma determinada linguagem ou de identificação da linguagem que o sistema fala, considerando o alfabeto da linguagem como o conjunto de eventos do sistema (CASSANDRAS, 1993). Esta abordagem foi utilizada no desenvolvimento da teoria de controle supervisorio, apresentada no próximo item.

Uma forma de descrever alguns tipos importantes de linguagens, ditas regulares,

- Operações comuns sobre conjuntos, como união e interseção.

Convenção-se representar a concatenação de $\{u\}$ e $\{v\}$ por uv , o Kleene closure de $\{u\}$ por $u^* = \{e, u, uu, uuu, \dots\}$ e a união de $\{u\}$ e $\{v\}$ por $(u + v)$. A combinação desses tipos de expressão, como, por exemplo, $(u + v)^*u$, resulta nas

denominadas expressões regulares. Cabe ressaltar ainda o conceito de sublinguagem, que é um subconjunto de uma linguagem.

Expressões regulares permitem uma representação finita compacta para linguagens potencialmente complexas com um número infinito de palavras.

Considerando que um autômato é um dispositivo que gera uma linguagem, cada tipo de autômato definido anteriormente reconhece ou aceita um determinado tipo de linguagem. Se o autômato é denotado por M , a linguagem reconhecida por M é denotada por $L(M)$.

A seguir, são apresentadas algumas relações importantes entre linguagens formais e autômatos:

É possível demonstrar que se L é uma linguagem gerada por algum autômato de estado finito não-determinístico, então existe um autômato finito determinístico que gera

L. Portanto, o AFN não expande a capacidade de geração de linguagens em relação ao AFD. No entanto, autômatos não-determinísticos permitem um meio mais simples de descrever certas linguagens, o que os torna úteis nesse contexto.

Um resultado muito importante a respeito da habilidade de expressão dos autômatos finitos é o Teorema de Kleene, que prova a equivalência entre esses autômatos e as linguagens regulares, isto é, aquelas que podem ser representadas por expressões regulares.

Assim, se uma linguagem é regular, então ela pode ser gerada por um autômato finito e, se ela é gerada por um autômato finito, então ela é uma linguagem regular.

PROUNYCES.

A linguagem gerada por G é denotada por $L(G)$ e é chamada linguagem livre de contexto.

As gramáticas livres de contexto estendem a habilidade de especificação incluindo algumas linguagens que não podem ser reconhecidas por autômatos finitos.

O autômato pushdown não-determinístico definido anteriormente tem a capacidade de reconhecer uma linguagem livre de contexto, aumentando o poder de modelagem em relação aos autômatos finitos.

As máquinas de Turing são mais gerais que autômatos finitos e autômatos pushdown, já que podem reconhecer linguagens regulares, linguagens livres de contexto e também muitos outros tipos de linguagem.

metodologia formal adequada para o controle de sistemas a eventos discretos (SED) (RAMADGE; WONHAM, 1989). A principal vantagem deste modelo é a separação do conceito de dinâmica de malha aberta (planta) do controle por realimentação (supervisor), permitindo a formulação e a solução de uma variedade de problemas de síntese de controle.

conjunto de todas as strings finitas de elementos do conjunto E , incluindo a string vazia

e. Uma string representa uma seqüência parcial de eventos. O conjunto de todos as seqüências fisicamente possíveis é então um subconjunto L de E^* . Este subconjunto de E^* constitui uma linguagem sobre o alfabeto E . Uma string $u \in E^*$ é um prefixo de uma string $v \in E^*$ se para algum $w \in E^*$, $v = uw$. Se v é uma seqüência admissível, todos os

O modelo proposto na teoria de controle supervisorio assume a planta como um sistema dinâmico discreto tal que, para cada estado, um conjunto de eventos pode ocorrer. A planta pode ser vista como um gerador de eventos e o conjunto de todas as seqüências de eventos forma uma linguagem, modelando matematicamente as possíveis execuções do sistema.

Mesmo que uma determinada planta não seja constituída somente por SEDs, um modelo SED simplificado da planta pode ser obtido através de uma função de mapeamento que transforma variáveis contínuas de estado da planta em um conjunto de eventos discretos. A escolha deste conjunto de eventos deve ser feita de forma que o supervisor receba informação suficiente para realizar a sua tarefa.

Para modelar o controle de um SED, a TCS postula que certos eventos do sistema podem ser desabilitados quando desejado e que o conjunto de eventos é dividido em eventos controláveis e não-controláveis: $E = E_c \cup E_n$. Os eventos controláveis em E_c podem ser desabilitados a qualquer tempo por sincronização com o supervisor,

Um gerador G , que modela a planta, é um autômato que consiste de um conjunto de estados Q , um estado inicial q_0 , um conjunto de eventos E e uma função de transição de estados $fte: E \times Q \rightarrow Q$, isto é, G é uma tupla (Q, E, fte, q_0) . A esta definição pode ser acrescido um conjunto de estados marcados Q_m , que representa a execução de determinadas tarefas pelo sistema. Portanto, $G = (Q, E, fte, q_0, Q_m)$. O gerador G é

prefixos de v também o serão. Se $pref(L)$ denota o conjunto de prefixos de strings em L , o modelo SED requer que $pref(L) = L$. Neste caso, L é dita uma linguagem de prefixo fechado. Assim, o comportamento de um SED é modelado por uma linguagem de prefixo fechado L sobre o alfabeto de eventos E e cada elemento de L representa uma possível sequência de eventos do SED.

enquanto os eventos não-controláveis em E_n são aqueles sobre os quais o supervisor não tem influência.

Um supervisor para um SED G é formalmente definido como um mapeamento

$f: E^* \rightarrow 2^E$, tal que $f(w) \supseteq E_n, \forall w \in E^*$. O conjunto $f(w)$ é o conjunto de eventos habilitados pelo supervisor, como função da string w de eventos passados. Uma entrada de controle para G é um subconjunto T de E , satisfazendo $E_n \subseteq T$. Se $\sigma \in T$, então σ está habilitado por T , caso contrário, σ está desabilitado. A condição $E_n \subseteq T$ significa que os eventos em E_n estão sempre habilitados.

O controle de um SED G consiste na geração de uma sequência de elementos em

a) $e \in L(G, f); e$

b) $wu \in L(G, f)$ sss $w \in L(G, f), u \in f(w)$ e $wu \in L \subseteq E^*$.

Assim como a planta G , o supervisor f pode ser representado por um SED S . Neste caso, a ação de controle de S sobre G está implícita na estrutura de transições de

S. As transições desabilitadas por f não aparecem na estrutura de transições de S .

Formalmente, se $s \in L(G, f)$ então $s \in L(S)$ e $sv \in L(S)$ somente se $v \in f(s)$.

Adicionalmente, se $s \in L(G, f)$, $sv \in L(G)$ e $v \in f(s)$, então $sv \in L(S)$. Supõe-se que S e

G executem em paralelo tal que um evento v pode ocorrer quando $S \times G$ está no estado

(x, q) somente se v é possível em S e G nesse ponto. A mudança de estado resultante é

$(x, q) \rightarrow (x', q')$, onde $x \rightarrow x'$ e $q \rightarrow q'$ são as transições em S e G , respectivamente, sob

v.

Desta forma, um supervisor é representado por um autômato com mapeamento

de saída, isto é, $S = (X, E, f, s, x_0)$ é um autômato e $\psi: X \rightarrow T$ é o mapeamento de saída.

quas linguagens prefixais $L, K \subseteq L$ e K é L-controlável sss $\text{pref}(K) \cap L \subseteq \text{pref}(K)$, onde

fechada sss $K = \text{pref}(K) \cap L$, e K é L-controlável sss $\text{pref}(K) \cap L \subseteq \text{pref}(K)$, onde $\text{pref}(K)$ é o conjunto de todos os prefixos das strings em K .

Dado um gerador G , a linguagem $K \subseteq Lm(G)$ modela a especificação desejada.

A linguagem K representa as tarefas a serem executadas sob supervisão. Portanto, o

objetivo é encontrar um supervisor apropriado S para G tal que o sistema em malha fechada satisfaça a condição $L(S/G) = K$.

Ramadge e Wonham provaram que existe um supervisor apropriado tal que $L(S/G) = K$ sss K é $L_m(G)$ -fechada e $L(G)$ -controlável. Quando estas condições não são satisfeitas, é possível obter a suprema sublinguagem controlável $\text{supcon}(K) \subseteq K$, correspondente ao comportamento controlável menos restritivo contido na especificação K , que é sempre computável para geradores de estado finito.

2.3 Teoria de Redes

para auxílio por comunicação visual de forma similar aos fluxogramas e diagramas de bloco.

As redes de Petri podem ser utilizadas por profissionais de aplicação e por teóricos, com benefícios mútuos, através da criação de metodologias formais de projeto de sistemas que resultam em uma melhor adequação da teoria à realidade.

A utilização da teoria de redes na formulação da arquitetura de integrons se deve às suas características favoráveis de representação de sistemas distribuídos, envolvendo concorrência, escolha (conflito), relações de causa-efeito entre eventos e capacidade de abstração em múltiplos níveis.

Algumas dessas propriedades podem ser vistas inicialmente com a ajuda de um modelo simples proposto em (THIAGARAJAN, 1986), conhecido como Sistemas de

- c) A extensão de mudança causada por uma transição é fixa, isto é, não depende do estado em que ela ocorre.
- d) Uma transição está habilitada a ocorrer em um estado se e somente se a extensão de mudança fixa associada com a transição é possível nesse estado.

Na interpretação de condição/evento, um estado distribuído é definido como um conjunto de condições concorrentes, isto é, um subconjunto $c \subseteq B$, denominado "case" e uma transição distribuída é definida por um conjunto de eventos concorrentes, denominado "step". Uma relação de transição específica como "cases" são transformados em "cases" pelas ocorrências de "steps". Isto é obtido na teoria de redes pelo estabelecimento de uma relação fixa de vizinhança entre as condições e os eventos, denominada relação de fluxo.

Formalmente, uma rede é uma tripla $N = (S, T, F)$, tal que:

- a) $S \cup T \neq \emptyset$ e $S \cap T = \emptyset$.
- b) $F \subseteq (S \times T) \cup (T \times S)$.

vizinhança entre os estados locais e as transições locais.

Agora é possível introduzir as noções de Sistema de Redes Elementares e de Rede Condição/Evento. Um Sistema de Redes Elementares é uma quadrupla $N = (B, E, F, c_0)$, onde (B, E, F) é a rede subjacente a N e $c_0 \subseteq B$ é chamado case inicial de N . Uma

Rede Condição/Evento é uma tripla $N = (B, E, F)$, com a seguinte restrição para habilitação de um evento $e \in E$ em um case $c \subseteq B$: $(e \subseteq c) \vee (e \cap c = \emptyset)$, isto é, e pode ocorrer em c se todas as precondições estão em c e nenhuma das poscondições está em c . Impondo a condição da rede ser livre de contato, isto é, $\forall e \in E, \forall c \in B, e \subseteq c \Rightarrow e \cap c = \emptyset$, facilitam-se a classificação de algumas situações fundamentais da teoria de redes e a generalização do modelo.

Algumas situações fundamentais na teoria de redes podem ser definidas com a ajuda do modelo de redes elementares. Em um "case" c de um sistema de redes elementares N , dois eventos e_1 e e_2 podem ser relacionados de pelo menos três formas:

relações conceitualmente, mas também matematicamente e graficamente. Essas relações entre eventos desempenham um papel relevante na formulação da arquitetura baseada em integrons.

Os modelos de redes elementares e de sistemas condigão/evento podem ser entendidos para as redes lugar/transigão, em que os S-elementos podem conter mais do que uma marca, ao contrário das condigões, e as relações de fluxo incluem um esquema de atribuição de pesos (REISIG, 1985) (MURATA, 1989). Outras extensões são possíveis, permitindo a descrição e o estudo de sistemas de processamento da informação concorrentes, assíncronos, distribuídos, paralelos, não-determinísticos e/ou estocásticos.

Genericamente, uma rede de Petri é um tipo particular de grafo orientado, juntamente com um estado inicial chamado marcação inicial, M_0 . Considerando o modelo de redes lugar/transigão, o grafo RP de uma rede de Petri é um grafo orientado,

RP = (P, T, F, W, M_0), onde:

$P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;

$T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transigões;

$F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos (relação de fluxo)

$W: F \rightarrow \{1, 2, 3, \dots\}$ é uma função peso,

$M_0: P \rightarrow \{0, 1, 2, 3, \dots\}$ é a marcação inicial,

$$P \cap T = \emptyset \text{ e } P \cup T = \emptyset$$

Uma marcação M de uma rede de Petri é um mapeamento $M: P \rightarrow I$, onde

$I = \{0, 1, 2, \dots\}$, ou seja, M associa a cada lugar um inteiro não-negativo. Se uma

marcação associa a cada lugar p um inteiro não-negativo k , diz-se que p está marcado

com k marcas. Graficamente, são colocados k pontos pretos dentro do círculo que

representa o lugar p . Matematicamente, a marcação M é um m-vetor, onde m é o

número total de lugares. O p -ésimo componente de M , denotado por $M(p)$, é o número

de marcas no lugar p .

de Petri muda de acordo com a regra de disparo das transições. Apesar de simples, esta regra tem profundas e complexas implicações na teoria de redes de Petri. A regra de disparo das transições se resume no seguinte procedimento:

1) Uma transição t é dita habilitada, se cada lugar de entrada p de t está marcado com pelo menos $w(p, t)$ marcas, onde $w(p, t)$ é o peso do arco de p para t ;

2) Uma transição habilitada pode ou não disparar, dependendo se o evento ocorrer efetivamente ou não;

3) O disparo de uma transição habilitada t remove $w(p, t)$ marcas de cada lugar de entrada p de t e adiciona $w(t, p)$ marcas a cada lugar de saída p de t , onde $w(t, p)$ é o peso do arco de t para p .

Portanto, a partir da regra acima, os sucessivos disparos das transições (e as mudanças resultantes nas marcações) em uma rede de Petri representam a evolução do sistema modelado através dos diferentes estados. Formalmente, a evolução de um

$M' = M + A \cdot \sigma$, onde A é a matriz de incidência da rede e σ é o vetor característico de disparo das transições (indica o número de disparos de cada transição).

A matriz de incidência $A = [a_{ij}]$ é uma matriz $n \times m$ (lugares \times transições) de inteiros $\{-1, 0, 1\}$, onde cada coluna corresponde à modificação da marcação quando a transição associada é disparada. Portanto, a matriz de incidência reflete a estrutura da rede, definindo as relações de fluxo entre os seus elementos, sendo, portanto, independente da marcação.

É importante para a formulação da arquitetura de integrons o fato de que a estrutura de uma rede de Petri, associada a regras gerais de disparo das transições, determina o seu comportamento. Isto acontece porque a rede é constituída por subgratos, que apresentam comportamentos locais determinados, segundo as regras de disparo das transições (locais). A interação entre os subgratos definida pela estrutura da

tipos: as propriedades comportamentais, que dependem da marcação inicial, e as propriedades estruturais, que independem da marcação inicial. Do primeiro grupo, fazem parte algumas propriedades fundamentais para o estudo do comportamento dinâmico do sistema, como alcançabilidade de marcações ou estados e detecção de "deadlocks". As propriedades do segundo grupo são aquelas que dependem das estruturas topológicas

das redes de Petri, como a repetitividade, os invariantes de estados e de transições e a distância síncrona.

A modelagem de um sistema por redes de Petri apresenta algumas vantagens potenciais: o sistema completo pode ser entendido mais facilmente devido à natureza gráfica e precisa do esquema de representação; o comportamento do sistema pode ser analisado através da teoria de redes de Petri, que inclui ferramentas para análise de propriedades, como árvores de marcação e invariantes; a teoria também inclui relações estabelecidas entre certas estruturas de rede e o comportamento dinâmico. Adicionalmente, considerando que as redes de Petri podem ser sintetizadas usando abordagens “bottom-up” e “top-down”, é possível projetar sistemas cujo comportamento

Petri padrão apresentado é efetivo na representação de relações lógicas entre os componentes de um sistema e entre as seqüências de eventos no sistema.

No entanto, uma grande desvantagem das redes de Petri é o problema de complexidade, isto é, os modelos baseados em redes de Petri tendem a se tornar muito

extensos para síntese e análise, mesmo para um sistema de porte modesto. Assim, na aplicação de redes de Petri, é necessário frequentemente adicionar modificações especiais ou restrições inerentes à aplicação em questão. Além da complexidade da representação, outro problema presente na definição original das redes de Petri é a ausência do conceito de tempo, o que restringe a análise dos sistemas modelados aos aspectos qualitativos ou lógicos, como alcançabilidade de estados e "deadlock", impedindo a avaliação do desempenho dos sistemas, o que pode ser relevante em muitos casos.

Para tentar superar essas limitações do modelo original, ampliando o campo de aplicação das redes de Petri, têm sido propostas algumas modificações e extensões. Visando permitir a avaliação do desempenho de sistemas dinâmicos e outras análises

das redes de Petri.

De forma geral, os modelos baseados em redes de Petri estendidas ou não já apresentam sucesso nas áreas de protocolos de comunicação e avaliação de desempenho (MURATA, 1989). Áreas promissoras de aplicação incluem modelagem e análise de

sistemas distribuídos, programas concorrentes e paralelos, sistemas de controle de processos industriais, manufatura flexível, sistemas a eventos discretos em geral, sistemas operacionais e compiladores, linguagens formais e programas lógicos. Outras aplicações interessantes envolvem redes locais, sistemas legais e fatores humanos, filtros digitais, modelos de decisão e redes neurais.

A seguir, são apresentadas algumas relações entre as redes de Petri e as linguagens de prefixo-fechado, utilizadas na teoria de controle supervisorio (TCS):

Seja L uma linguagem sobre o alfabeto E . Seu "prefix closure" é o conjunto L_p de todos os prefixos de strings em L . Uma linguagem é dita fechada se $L = L_p$.

Além dos modelos baseados em autómatos finitos, as redes de Petri podem ser utilizadas para a modelagem de SEDs. Os modelos de Rede de Petri (RP) geralmente utilizados no contexto do controle supervisorio de SED são "bounded" RP, uma subclasse das redes lugar/transição em que o conjunto de marcações alcançáveis é finito.

Assim, a classe de linguagens gerada por esses modelos é equivalente à classe de linguagens regulares e todos os resultados para linguagens regulares podem ser aplicados para "bounded" RP.

Uma extensão natural é permitir um conjunto infinito de marcações alcançáveis. Neste caso, a classe de linguagens gerada por essas redes é um "superset" próprio das linguagens regulares, aumentando o poder de modelagem.

Redes de Petri "unbounded" e máquinas de Turing podem ser utilizadas para modelagem de sistemas e supervisores de estado infinito, quando a linguagem gerada pelo sistema é não regular ou a especificação é uma linguagem de prefixo fechado não

Porém, muitas vezes um aumento no poder de modelagem acarreta um decréscimo no poder de decisão. A escolha mais adequada de um modelo deve observar esta relação de compromisso.

2.4 Conceitos da Teoria dos Grafos

Um grafo é um par ordenado $G = (V, E)$, onde V é chamado conjunto de vértices

e $E \subseteq V \times V$ é chamada relação de incidência. Um grafo $G = (V, E)$ é dito não orientado

se a relação E é simétrica; caso contrário, G é dito orientado.

Uma sequência $d = (v_1, v_2, \dots, v_{n+1})$ de vértices em G é dita um caminho de

comprimento n se $(v_i, v_{i+1}) \in E$ para todo $i \in \{1, 2, \dots, n\}$. Um caminho d é um caminho

simples em G se $\{v_i, v_{i+1}\} \neq \{v_j, v_{j+1}\}$ para todo $i, j \in \{1, 2, \dots, n\}$ tal que $i \neq j$. Um

caminho d é um ciclo se $v_1 = v_{n+1}$. Uma árvore é um grafo que não tem ciclos simples.

Capítulo 3

A Arquitetura Baseada em Integrans

3.1 Definições e Hipóteses

Existem diversas definições para sistemas complexos (PARISI, 1987). Diversos pesquisadores têm buscado princípios ou leis gerais em diversos fenômenos de difícil entendimento, visando formular uma possível teoria de sistemas complexos (WOLFRAM, 1994), que possa explicar de forma unificada o funcionamento de sistemas aparentemente muito distintos entre si, como o cérebro humano e o clima.

Exemplos de sistemas autônomos de comportamento complexo são encontrados na natureza, especialmente em sistemas biológicos, como redes neurais de grandes caso das máquinas autônomas.

Exemplos de sistemas autônomos de comportamento complexo são encontrados

na natureza, especialmente em sistemas biológicos, como redes neurais de grandes

mamíferos, sistemas imunes, regulação por hormônios e sociedades de insetos. No entanto, a dificuldade de estudo e modelagem desses sistemas e o pouco conhecimento adquirido até o momento tornam arriscada a realização de analogias e abstrações para a formulação de metodologias para sistemas artificiais. Outra abordagem de sistemas autônomos, inspirada nas ciências comportamentais (etologia, neuroetologia, psicologia), conhecida como "robótica baseada em comportamento" (ARKIN, 1998) (ANDERSON; DONATH, 1990), vem sendo estudada e testada recentemente com algum sucesso. No entanto, a abordagem comportamental tem poucas chances de apresentar bons resultados na construção de sistemas de alto nível de autonomia, pois é baseada apenas na observação dos procedimentos emergentes, sem modelar os verdadeiros mecanismos geradores da complexidade.

favorável. Dentro da organização, as ameças permanecem constituídas com as suas membranas e, portanto, mantêm o seu meio interno dentro de determinados limites (homeostase). Além disso, cada ameça é constituída por organelas que interagem entre si.

natureza é possível obter um elevado grau de autonomia através da geração de um comportamento complexo caracterizado por diversas propriedades obtidas da integração hierárquica de componentes primitivos simples. O aumento da complexidade é obtido a partir da integração recursiva de componentes diferenciados, gerando novos componentes estáveis que contribuem para a sobrevivência do organismo como um todo.

Portanto, mecanismos de chaveamento e geração de oscilações e pulsos de cAMP pelas células realizam a integração entre elas, ou seja, basicamente o cAMP carrega, comprime e codifica a informação, criando correlações entre estados nas células. Logo, estas de forma geral não se comunicam diretamente entre si e o cAMP age como

As amebas inicialmente idênticas se diferenciam em dois tipos básicos de células, que se organizam em grupos com participação diferente na formação, migração e desenvolvimento futuro da "lesma". Esses grupos se comunicam entre si através de ondas de uma substância química chamada cAMP, mantendo uma certa independência entre si, que pode ser vista como uma homeostase de grupo ou de ordem superior. Já foi verificada a ação do mesmo cAMP na fase de agregação das amebas.

A organização descentralizada e hierárquica dos organismos multicelulares permite reações mais rápidas e mais eficientes a perturbações do ambiente. Este esquema é representado de forma simplificada pelo conceito de *integron*, introduzido em (JACOB, 1970). A noção de organização biológica hierárquica obtida pela interação cooperativa de determinados elementos também é apresentada em (NEDERBRAGT, 1997), associada à evolução e à adaptação.

A homeostase dos integrons está associada à existência de estados críticos que certamente seriam atingidos caso não fosse possível a comunicação e o sincronismo entre os seus componentes. Logo, as propriedades dos integrons resultam dos mecanismos de controle que garantem a sua homeostase e estão associadas ao seu comportamento

AUTONOMIA \Rightarrow COMPLEXIDADE \Rightarrow INTEGRAÇÃO,

onde a integração será entendida como interação com homeostase (estabilidade) gerando novas propriedades.

assim, formular um modelo para sistemas complexos artificiais. Não há qualquer interesse em efetivamente desvendar todos os mecanismos de sinalização utilizados pelas amebas ou qualquer outro sistema natural, já que a inteligência da máquina não necessariamente deve se desenvolver de forma idêntica a dos seres vivos, já que se apoiam em plataformas de hardware distintas. Os modelos computacionais desenvolvidos

O esquema proposto guarda semelhança com alguns pontos da teoria da autopoiese (MATURANA; VARELA; URIBE, 1974), que procura explicar a organização dos seres vivos com base nos conceitos de homeostase, evolução,

Desta forma, considerando que é inviável a especificação completa ou o planejamento de um comportamento complexo em ambiente não estruturado, talvez seja possível obter alguns componentes primitivos abertos e um mecanismo adequado simples de interação recursiva entre eles que possa gerar as propriedades desejadas, assim como o esquema do CAMP utilizado pelos microorganismos *Dicystostelium* pode explicar diversos comportamentos coletivos complexos (agregação, migração em direção a fontes de luz e calor).

com sucesso para explicar o comportamento complexo das amebas *Dicystelium* (MARÉE et al., 1999) (REIMANN, 1998) (SAVILL; HOGEWEG, 1997) são suficientes, neste momento, para a identificação de algumas características relevantes para a autonomia e a complexidade de organismos e máquinas. Conforme observado em (ARKIN, 1990), o estudo de sistemas biológicos bem sucedidos certamente gera conhecimento útil para o "design" de máquinas inteligentes, no entanto, não há necessidade de se criar sistemas artificiais que sejam reproduções exatas de seus correspondentes biológicos nem de se provar teorias específicas de comportamento humano ou animal. Basta que os modelos gerados por estas teorias sejam úteis para a autonomia de máquinas. Portanto, os princípios aqui desenvolvidos para modelar a autonomia de sistemas computacionais permanecem válidos independentemente de novas

interação interna entre estes e da interação com o ambiente externo. Se o comportamento que emerge e pode ser observado indica algum tipo de organização ou realiza um conjunto bem determinado de funções, possivelmente associadas a algum objetivo, pode-se dizer que a interação entre os subsistemas é de natureza cooperativa e que os mesmos estão integrados (RAMOS; SILVA, 1997) (SILVA, 1998).

A seguir, algumas definições e hipóteses relevantes serão apresentadas

informalmente:

Def. 3.1 Sistema aberto, no sentido termodinâmico, é aquele que troca matéria e energia com o meio externo. Genericamente, pode-se dizer que esses sistemas possuem a capacidade de interagir ou de se comunicar com outros sistemas.

Def. 3.2 Homeostase é a propriedade que determinados sistemas (especialmente biológicos) apresentam de que algumas variáveis internas (concentrações de substâncias químicas) são mantidas dentro de determinados limites. Genericamente, pode-se dizer que esses sistemas visitam um conjunto de estados

Def. 3.4 Afinidade entre subsistemas dinâmicos é a característica que define a capacidade e a forma de interação entre eles.

Def. 3.5 Sistemas quase independentes são sistemas abertos que apresentam um comportamento independente na ausência de interação com outros sistemas, mas

interação sucessiva "bottom-up" de componentes quase independentes, segundo uma organização espacial e temporal descentralizada e hierárquica, gerando novos componentes e propriedades, possivelmente segundo objetivos locais subordinados a uma finalidade ou missão global. Cada novo componente gerado é constituído por componentes de nível inferior e pode participar na formação de

Def. 3.7 Integração é a criação de correlações entre variáveis internas e de restrições de controle sobre subsistemas, gerando configurações estáveis (aceitas pelo ambiente externo).

Def. 3.6 Sistema integrado é aquele constituído por um conjunto de alguns tipos de componentes primitivos quase independentes, que funcionam concorrentemente e interagem entre si gerando novas propriedades, possivelmente associadas a um objetivo global.

que permitam a inibição ou habilitação de seqüências de estados/eventos e/ou observação por outros sistemas.

somente um componente de nível superior. O sistema global, no último nível da hierarquia, é constituído pelo acoplamento dos componentes do nível inferior e pode sempre ser reduzido à interação de seus componentes primitivos, no nível mais básico de descrição. A arquitetura descentralizada e hierárquica emula o modelo do *integron* biológico que define a formação de células, órgãos, tecidos e organismos segundo uma complexidade crescente regulada pelas afinidades entre os componentes.

Hip. 3.2 Os componentes primitivos atuam de forma especializada e possuem os recursos do sistema (p.ex., sensores e atuadores), que são acessados de forma mutuamente exclusiva, isto é, um componente só pode utilizar os recursos de

Hip. 3.4 Os componentes primitivos de um sistema complexo artificial podem ser representados por SEDs dinâmicos determinísticos controlados localmente (o controle local faz com que o sistema controlado oscile entre determinados estados, apresentando um comportamento cíclico). Sistemas SVCs acoplados a

controladores locais podem ser aproximados por SEDs, através da associação de eventos à ultrapassagem de limites das variáveis contínuas e discretização dos comandos. Como os recursos são mutuamente exclusivos e os componentes executam funções distintas e especializadas, os conjuntos de estados internos e de eventos dos SEDs primitivos são disjuntos entre si.

Hip. 3.5 A integração de subsistemas SED é realizada pela ordenação de alguns eventos, restringindo o comportamento possível dos mesmos segundo regras que inibem ou habilitam seqüências de eventos em um subsistema em função de condições ou eventos ocorridos em outro subsistema.

provocar falhas irrecuperáveis caso eles funcionassem isoladamente.

No escopo deste trabalho, o termo **sistema integrado** estará sempre associado ao conceito de **sistemas complexos**.

3.2 Integron - o componente básico da estrutura de integração

Considerando as definições e hipóteses formuladas, pode-se introduzir o conceito de *integron artificial* ou simplesmente *integron*, como se segue:

Def. 3.9 Integron é um objeto matemático que representa um componente de um

sistema complexo artificial ou o próprio sistema como um todo. Assim, um integron é um subsistema aberto constituído pela integração de subsistemas mais simples e que possui um conjunto de propriedades que define a sua participação na formação de um sistema integrado global. Cada subsistema que compõe um integron pode ser, em si, um integron ou um componente primitivo.

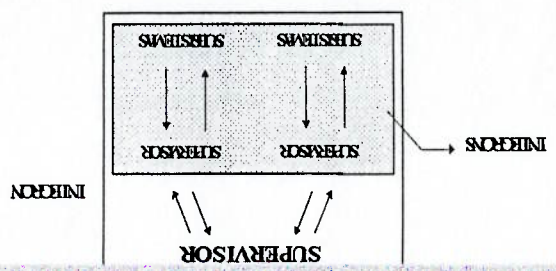


Figura 3.1 Arquitetura hierárquica e recursiva do integron.

3.3 Representação formal da arquitetura baseada em integrons

A arquitetura descentralizada e hierárquica de um sistema integrado complexo pode ser formalizada com base na teoria dos grafos, que fornece ferramentas importantes para a representação de estruturas.

A estrutura de um sistema integrado S pode ser representada por um grafo do tipo árvore, isto é, um grafo conectado, sem ciclos, com N vértices e $N-1$ ramos ou arestas. Cada vértice representa um integron. Na raiz da árvore está o integron que representa o sistema integrado como um todo. Os vértices do nível seguinte, diretamente conectados à raiz, representam os integrons que formam o sistema global. Os níveis

Considerando que nem todos os eventos de um conjunto de integrons são relevantes para a formação de um integron de nível superior, devem ser adicionados ao modelo mecanismos de observação (mapeamentos de abstração) que atuam com os supervisores (mapeamentos de controle), associados a cada vértice da árvore.

As propriedades locais do sistema integrado emergem das regras de integração impostas sobre os eventos e condições dos integrons e da ação dos mecanismos de observação/abstração e de supervisão/controle.

Formalmente, a árvore de integrons que modela o sistema integrado complexo S pode ser representada pela n -upla $S = (I_n, I_{n-1}, P_n, R_n, A_n, S_n)$, representando, respectivamente, o conjunto de integrons de nível n , o conjunto de integrons de nível $n-1$, o conjunto de propriedades associadas aos integrons de nível n , o conjunto dos mapeamentos de interação de nível n , o conjunto dos mapeamentos de nível n e o conjunto de supervisores de nível n . Tem-se, ainda:

Um modelo matemático também baseado na teoria dos grafos, semelhante à árvore de integrons e denominado hiperestruturas foi proposto em (BAAS, 1992). No entanto, Baas não esclarece como estruturas de nível inferior interagem para formar estruturas de nível superior com novas propriedades, indicando uma formalização excessivamente genérica baseada na teoria de categorias, com seus conceitos de objetos

e mortismos. Na arquitetura proposta no presente trabalho, os integrons interagem

através de supervisores, que evitam a entrada em estados críticos, gerando novas propriedades vinculadas a integrons estáveis de nível superior. A ação de controle sobre

a dinâmica dos integrons gera o significado físico da interação, ausente nas

hiperestruturas de Baas.

A representação formal do modelo proposto da árvore de integrons será baseada

na TCS de Ramadge e Wonham e na teoria de redes. Os requisitos de interação com o mundo exterior e com outros integrons, bem como de recursividade, abstração e

hierarquização exigem extensões nas teorias fundamentais, levando à proposta de criação

de uma nova classe de autómatos (autómatos integrados) e de uma nova classe de redes

Capítulo 4

Autômatos Integrados

4.1 Motivação

Como definido no capítulo anterior, o *integrón* é um subsistema capaz de interagir com outros *integróns* e de reagir ao ambiente externo.

A TCS modela o subsistema não-controlado (planta) como um gerador de eventos, representado formalmente por um autômato finito determinístico (AFD). Este

de autômatos, substituindo os autômatos AFD originais por autômatos determinísticos com eventos externos e internos de entrada e saída. Os autômatos que fazem parte dessa nova classe, proposta formalmente neste capítulo, serão designados por *autômatos integrados*.

Uma abordagem semelhante unicamente sob o ponto-de-vista de entradas/saídas pode ser encontrada em (BALEMI, 1993). O controle descentralizado e hierárquico de SEDs a partir da TCS foi apresentado em (LIN; WONHAM, 1990) e (ZHONG; WONHAM, 1990), limitado a poucos níveis específicos (supervisão e coordenação) e utilizando uma representação de eventos inadequada para a integração em sistemas complexos.

Portanto, considerando que em qualquer nível da árvore de integrons, um determinado integron é composto a partir de integrons de nível inferior através de mecanismos de observação e controle (supervisor), gerando propriedades locais, é fundamental modelar formalmente esse processo básico de composição. Este capítulo

define a participação de M_{nj} no contexto do sistema integrado global..

A estrutura resultante é uma n-upla $(M_{nj}; M_{n-1,k}; P_{nj}; S_{nj})$. Por simplificação, essa estrutura será representada pela n-upla (A, I_j, PL, h) , $j = 1 \dots n$, representando a relação de formação do integron A a partir dos integrons I_j de nível inferior. A propriedade local

associada ao integron A é dada por PL e h representa a função de realimentação de estados estática executada pelo elemento de integração (supervisor). Esta função implementa o esquema de comunicação e controle dos integrons.

4.2 Modelo formal dos autômatos integrados

A arquitetura de integrons proposta não é necessariamente formada apenas por subistemas SEDs propriamente ditos e admite que alguns integrons possam estar relacionados a SVCs.

Os subistemas SVC (Sistemas de Variáveis Contínuas) são normalmente

Assim, a arquitetura proposta admite integrons relacionados com SVCs, desde que possuam um supervisor ou elemento de integração capaz de filtrar as informações relevantes para o processo cooperativo, gerando SEDs. A aproximação de SVCs por SEDs pode ser feita pela associação de eventos discretos a determinados limites das variáveis contínuas (HOLCOMBE, 1982).

Portanto, este trabalho considera, no escopo da TCS de Ramadge e Wonham, os subsistemas como SEDs, sejam SEDs propriamente ditos ou aproximações de SVCs, estes possivelmente acoplados a controladores locais.

Inicialmente, os integrons I_j , que formam o integron de nível superior A e representam subsistemas SED controlados localmente (subsistema planta-supervisor local em malha fechada) serão modelados por autômatos integrados, expressos formalmente como se segue:

Def. 4.1 Autômatos Integrados: $I_j = (Q_j, E_j, Y_j, F_j, G_j, q_{j0}), j = 1 \dots n$
- Q_j : conjunto finito de estados internos; q_{j0} : estado inicial;

- $F_j: Q_j \times E_j \rightarrow Q_j$ é a função de transição de estados;

- $G_j: Q_j \times E_j \rightarrow Y_j$ é a função de saída.

Os conjuntos $Q_j, E_j, Y_j, j = 1 \dots n$ são disjuntos entre si.

A modelagem por autômatos integrados expressa as relações de causa-efeito de forma explícita por dois alfabetos distintos, um de entrada e um de saída, como nos modelos de máquinas de Mealy e de Moore. Isto permite separar os distúrbios, estímulos externos e procedimentos internos que afetam o comportamento do integron (entradas) dos eventos que resultam das reações do mesmo (saídas). Assim, este modelo caracteriza o integron como um sistema aberto, capaz de interagir com o meio externo.

Apenas alguns eventos ou estados internos são realmente observados pelo supervisor, caso contrário o número elevado de interações entre este e os integrons-plantas poderia inviabilizar a composição do integron de nível superior, devido aos muitos recursos de comunicação necessários. No entanto, todos os eventos de percepção e de

da arquitetura deve sempre ser compatível com as propriedades correspondentes de nível superior, permitindo que estas se realizem.

Para ilustrar o conceito de abstração e observação de autômatos integrados, considere-se, por exemplo, um sistema que possui dois sensores monitorando um mesmo

ambiente, um de temperatura e um de detecção de fumaça. Os eventos ocorridos de temperatura alta e presença de fumaça (eventos de entrada) resultam em eventos de saída distintos e, no caso de ocorrência conjunta, eles podem ser combinados em uma informação (evento de saída) única de ocorrência de incêndio. Portanto, três eventos de saída são gerados pelo integron em questão: temperatura alta, presença de fumaça e ocorrência de incêndio. No entanto, dependendo do objetivo local de integração (reagir ao incêndio), apenas a última informação pode ser relevante. Desta forma, os dois primeiros eventos de saída seriam omitidos (p.ex., através do nome λ) e apenas o último evento seria observado pelo supervisor. Naturalmente, a ocorrência isolada dos eventos de entrada temperatura alta e presença de fumaça resultaria normalmente em ações internas do integron, isto é, a competência do integron se restringiria ao tratamento dos

1) permitir sintetizar e analisar um subsistema como um componente quase independente de um sistema integrado, através da representação dos seus diversos procedimentos internos;

2) expressar os mecanismos de interação com o ambiente externo e com outros subsistemas, que afetam a seleção de procedimentos internos e permitem a

formação do sistema integrado.

As entradas e saídas relevantes para a interação com outros integrons estão associadas a estados internos que devem ser observados por um supervisor, afetando o comportamento deste. Formalmente, este esquema de interação via controle por realimentação de estados estática pode ser representado pela função parcial h , definida a

seguir:

$$h: Q \rightarrow 2^{X_{int}}, \text{ onde } Q = \Pi Q_j, X_{int} = \cup X_{j,int}, j = 1 \dots n.$$

a seguir:

$$P = \Pi I_j = (Q, E, Y, F, G, q_0), j = 1 \dots n.$$

$$Q = \Pi Q_j, j = 1 \dots n.$$

$$E = \cup E_j = E_{int} \cup E_{ext}, E_{int} = \cup E_{j,int}, E_{ext} = \cup E_{j,ext}, j = 1 \dots n.$$

$$Y = \cup Y_j, j = 1 \dots n.$$

$V_{ext} = X_{ext} = E_{ext} \setminus X_{int} = E_{ext} \cap X_{int} = V_{int} = E_{int} \cap X_{int}$, ou seja, as entradas externas a cada integron I_j associadas a outros integrons componentes de A tornam-se internas ao integron A , enquanto as entradas externas a cada integron I_j associadas ao ambiente formam as entradas externas ao integron A .

$$\Sigma_{j,ext}: Q_j \rightarrow 2^{E_{j,ext}}$$

pode ser representada pelas seguintes funções:

A ocorrência de eventos em um determinado estado interno de um integron I_j

Prova:

integrado:

autômatos integrados I_j através do supervisor h também é um autômato

Proposição 4.1 O integron de nível superior A obtido pela interação dos

$$q_0 = (q_{10}, \dots, q_{n0})$$

$$G(q_1, \dots, q_n, e) = G_j(q_j, e), \text{ se } e \in E_j$$

$$F(q_1, \dots, q_n, e) = (q_1, \dots, F_j(q_j, e), \dots, q_n), \text{ se } e \in E_j$$

No entanto, para que o integron A seja classificado como autômato integrado ainda é necessário provar que existem funções $\Sigma^{ext}: Q \rightarrow 2^{V^{ext}}$ e $\Sigma^{int}: Q \rightarrow 2^{V^{int}}$, que determinam, respectivamente, os eventos externos e internos possíveis em cada estado do integron A. Além disso, as funções F' e G' devem ser obtidas.

As funções Σ^{ext} e Σ^{int} podem ser obtidas como se segue:

$$\Sigma^{ext}(q) = \Sigma^{ext}(q_1, q_2, \dots, q_n) = (\Sigma_{1,ext}(q_1) \cup \Sigma_{2,ext}(q_2) \cup \dots \cup \Sigma_{n,ext}(q_n)) \setminus X_{int} = (\cup \Sigma_{j,ext}^*(q) \setminus X_{int}, j = 1 \dots n, \text{ onde as funções } \Sigma_{j,ext}^*: Q \rightarrow 2^{E_{j,ext}} \text{ são construídas sobre o domínio } Q, \text{ tal que } \Sigma_{j,ext}^*(q_1, q_2, \dots, q_n) = \Sigma_{j,ext}(q_j)).$$

$$\Sigma^{int}(q) = \Sigma^{int}(q_1, q_2, \dots, q_n) = (\Sigma_{1,int}(q_1) \cup \Sigma_{2,int}(q_2) \cup \dots \cup \Sigma_{n,int}(q_n)) \cup h(q), \text{ onde o supervisor } h \text{ é uma função } h: Q \rightarrow 2^{X^{int}}, \text{ como definido anteriormente. De forma}$$

As funções $F'_j: Q \times V \rightarrow Q_j$ são construídas de tal forma que $F'_j(q, v) = F_j(q, v)$, se $v \in E_j$ e $F'_j(q, v) = q_j$, se $v \notin E_j$, onde $q = (q_1, \dots, q_j, \dots, q_n)$.

As funções $G'_j: Q \times V \rightarrow Y_j$ são construídas de tal forma que $G'_j(q, v) = G_j(q, v) = G'_j(q, v)$, se $v \in E_j$, $q = (q_1, \dots, q_j, \dots, q_n)$.

$S = (Q, B^*, X_{int}, r_s, q^0)$, onde Q é o conjunto de estados do supervisor, B^* é o conjunto de strings sobre B , X_{int} é o conjunto de eventos externos a cada integron associados à comunicação com outros integrons pertencentes ao

integrons de nível inferior pode ser realizado pelo autômato S definido a seguir:

onde i_{nj} é o integron j do nível n da arquitetura em árvore (os componentes primitivos formam o nível 0), $i_{n-1,kl}$, $l = 1 \dots p$, são os integrons que pertencem ao conjunto k do nível $n-1$ e o supervisor h_{nj} é implementado por um SED que observa apenas os eventos do

$$i_{nj} = ((i_{n-1,kl}) \parallel_B h_{nj}, l = 1 \dots p,$$

Portanto, o integron A obtido também é um autômato integrado com uma nova propriedade PL e pode participar da formação de um integron de nível superior, de acordo com o esquema recursivo representado pela expressão a seguir:

$$= G_j(q_j, v), \text{ se } v \in E_j.$$

modelagem da emergência de propriedades segundo níveis hierárquicos. No entanto, o modelo de Holland é baseado em sistemas de tempo discreto, cujos estados se alteram sincronamente em intervalos fixos de tempo, o que não representa adequadamente a dinâmica de sistemas complexos artificiais, associada geralmente à ocorrência de eventos ou distúrbios no ambiente. Por outro lado, as noções de estabilidade e de ocorrência de

integração.

onde F^* e G^* são as funções de transição de estados e de saída, respectivamente, do autômato-produto estendidas sobre strings e $\phi: Y^* \rightarrow Y^*$ é uma função de abstração sobre strings de Y que omite os eventos de saída não relevantes para a

$$r_s: Q^i \rightarrow 2^{X_{int}} \mid r_s(r_s(q_0, b)) = h(q), b \in B^*, q \in Q = \Pi Q_i, i = 1 \dots p.$$

$$q = F^*(q_0, x) \mid b = \phi(G^*(q_0, x)), x \in E^*, E = \cup E_i, i = 1 \dots p,$$

$$r_e: Q^i \times B^* \rightarrow Q^i$$

r_e e a função de saída r_s são definidas conforme se segue:
conjunto k e q_0^i é o estado inicial do supervisor. A função de transição de estados

estados críticos associadas às propriedades emergentes, que são centrais na arquitetura baseada em integrons, não são abordadas por Holland, tornando o seu modelo excessivamente abstrato e simplificado, dificultando a aplicação em problemas reais.

O modelo formal de autômatos integrados, apesar de permitir a construção de sistemas de nível razoável de complexidade, apresenta uma dificuldade inerente à representação por autômatos, vinculada à ausência do conceito de estado como um conjunto de elementos distribuídos. Assim, foi proposto um modelo equivalente baseado em redes de Petri, apresentado a seguir.

Capítulo 5

Redes de Petri Integradas

5.1 Modelos matemáticos para sistemas integrados

Como observado anteriormente, os autômatos integrados permitem a modelagem formal de uma classe de SEDs complexos, mas apresentam uma dificuldade inerente à teoria de autômatos para a representação de sistemas concorrentes.

Como o conjunto de estados do sistema global é obtido a partir do produto

utilização de um sistema de representação matemática que já traga na sua formulação básica o conceito de concorrência. No entanto, também é importante que os resultados e modificações da TCS desenvolvidos para os autômatos integrados sejam mantidos, já que eles constituem uma base formal segura para o controle integrado de SEDs.

conceito de modularidade, que difere da idéia simples de elementos concorrentes e distribuídos. A presença de um componente ou módulo, dentro do modelo de integrons, pressupõe a existência de uma semântica e de um comportamento bem definidos, quase independentes da estrutura global, ou seja, o sistema integrado resulta de componentes

Assim, diferentemente do autômato-produto, o estado de uma rede de Petri é representado de forma distribuída segundo seus componentes, através da marcação dos lugares ou condições e, por isso, o crescimento ou a substituição de um componente não afeta a estrutura da rede da mesma forma que na modelagem por autômatos.

Visando alcançar esses objetivos, além de suportar uma arquitetura mais geral para sistemas integrados, propõe-se um modelo com base na teoria de redes. As redes de Petri já trazem na sua conceituação original uma abordagem da concorrência, através de elementos locais distribuídos, que representam estados e transições de estado, segundo uma estrutura de grafo bipartido. As transições de estado locais influenciam os estados dos componentes vizinhos, habilitando ou inibindo as suas transições.

individuais que já existiam previamente, com uma funcionalidade específica, segundo um determinado esquema "bottom-up".

O conceito de componente necessário a uma arquitetura geral de sistemas complexos integrados se assemelha aos conceitos de processo, objeto ou agente desenvolvidos pela ciência da computação, porém ainda sem um formalismo adequado ou aceite de modo consensual. Juntamente com o conceito de componente, surge a necessidade de estabelecimento de algum tipo de esquema de comunicação entre os mesmos, que permita a integração e a emergência de novos comportamentos na estrutura. Algumas propostas importantes surgiram, entre as quais se destacam o CSP ("Communicating Sequential Processes"), de Hoare (HOARE, 1985) e o CCS

hierarquização de módulos (subredes) segundo propriedades desejadas. Uma proposta neste sentido foi apresentada em (DENG et al., 1993) baseada nas chamadas G-Nets, visando, em especial, aplicações em sistemas complexos de informação. Em um contexto mais voltado para sistemas automatizados e visando fornecer uma representação formal para SEDs, esta em desenvolvimento a rede GHENeSys (General Hierarchical Enhanced

Net System) (SILVA; MIYAGI, 1996), inspirada no PFS/MFG (Production Flow Schema/Mark Flow Graph) (MIYAGI, 1996, 1988, 1985). Uma proposta inicial de definição formal da rede GHENeSys, adequada para o escopo do presente trabalho, é fornecida a seguir:

Def. 5.1 Uma rede GHENeSys é uma n-upla $R = (B, BE, A, F, FE, SR, \phi; M_0)$,

onde:

B é o conjunto de boxes ou condições,

BE é o conjunto de boxes externos, também denominados pseudocondições,

cujas marcações estão associadas ao meio externo,

A é o conjunto de atividades ou eventos,

Qualquer rede hierárquica GHENeSys pode ser transformada em uma rede primitiva GHENeSys R_0 , cujos elementos boxes e atividades são atômicos, através da aplicação dos mapeamentos ϕ nos diversos níveis da rede.

O modelo geral proposto para sistemas dinâmicos integrados não considera aspectos de tempo real ou desempenho do sistema, restringindo-se apenas às propriedades lógicas, associadas a sequências de eventos e/ou estados proibidos ou desejados, por isso não houve necessidade de introdução de características de temporização na rede.

As redes de Petri permitem uma representação matemática e também gráfica de sistemas dinâmicos, particularmente SEDs (DAVID; ALLA, 1994).

5.2 Representação de SEDs por redes de Petri elementares

Neste capítulo, os resultados obtidos com os autômatos integrados serão estendidos para a representação de SEDs por redes de Petri e será desenvolvido o modelo formal do *integron* utilizando as chamadas *redes GHENeSys integradas*. Particularmente importante é o resultado obtido que demonstra a recursividade do *integron* no contexto do modelo de redes de Petri proposto.

Um SED M , de comportamento dado por uma linguagem sobre o alfabeto A , pode ser modelado por um sistema de redes elementares baseado em uma rede de Petri C/E (Condição/Evento) (ver cap. 2) rotulada, ou seja, com nomes ou rótulos atribuídos aos eventos ou transições, da seguinte forma:

$$\text{SED } M = (R, m_0, n)$$

$R = (B, E, F)$ - estrutura da rede de Petri C/E associada

m_0 - marcação ou estado inicial

$n: E \rightarrow A$ - função de nomeação dos eventos ou transições

O comportamento fechado de M é dado pela linguagem $L(M) = \{n(v) \in A^* \mid v \in$

Da mesma forma que os autômatos integrados representam explicitamente a interação dos SEDs com o mundo exterior, através de eventos de entrada e saída, a rede de Petri que modela um SED real e, portanto, não isolado, também deve representar essa interação. A rede GHENeSys, baseada nas redes C/E, permite a representação de um

SED aberto representado por um autômato integrado. Uma possível representação é

mostrada abaixo:

$$M = (Q, E, Y, F, G, q_0) \Leftrightarrow R = (B, BE, A; F, FE; SR, \phi, M_0)$$

$$B \text{ (conjunto de boxes)} = Q$$

$$BE \text{ (conjunto de boxes externos)} = E \cup Y, E = E_{ext} \cup E_{int}$$

A (conjunto de atividades) = $\{ a_{qe} : q \in Q, e \in \Sigma(q) \}$, onde $\Sigma(q)$ é o conjunto de

eventos internos e entradas externas possíveis em q .

$$F = \{ (q, a_{qe}) : q \in Q, a_{qe} \in A \} \cup \{ (a_{qe}, z) : a_{qe} \in A, z \in Q, F(q, e) = z \}$$

$$FE = \{ (e, a_{qe}) : e \in \Sigma(q), a_{qe} \in A \} \cup \{ (a_{qe}, y) : a_{qe} \in A, y \in Y, G(q, e) = y \}$$

$$M_0(b) = 1, \text{ se } b = q_0; 0, \text{ caso contrário.}$$

Na definição inicial da rede GHENeSys (def. 5.1), o conjunto de boxes externos representavam apenas pseudocondições associadas a distúrbios causados pelo ambiente externo propriamente dito. Neste caso, a marca de um box externo era substituída ou removida unicamente pelo mundo exterior. Na rede GHENeSys obtida por equivalência com um autômato integrado, o conjunto de boxes externos BE e a relação de fluxo

externo FE também representam os eventos de comunicação e controle associados aos supervisores e os eventos de saída do sistema.

Portanto, algumas entradas representam a interação da rede com o ambiente exterior propriamente dito (interação externa ao sistema global), outras representam os eventos de comunicação da rede com outros integrons através de supervisores (interação externa à rede, mas interna ao sistema global) e também são representados os eventos internos determinados somente pelo estado corrente do sistema. Da mesma forma, as saídas representam respostas para o exterior propriamente dito e respostas de comunicação para outros integrons através de supervisores. Como cada um destes tipos de evento interfere na dinâmica da rede de uma forma particular, é conveniente

subconjunto a ser designado por CE. As saídas de comunicação com outros integrons através de supervisores constituirão o subconjunto $CS \subseteq Y$, de boxes de comunicação de saída.

de "OR-state" e "AND-state" sugeridos em (BRAVE; HEYMAN, 1993) para máquinas de estado hierárquicas são representados nas redes GHENeSys através de elementos estruturais, que realizam a abstração e a concorrência entre estados. A modelagem baseada em elementos e propriedades estruturais permite tratar a

O integron representado pela estrutura do grafo R é um SED M, cuja semântica

cíclica, ou seja, qualquer atividade pode ocorrer novamente (REISIG, 1985).

Como um integron ou um módulo primitivo é um sistema controlado, a rede é

$$\text{onde } BE = GE \cup GS \cup CE \cup CS.$$

$$R = (B, GE, GS, CE, CS, A, F, FE, SR, \phi, M_0),$$

Def. 5.2 Uma rede GHENeSys integrada é uma n-upla

um integron ou componente primitivo, pode ser representada como se segue:

A rede modificada, denominada rede GHENeSys integrada, capaz de representar

Segundo este conceito, dois ou mais sistemas executam de forma paralela e independente, exceto para um determinado conjunto de eventos comuns, cuja execução só ocorre se for possível em todos os sistemas em um determinado instante. Esta operação é conhecida como composição paralela com sincronização ou composição

Por hipótese, os alfabetos dos integrons são disjuntos entre si e somente o

Um integron em qualquer nível da arquitetura proposta é composto pelo funcionamento concorrente e sincronizado de alguns integrons de nível inferior com um supervisor ou elemento de integração.

5.4 Comunicação, Sincronismo e Composição Síncrona de SEDs representados por redes GENesys integradas

complexidade de forma adequada com a utilização de métodos computacionais específicos.

sincrona sobre um determinado alfabeto H de eventos comuns e é simbolizada por III.

Uma definição formal é apresentada em (VOGLER, 1986), porém a composição sincrona sobre um alfabeto H entre SEDs representados por redes de Petri pode ser vista, de forma simplificada, como a fusão estrutural de transições de mesmo nome pertencentes ao alfabeto H, já que as transições sincronizadas através do mesmo nome são habilitadas através da união das suas condições em todos os sistemas e seus efeitos se estendem à união das suas posições. Desta forma, o disparo das transições sincronizadas só pode ocorrer em um determinado instante se elas estiverem habilitadas por todos os sistemas nesse instante.

A troca de mensagens é realizada através das marcas das boxes de comunicação de entrada e saída dos integrons e dos supervisores. Ao contrário dos gates, cujas marcas são determinadas pelo mundo exterior, as marcas dos lugares de comunicação são retiradas ou colocadas conforme a regra convencional de disparo das transições.

Considerando a partição do conjunto de eventos da TCS, em eventos não-controláveis e eventos controláveis, os gates representam os primeiros e os boxes de comunicação representam os últimos. Por extensão, as atividades associadas podem ser divididas em não-controláveis e controláveis, respectivamente. Com esses elementos na rede GHENeSys é possível representar um integron de forma isolada, destacando-se as influências externas sobre as atividades ou sequências de eventos. Assim, os boxes externos e as atividades associadas a eles modelam a interação do integron com o ambiente e com outros integrons.

De forma análoga aos autômatos integrados, a interação entre redes integradas será modelada pela atuação de um supervisor sobre as redes segundo um esquema de

U é o conjunto de funções $u: CE \rightarrow \{0,1\}$, CE : conjunto dos boxes de comunicação de entrada da rede composta $\parallel I_p$, obtido pela união dos conjuntos dos boxes de comunicação das redes componentes.

Analogamente aos autômatos integrados, é possível propor que o integron de nível superior formado pela atuação do supervisor r , como elemento de integração, sobre um conjunto de redes GHeNeSys integradas também é uma rede GHeNeSys integrada.

Proposição 5.1 A atuação do supervisor r sobre as redes GHeNeSys integradas concorrentes $I_j, j = 1 \dots n$, gera um integron de nível superior G , também modelado por uma rede GHeNeSys integrada.

Prova:

Seja M_{jk} o vetor de marcação da rede componente I_j no instante lógico k . A evolução dinâmica da rede pode ser dada pela equação de transição de estados:

de atividades ou transições externas, que são habilitadas pelo ambiente externo ao integron G (subconjunto $X_{j,ext}$) ou pela comunicação com outras redes componentes de G , sendo, portanto, internas a G (subconjunto $X_{j,int}$).

Em um determinado instante lógico k , a marcação M_k da rede composta que forma o integron G pode ser representada pela n -upla $(M_{1,k}, M_{2,k}, \dots, M_{n,k})$, $j =$

1...n. O conjunto $S = \Pi S_j$ representa todas as possíveis marcações de G. O supervisor r não afeta o conjunto de estados fisicamente possíveis das redes componentes, restringindo apenas determinadas seqüências de transições que poderiam levar o sistema a um estado crítico.

Assim como no modelo dos autômatos integrados, o conjunto de transições pode ser dividido entre transições externas V_{ext} e transições internas V_{int} ao integron G, como se segue:

$$V_{ext} = X_{ext} = \cup X_{j,ext}, j = 1...n$$

$$V_{int} = T_{int} \cup X_{int}, \text{ onde } T_{int} = \cup T_{j,int} \text{ e } X_{int} = \cup X_{j,int}, j = 1...n$$

Cada transição interna em T_{int} é habilitada exclusivamente pela marcação da rede I_j a qual pertence e, portanto, também pela marcação M_k da rede composta. Porém, algumas transições internas a G, pertencentes ao conjunto X_{int} , são afetadas pelo supervisor r, já que estão associadas à integração entre as redes. Essas transições são habilitadas pela marcação da rede I_j a qual pertencem e pela

marcação dos boxes de comunicação associados. No entanto, a marcação desses boxes é dada por $r(M_k)$, conforme a definição anterior do supervisor r , ou seja, também depende da marcação da rede composta. Logo, cada transição interna em X_{int} também é habilitada somente por M_k e, assim, todas as atividades ou transições internas de G , pertencentes a V_{int} , dependem apenas de M_k , como ocorre nas redes GHENeSys integradas.

Uma vez demonstrado que as transições externas e internas do integron de nível superior G atendem aos requisitos das redes GHENeSys integradas, resta exibir a função f de transição de estados de G a partir das funções f_j de transição de estados dos componentes I_j , o que pode ser feito como se segue:

$$M_k = (M_{1k}, \dots, M_{jk}, \dots, M_{nk}).$$

$$Y^{k+1} = g(M_k, t_k) = g_j^*(M_k, t_k) = g_j(M_{jk}, t_k), \text{ se } t_k \in T_j, \text{ onde}$$

Assim como no modelo dos autômatos integrados, uma função de saída pode ser associada ao modelo por redes integradas, definida a seguir:

As funções $g^*_j: S \times V \rightarrow Y_j$ são construídas de tal forma que $g^*_j(M_k, t_k) = g_j(M_{j,k}, t_k)$, se $t_k \in T_j$, onde $M_k = (M_{1,k}, \dots, M_{j,k}, \dots, M_{n,k})$, Y_j é o conjunto de boxes externos de saída da rede I_j e $g_j: S_j \times V \rightarrow Y_j$ é a função de saída de I_j .

As equações de transição de estados e de saída obtidas para o integron de nível superior G modelam o seu comportamento, representando a emergência de novas propriedades locais. O integron G , modelado por uma rede GHENeSys integrada conforme demonstrado na proposição 5.1, pode ser composto com outros integrons segundo um esquema recursivo formal análogo ao dos autômatos integrados, baseado na observação parcial de atividades ou transições e na estabilidade associada a marcações ou estados críticos.

As restrições de controle podem ser representadas por um conjunto de inequações sobre os elementos do vetor de marcação (YAMALIDOU et al., 1996). Estas restrições de acesso a estados críticos estão associadas às propriedades locais do integron e determinam invariantes de lugar sobre a rede controlada (sistema em malha-fechada).

Desta forma, é possível representar os componentes, o elemento de interação e o integron conforme a estrutura da rede, com base na álgebra de matrizes, o que simplifica o modelo e a simulação do seu comportamento.

O supervisor atua em composição paralela síncrona com os integrons (fusão de atividades correspondentes) e a sua marcação inicial e estrutura obedecem às equações abaixo em forma matricial (MOODY; ANTSAKLIS, 1996; MOODY et al., 1995):

$$M_{c0} = b' - L' \cdot M_{p0}$$

$$D_c = -L' \cdot D_p,$$

onde M_{p0} e D_p são a marcação inicial e a matriz de incidência da rede composta

GHENeSys integradas, baseadas no esquema condição/evento, a marcação de um box indica que o sistema pode estar em qualquer um dos estados agregados correspondentes. Portanto, uma restrição de controle que evite a entrada em um estado agregador também impede a entrada em qualquer dos sub-estados associados a ele. Esta característica pode

ser utilizada para sintetizar supervisores a partir de modelos simplificados dos componentes da planta.

A utilização de invariantes de lugar representativos das especificações de projeto (restrições associadas a estados críticos) de um sistema de controle integrado possibilita que mecanismos de integração (controladores) por realimentação sejam obtidos eficientemente através de multiplicações de matrizes. Conforme ressaltado em (YAMALIDOU et al., 1996), apesar do sistema controlado obtido não ser em geral ótimo com relação ao número de lugares da rede controladora, ele pode representar um ponto de partida adequado para otimizações subsequentes.

Considerando, portanto, que o supervisor que realiza a integração das redes componentes, segundo uma determinada especificação, também pode ser representado por uma rede de Petri pelo método dos invariantes de lugar, cada integração na árvore pode ser obtido recursivamente de forma análoga ao modelo por autômatos integrados conforme se segue:

$$I_{n_j} = (\| I_{n-1, k} \| \| c_{n_j} \|)_{H, I = 1, \dots, p,}$$

onde os integrons de nível n e $n-1$ e o supervisor c_{n_j} são representados por redes integradas e o conjunto H representa as atividades relevantes para a integração.

5.6 Princípios para uma metodologia de projeto de sistemas de controle integrados

O modelo baseado em integrons sugere alguns princípios para a elaboração de uma metodologia de projeto para sistemas de controle integrados, envolvendo em particular as máquinas autônomas. Esta classe de sistemas complexos inclui robôs e veículos para exploração de ambientes não estruturados, exigindo diversas propriedades para a execução de seu objetivo ou missão. Algumas etapas de projeto são apresentadas

- Cada propriedade é realizada por um integron, obtido a partir dos integrons primitivos segundo o esquema "bottom-up" recursivo mencionado em 5.5.
- Cada propriedade em um determinado nível da arquitetura deve ser obtida

através da integração de integrons-redes do nível inferior, estando associada a estados críticos representados por invariantes de lugar;

- Caso não seja possível a geração de uma propriedade desejada, em virtude, por exemplo, do supervisor sintetizado necessitar desabilitar uma atividade não-controlável, o esquema de decomposição de propriedades deve ser refeito, buscando encontrar uma nova configuração que atenda ao objetivo do sistema; caso não exista outra configuração de propriedades viável, os recursos de percepção e ação do sistema podem ser ampliados, possibilitando uma variedade maior de opções de integração. Caso esta alternativa também não seja possível por critério de custo, por exemplo, a missão ou objetivo

Capítulo 6

Estudo de Caso: Veículos Autônomos

A pesquisa e o desenvolvimento de veículos autônomos têm sido objeto de esforços crescentes, motivados principalmente pela curiosidade intelectual associada à busca de novas e complexas funções para robôs e máquinas em geral. Aplicações de veículos operados remotamente, veículos autônomos terrestres, submarinos, aéreos e espaciais em exploração de ambientes complexos e perigosos para o ser humano, em manufatura automatizada (veículos autônomos guiados - AGVs), em missões de caráter militar ou estratégico e na indústria de petróleo (exploração de águas profundas) têm contribuído adicionalmente para a pesquisa na área (SOOD et al., 1990).

de grandes perturbações e não-linearidades associadas às características hidrodinâmicas do veículo.

Desta forma, as técnicas de controle contínuo, mesmo baseadas em modelos não-lineares de equações diferenciais, não são suficientes para manter o veículo dentro dos limites desejados de velocidade, posição e profundidade/altitude (em relação ao

fundo do mar), pois as perturbações ocasionadas por obstáculos, correntes marítimas e problemas de energia são imprevisíveis e podem ser de grande magnitude. Esta característica do ambiente oceânico inviabiliza a imposição de limites artificiais às perturbações no modelo matemático do sistema, exigindo um esquema de resposta a eventos externos. Por outro lado, a existência de seis graus de liberdade no AUV, movimentando-se em três dimensões, gera diversas possibilidades de ação como resposta a eventos, ao contrário do veículo terrestre, o que exige um mecanismo de tomada de decisões e controle adequado. Um projeto inadequado de um AUV pode tornar a vantagem de multiplicidade de ações para a autonomia em enorme desvantagem, ocasionando a perda ou comprometendo a segurança do veículo.

baseada na arquitetura de redes integradas, associadas a SEDs,

1996), onde foi utilizado um modelo multi-agentes para a cooperação entre processos de manobra do veículo ("pilot") e mapeamento de obstáculos ("mapper"), visando manter o veículo seguindo o relevo do fundo do mar a uma determinada altitude mínima. O sistema em questão se restringiu à guiagem no plano vertical.

O modelo multi-agentes utilizado no referido trabalho foi substituído pela arquitetura baseada em integrons, visando obter todas as vantagens de um método formal, capaz de orientar o projeto do sistema de controle do AUV de modo seguro. O

implicações para o controle do AUV.

de forma apenas superior, mesmo para sistemas

de forma apenas superior, mesmo para sistemas

Para o estudo de caso apresentado a seguir, foi acrescentada ao AUV uma

Este exemplo ilustra um aspecto importante dos veículos autônomos, relacionado à integração entre mecanismos de percepção e ação. A manutenção da altitude no "setpoint" desejado poderia ser feita segundo um esquema de controle contínuo em cascata, variando o ângulo de "pitch", com atuação em superfícies de controle do veículo. No entanto, a possibilidade de colisão com o relevo do fundo do mar durante a execução da exploração exige a alteração do "setpoint" provisoriamente, com base em dados do sonar e do mapa de obstáculos gerado dinamicamente. Assim, é necessário um nível superior de controle que reaja ao ambiente, comutando a operação de forma adequada e segura.

requisito de concorrência e/ou paralelismo do controle do AUV está relacionado aos diferentes tempos de resposta dos sensores e atuadores, que impossibilitam um tratamento sequencial dos processos envolvidos. Este requisito é atendido pela arquitetura de multi-agentes, no entanto, a arquitetura baseada em integrons possui mecanismos formais de integração entre processos concorrentes, tratando a cooperação entre processos como um problema de controle supervisorio de SEDs, através do método dos invariantes de lugar.

A metodologia baseada em integrons, exposta no item 5.6 do presente trabalho, inicia com a definição da missão ou objetivo do sistema. No caso em questão, a missão do AUV é seguir o relevo do fundo do mar, mantendo uma altitude mínima em relação

à superfície das suas baterias de forma automática. Esta recarga é

- Guiagem segundo uma altitude e uma posição determinadas, evitando colisões.
- Detecção da disponibilidade de energia.

A propriedade de guiagem evitando colisões pode ser decomposta em duas propriedades:

- Guiagem segundo uma altitude e uma posição determinadas, com detecção de colisões.

- Mapeamento dinâmico de obstáculos. Esta propriedade permite a navegação com maior segurança, através da construção de um mapa do ambiente enquanto o veículo se desloca, permitindo considerar obstáculos previamente encontrados para a tomada de decisões e a procura de espaços vazios. O método de construção do mapa não é relevante para este exemplo, cujo

propriedade permite que o veículo se desloque em espaços vazios.

atingir um ponto cujas coordenadas são fornecidas). economia de energia) e em um modo de ponto de referência (manobra para terreno), no modo de segurança (menor profundidade, resultando em

- Detecção de obstáculos, que pode ser baseada em um sonar.

Portanto, as propriedades primitivas obtidas são:

- Detecção da disponibilidade de energia.
- Mapeamento dinâmico de obstáculos.
- Detecção de obstáculos.
- Guiagem segundo uma altitude e uma posição determinadas, com três modos de operação: normal, segurança/economia de energia e ponto de referência.

- Detecção de obstáculos

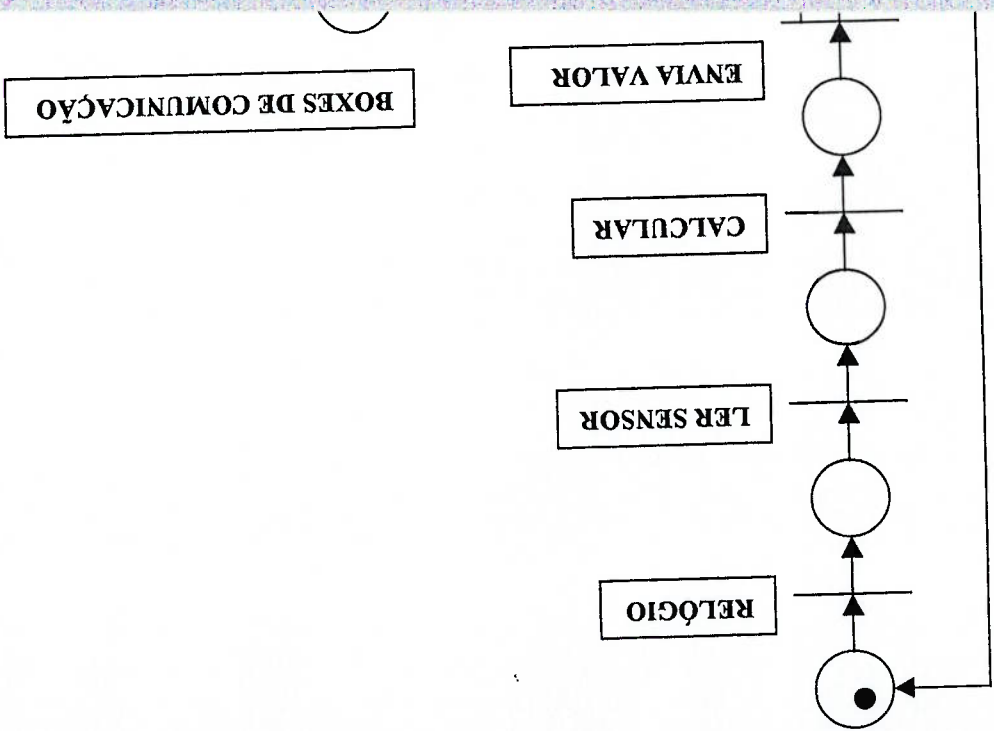
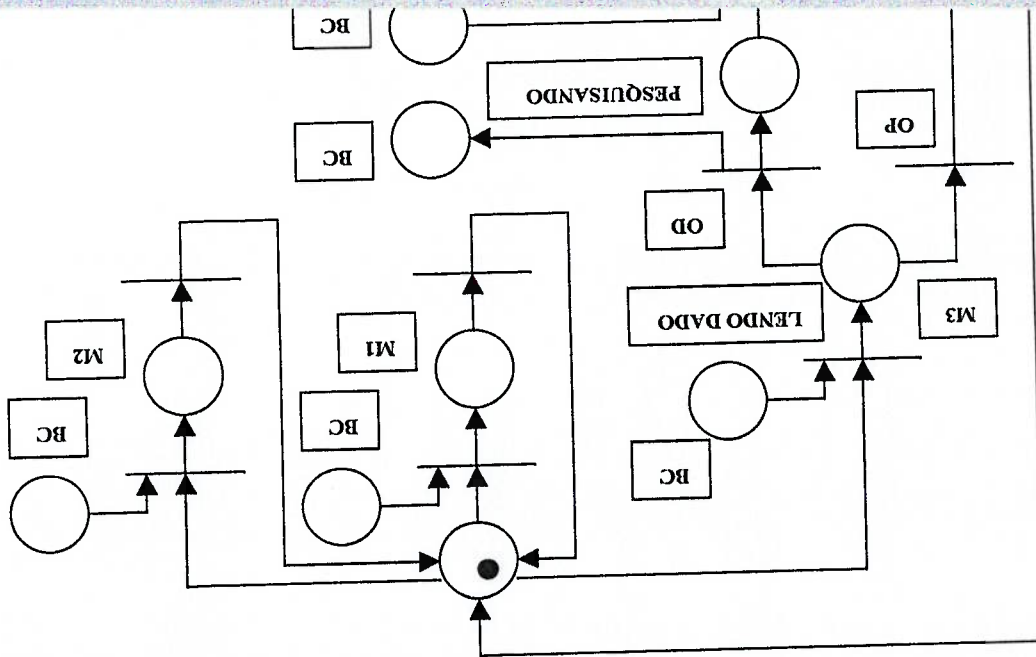
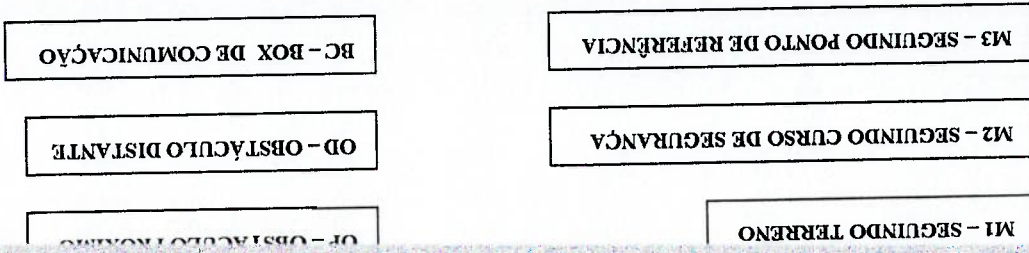


Fig. 6.2 Guiagem com controle de altitude e posição em três modos (piloto)



Guiagem com controle de altitude e posição em três modos (piloto)

- Mapeamento dinâmico de obstáculos

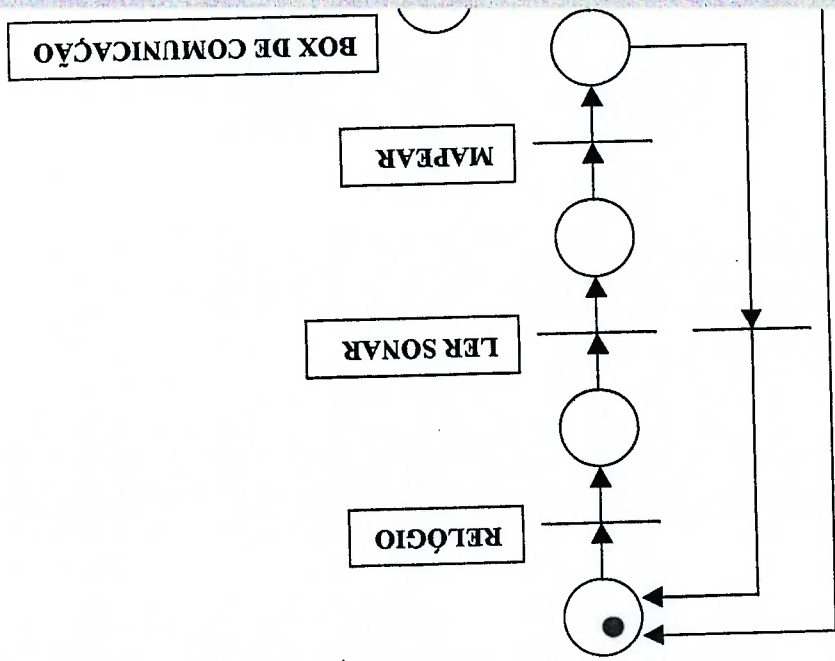


Fig. 6.3 Mapeamento dinâmico de obstáculos

- Detecção de disponibilidade de energia

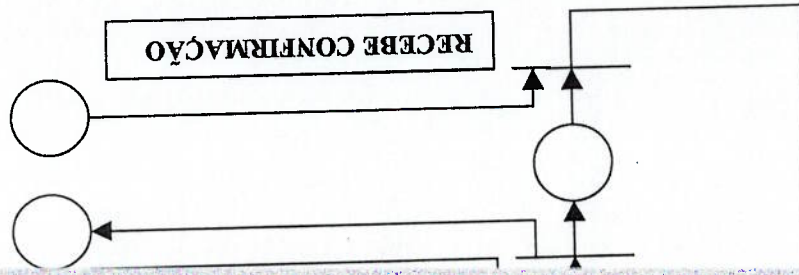
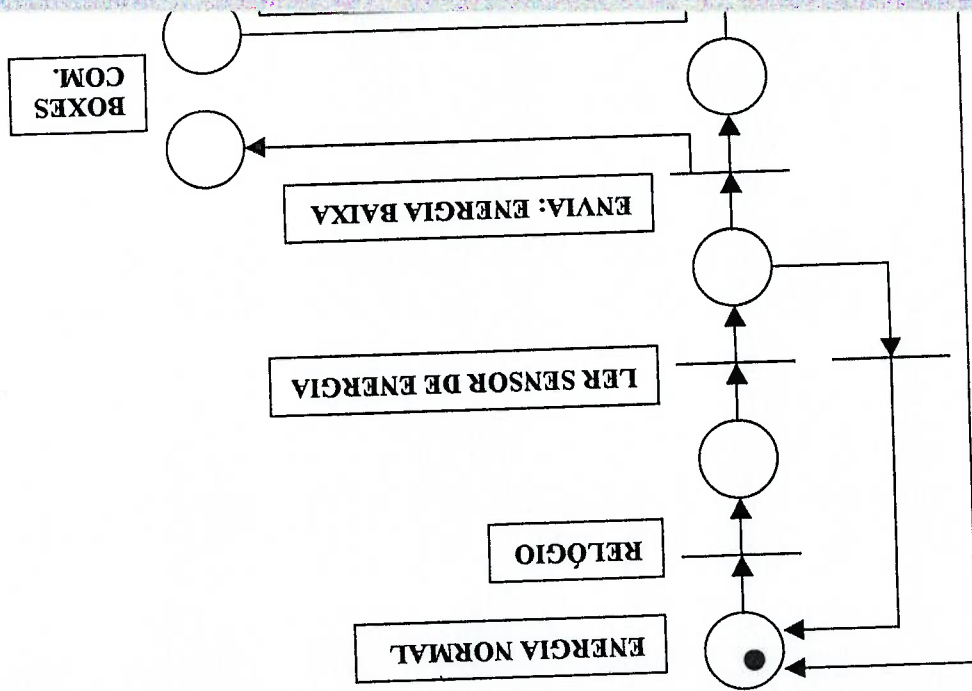


Fig. 6.4 Detecção de disponibilidade de energia

Após a modelagem dos integrons primitivos por redes integradas, a próxima etapa é a síntese dos supervisores de integração conforme as propriedades desejadas, segundo o esquema "bottom-up" estabelecido.

Portanto, a primeira propriedade a ser obtida é a de guiagem com detecção de colisões, a partir da integração dos integrons primitivos de guiagem (piloto) e de detecção de obstáculos. Para a integração das redes deve ser feita inicialmente uma transformação estrutural, que consiste na abstração das atividades que não participam da integração (não-observáveis pelo supervisor) através de boxes de nível superior representando sub-redes. A figura 6.5 ilustra o resultado dessa transformação juntamente com o supervisor sintetizado pelo método dos invariantes de lugar:

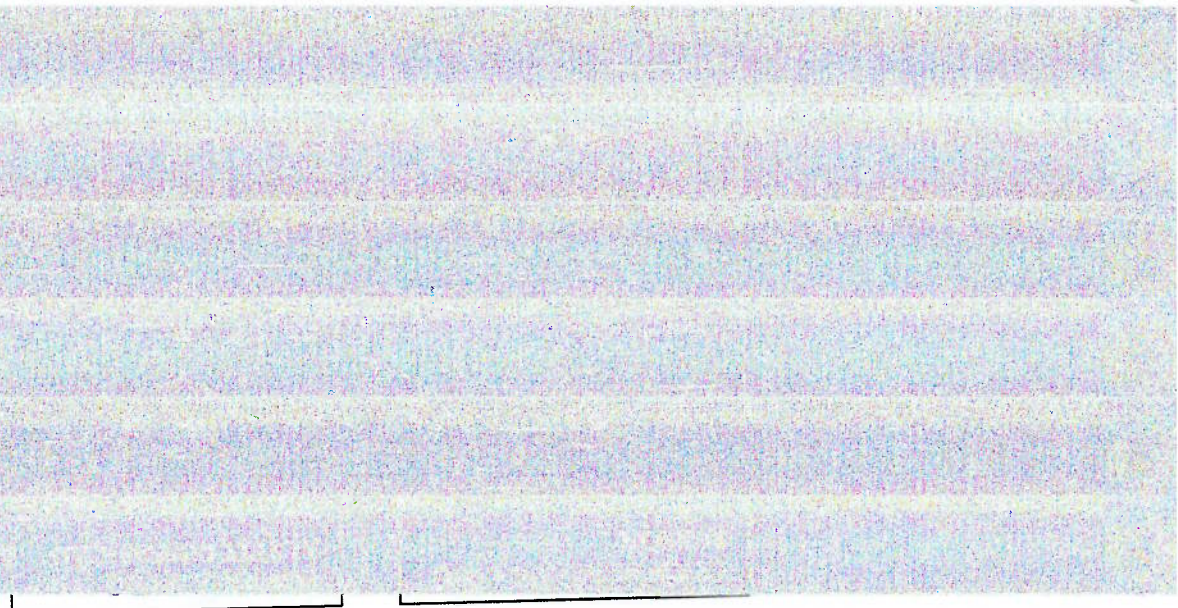


Fig. 6.5 Integron que representa a propriedade de guiagem com detecção de obstáculos

O integron detetor de obstáculos envia o valor mais recente obtido do sensor, gerando um ponto de referência que é utilizado pelo piloto para executar a manobra adequada.

O supervisor representado pelo box tracejado na figura 6.5 é obtido pelo método dos invariantes, conforme a especificação $x^1 + x^4 \leq 1$, ou seja os boxes p1 e p4 não podem estar marcados simultaneamente, o que representaria um estado crítico associado a uma manobra sem informação correspondente do sensor de obstáculo.

Esta especificação pode ser representada sob a forma de matriz como se segue:

$$L = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \quad b = 1$$

A matriz de incidência da rede obtida pela composição paralela sem sincronização das redes *piloto* e *sensor* é:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$$

$$M_{co} = b - L M_{p0} = 1 - \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Logo, $M_{co} = 0$.

A partir da matriz de incidência D_c e da marcação inicial M_{c0} , a rede que integra o piloto e o sensor pode ser obtida, sendo representada na figura 6.5 pelo box e pelos arcos tracejados.

A próxima etapa consiste na integração do integron obtido com o integron de mapeamento dos obstáculos. O procedimento é recursivo já que o integron gerado anteriormente também é uma rede integrada, como foi provado no capítulo 5. Procedendo analogamente, a rede integrada obtida é ilustrada na figura 6.6 a seguir:

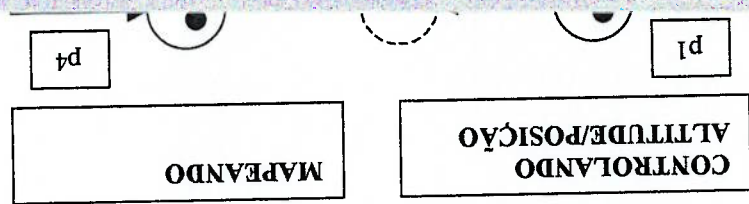


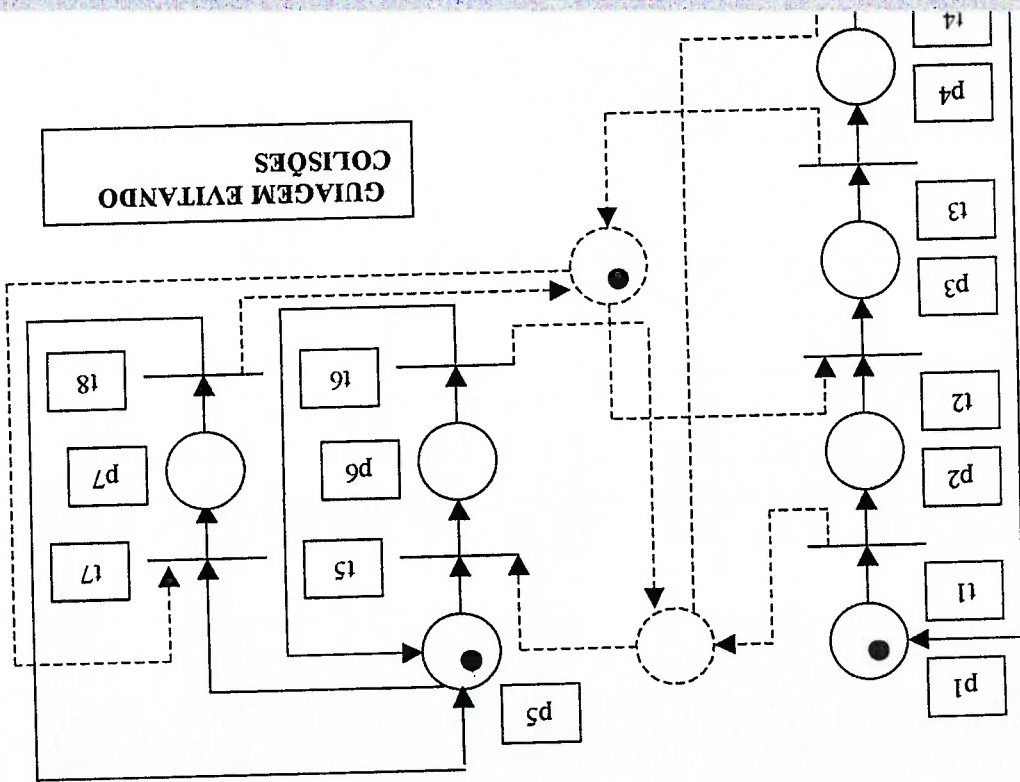
Fig. 6.6 Integron que executa a guiação evitando colisões

A rede de integração representada pelos dois boxes e pelas linhas tracejadas da figura 6.6 é obtida a partir das especificações $x_1 + x_5 \leq 1$ e $x_3 + x_5 \leq 1$, que indicam, respectivamente, que os boxes p1/p5 e p3/p5 não podem ser marcados simultaneamente.

Estes estados críticos estão associados ao fato do integron de mapeamento responder somente quando solicitado pelo piloto, e de que este só pode executar a manobra para evitar a colisão após receber a resposta dos espaços vazios extraída do mapa.

Uma vez obtido o integron de guiagem evitando colisões através do mapeamento dinâmico, resta integrá-lo ao componente que monitora a energia para obter o sistema de controle integrado que executa a missão desejada. A figura 6.7 a

normal e energia baixa. Os estados p6 e p7 representam, respectivamente, os modos de guilagem *segurança com economia de energia (menor profundidade)* e *segundo o terreno*.



A rede de integração (supervisor) da figura 6.7 é obtida através das

especificações $x_1 + x_6 \leq 1$ e $x_3 + x_7 \leq 1$. O estado crítico p1/p6 significa que se a

energia está normal, o veículo não deve navegar em modo de segurança (menor

profundidade), pois isso compromete a sua missão de seguir o relevo do fundo do mar.

O estado crítico p3/p7 significa que se o nível de energia estiver baixo, o veículo não

deve seguir o terreno, pois isso compromete a sua segurança e aumenta o consumo de

energia.

Portanto, o estudo de caso apresentado do AUV ilustra a metodologia de

projeto de sistemas de controle integrados baseada em integrons. Foram sintetizados

quatro integrons primitivos e três supervisores que realizam as propriedades necessárias

Capítulo 7

Conclusões

O estudo de caso realizado do AUV mostrou a eficiência da arquitetura baseada em integrons para o modelo e projeto de sistemas de controle complexos. No entanto, a necessidade do projetista determinar os estados críticos com base nas propriedades associadas à missão representa um sério limite para a construção de veículos autônomos que interagem com ambientes não estruturados.

Desta forma, a questão mais importante para pesquisa futura e continuidade

deste trabalho é definir um mecanismo pelo qual os estados críticos sejam identificados

em condições de operação do veículo, possivelmente de forma evolutiva a partir de

Referências Bibliográficas

- ANDERSON, T.L. e DONATH, M. Animal Behavior as a Paradigm for Developing Robot Autonomy, Robotics and Autonomous Systems, No. 6, pp. 145-168, 1990.
- ANDRÉ, C. Partial Observation On a Petri Net, Dynamical Systems and Cellular Automata, Academic Press, London, 1985.
- ARKIN, R.C. The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 6, 1990.
- BAAS, N.A. Emergence, Hierarchies, and Hyperstructures, Artificial Life III, C.G. Langton, Ed., Proceedings Volume in the Santa Fe Institute Studies in the Sciences of Complexity, Vol. XVII, Addison-Wesley, 1994.
- BALEMI, S. et al. Supervisory Control of a Rapid Thermal Multiprocessor, IEEE Transactions on Automatic Control, Vol.38, No.7, July 1993, pp.1040-1059.
- de BARROS, E.A. et al. Application of a Cooperative Control System to the Collision Avoidance Guidance of AUVs, Journal of The Society of Naval Architects
- DAVID, R. e ALTA, H. Petri Nets for Modeling of Dynamic Systems - A Survey, Automatica, Vol. 30, No. 2, pp. 175-202, 1994.
- DENG, Y., CHANG, S.K., FIGUEIREDO, J.C.A. e PERKUSICH, A. Integrating Software Engineering Methods and Petri Nets for the Specification and Prototyping of Complex Information Systems, Lecture Notes in Computer Science, 1993.
- ELFES, A. Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception, Proc. of the sixth Conf. on Uncertainty in AI, 1990.

- HAKEN, H. *Synergetics*, Springer, Berlin, Heidelberg, New York, 1978.
- HOARE, C.A.R. *Communicating Sequential Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1985.
- HOLCOMBE, W.M.L. *Algebraic Automata Theory - (Cambridge Studies in Advanced Mathematics; 1)*, Cambridge University Press, 1982.
- HOLLAND, J. *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: The University of Michigan Press, 1975.
- HOLLAND, J. *Emergence: From Chaos To Order*, Addison-Wesley, 1998.
- HOPFIELD, J.J. *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Natl. Acad. Sci. USA 79, pp. 2554-2558.
- JACOB, F. *La Logique du Vivant : Une Histoire de l'Herédité*, Editions Gallimard, Paris, 1970.
- KELLEY, D. *Automata and Formal Languages : An Introduction*, Englewood-Cliffs, Prentice-Hall, 1995.
- PROGRAM WITH Decentralized Control and Coordination of Discrete-
- MIYAGI, P.E. *A Study on Mark Flow Graph Based Programming Method for Robots*, Master Thesis, Tokyo Institute of Technology, 1985.
- MIYAGI, P.E. *Controle Programável : Fundamentos do Controle de Sistemas a Eventos Discretos*, Editora Edgard Blücher Ltda., 1996.
- MOODY, J.O. et al. *Feedback Petri Net Control Design In the Presence of Uncontrollable Transitions*, Proceedings of the 34th Conference on Decision & Control, 1995.

MOODY, J.O. e ANTSAKLIS, J.A. Supervisory Control of Petri Nets with Uncontrollable/Unobservable Transitions, Proceedings of the 35th Conference on Decision & Control, 1996.

MURATA, T. Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE, Vol. 77, pp. 541-580, 1989.

NAGANO, S. Diffusion-Assisted Aggregation and Synchronization in *Dictyostelium discoideum*, Physical Review Letters, Vol. 80, No. 21, 1998.

NEDERBRAGT, H. Hierarchical Organization of Biological Systems and the Structure of Adaptation in Evolution and Tumorigenesis, J. Theor. Biol., No. 184, pp. 149-156, 1997.

NICOLIS, G. e PRIGOGINE, I. Self-Organization in Non Equilibrium Systems, Wiley, New York, London, Sydney, 1977.

PARISI, G. Facing Complexity, Physica Scripta, No. 35, pp. 123-124, 1987.

PETRI, C.A. Kommunikation mit Automaten, Schriften des Institutes für Instrumentelle Mathematik, Bonn, 1962.

REISIG, W. Petri Nets, FACTS Monographs on Theoretical Computer Science, W. Brauer, G. Rozenberg, A. Salomaa, Eds., Springer-Verlag, New York, 1983.

SAVILL, N.J. e HOGGEWEG, P. Modelling Morphogenesis: From Single Cells to Crawling Slugs, J. Theor. Biol., No. 184, pp. 229-235, 1997.

SHANNON, C.E. e WEAVER, W.W. The Mathematical Theory of Communication, University of Illinois Press, Urbana, IL, 1949.

SILVA, J.R. e MIYAGI, P.E. PFS/MFG: A High Level Net to the Modeling of Discrete Manufacturing Systems, In: Camarinha-Matos, L.M. and Afsarmanesh, H. (Eds.) Balanced Automation Systems - Architectures and Design Methods, IFIP/Chapman & Hall, London, 1995.

SILVA, J.R. e MIYAGI, P.E. A Formal Approach to PFS/MFG: A Petri Net Representation of Discrete Manufacturing Systems, Studies in Informatics and Control, IC Publications, Romania, 1996.

SILVA, J.R., RAMOS, R.T.C.B. e MIYAGI, P.E. Supervisory Control of Integrated Building Systems: A Balanced Approach, In: Camarinha-Matos, L.M. and Afsarmanesh, H. (Eds.) Balanced Automation Systems - Architectures and Design Methods, IFIP/Chapman & Hall, London, 1996.

SILVA, J.R. Interactive Design of Integrated Systems, Proceedings of BASYS'98, Prague, Czech Republic, 1998.

SIMON, H.A. The Architecture of Complexity, Proc. Am. Philos. Soc., Vol. 106, No. 467, 1962.

SOOD, A.K. et al. Computational Perspective on Perception, Planning, Action and Systems Integration, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20,

Lecture Notes in Computer Science, Vol. 029, Springer, 1998.

VON NEUMANN, J. Theory of Self-Reproducing Automata, A.W. Burks, Ed., University of Illinois Press, Champaign, IL, 1966.

WIENER, N. Cybernetics, or Control and Communication in the Animal and the Machine, MIT Press, Cambridge, MA, 1948.

WOLFRAM, S. Computation Theory of Cellular Automata, Commun. Math. Phys., Vol. 96, No. 15, 1984.

- WOLFRAM, S. *Cellular Automata and Complexity : Collected Papers*, Addison-Wesley, 1994.
- WOLLE, F.A.C. e ROCHA, G.N.T.V. *Sistema Supervisório Descentralizado Baseado em Integrons*, Trabalho de Curso T.F.-98, Orientador J.R. Silva, São Paulo, EPMc, 1998.
- YAMALIDOU, K. et al. *Feedback Control of Petri Nets Based on Place Invariants*, *Automatica*, Vol.32, No.1, pp. 15-28, 1996.
- ZEIGLER, B.P. *DEVS Representation of Dynamical Systems: Event-Based Intelligent Control*, *Proceedings of the IEEE*, Vol.77, No.1, January 1989.
- ZHONG, H. e WONHAM, W.M. *On the Consistency of Hierarchical Supervision in Discrete-Event Systems*, *IEEE Transactions on Automatic Control*, Vol. 35, No. 10, pp. 1125-1134, 1990.
- ZUREK, W.H., ed. *Complexity, Entropy, and The Physics of Information*, Addison-Wesley, Redwood City, CA, 1990.