

Uma Arquitetura de Controle Distribuída para um Sistema de Visão Computacional Propositada.

Reinaldo A. C. Bianchi¹

Anna H. R. C. Rillo²

¹Divisão de Automação e Inteligência Artificial - Laboratório de Sistemas Integráveis

²Departamento de Engenharia de Computação e Sistemas Digitais

Escola Politécnica - Universidade de São Paulo - Brasil.

Av. Prof. Luciano Gualberto, travessa 3, 158. 05508-900 São Paulo, SP, Brasil.

E-mail: rbianchi@lsi.usp.br, arillo@pcs.usp.br

RESUMO

Este trabalho é o resumo de uma dissertação de mestrado em Engenharia Elétrica, em andamento na Escola Politécnica da USP, e apresenta uma arquitetura de controle distribuída para um sistema de visão computacional propositada, realizando tarefas simples de montagens visualmente guiadas. Esta arquitetura é baseada em modelos de IA Distribuída, onde os propósitos do sistema são traduzidos em um conjunto de comportamentos. Uma abordagem multi-agentes é usada para modelar propósitos e comportamentos, assim como suas relações. Os propósitos de um sistema são representados por uma sociedade dinâmica de agentes autônomos, cada qual responsável por um comportamento específico. Finalmente, é descrita uma implementação da arquitetura proposta.

1. Introdução

Motivada pela busca de arquiteturas mais flexíveis e adequadas às novas abordagens para sistemas de visão computacional que interagem com o ambiente através de percepção e ação, esta dissertação apresenta uma Arquitetura de Controle Distribuída para sistemas de Visão Computacional Propositada, modelada segundo uma abordagem de sistemas Multi-Agentes.

A abordagem Multi-Agentes, conhecida como MAS (*Multi-Agents Systems*), é uma das principais áreas de Inteligência Artificial Distribuída (DAI), e se preocupa com a coordenação dos comportamentos de diversos agentes inteligentes autônomos para atingir um ou mais objetivos.

Por outro lado, a abordagem propositada para visão computacional [1], [2], [3] considera a visão dentro de um contexto de tarefas que um agente deve realizar, retirando dos propósitos do agente as restrições para solucionar o problema de visão. Desta forma, esta abordagem leva a soluções de tarefas específicas e não de uso geral, sendo que uma visão genérica (*multi-purpose*) surge da organização de diversas soluções dedicadas a diferentes tarefas visuais. Portanto, um dos maiores problemas torna-se em como organizar estas soluções e definir tarefas primitivas, focando a pesquisa em arquiteturas para a integração de sistemas visuais.

Dentro deste contexto, uma arquitetura multi-agentes é proposta para modelar o sistema de visão propositada, onde os propósitos do sistema são traduzidos em um

conjunto de comportamentos. Cada propósito é modelado em uma sociedade de agentes autônomos.

Vários trabalhos influenciam esta dissertação, dentre os quais cita-se a *Subsumption Architecture* de Brooks [4], onde um sistema é composto por camadas com tarefas específicas e cada camada interage diretamente com o mundo através de percepção e ação; a Arquitetura de Controle Distribuída de Elfes [5], onde um sistema é dividido em níveis de processamento e processos independentes comunicam entre si através de um *blackboard*; os trabalhos *ASIC Multi-Agents Control Architecture* e o sistema MAVI de Boissier e Demazeau [6],[7], que integram diferentes módulos visuais usando a teoria de Multi-Agentes.

2. A Arquitetura Proposta

A arquitetura multi-agentes proposta consiste numa sociedade de Agentes Autônomos (AAs), onde cada AA é responsável por um comportamento diverso. Os AAs comunicam entre si através de uma rede descentralizada e totalmente conectada e são organizados segundo uma estrutura de autoridade e regras de comportamento.

2.1. A Sociedade de Agentes Autônomos

A definição da sociedade de agentes é realizada dinamicamente, permitindo que AAs sejam inseridos ou retirados da sociedade durante a operação do sistema, segundo o controle de um ou vários agentes membros da sociedade. Deste modo, é possível a criação de um sistema auto-gerenciado e multi-propósito, bastando a definição dos agentes necessários para a realização de cada propósito e a criação de um agente que controle a troca de propósitos, com a função de gerenciar os AAs necessários a cada propósito.

Nesta arquitetura um recurso é definido como sendo parte do sistema com uso compartilhado pelos agentes e que pode ser controlada por somente um agente em um determinado momento. Por exemplo, um manipulador robótico em uma célula de montagem é um recurso; por outro lado, uma câmera fixa não é um recurso, pois todos os agentes podem ter a imagem capturada por ela ao mesmo tempo. Caso esta câmera faça parte de um sistema de visão ativa, ela torna-se um recurso, uma vez que os agentes competem pelo controle do processo de aquisição de imagens.

O que torna possível a decisão sobre como os recursos do sistema devem ser alocados é a *estrutura de autoridade* da sociedade. Ela define a relação de prioridades entre AAs para o controle de um recurso em um determinado momento.

Esta estrutura de autoridade possui ainda uma profunda relação com a precedência e a dependência entre as ações dos comportamentos dos agentes presentes na sociedade. Sua definição deve ser feita a partir de um estudo da linearização de um plano de atividades, que tem nos agentes autônomos seus operadores e cujo objetivo é o de realizar os propósitos do sistema. Vale notar que esta estrutura relaciona a arquitetura proposta à *Subsumption Architecture* de Brooks[4], pois permite que os agentes autônomos inibam e suprimam uns aos outros, retirando os recursos que necessitam. Os Agentes Autônomos ainda se relacionam com as camadas da arquitetura de Brooks, pois percebem e agem diretamente no mundo e podem suprimir as ações uns dos outros.

A maneira pela qual a decisão sobre a alocação dos recursos é tomada depende das

regras de comportamento definidas para a sociedade em uma implementação específica e pode ser, por exemplo, o resultado de uma competição entre agentes.

2.2. Definição de um Agente Autônomo

Baseado no trabalho de Boissier e Demazeau [7], define-se um agente autônomo como:

$\langle \text{Agent} \rangle ::= \langle \text{rules} \rangle \ \langle \text{agents} \rangle \ \langle \text{state} \rangle \ \langle \text{communication language} \rangle \ \langle \text{decision capabilities} \rangle$

Onde:

$\langle \text{rules} \rangle$ são as regras de comportamento da sociedade em que o agente está inserido;

$\langle \text{agents} \rangle$ são os agentes na sociedade e a estrutura de autoridade entre eles;

$\langle \text{state} \rangle$ é a descrição do estado dos elementos pertinentes que façam parte do domínio, assim como da alocação e do estado dos recursos da sociedade;

$\langle \text{communication language} \rangle$ é a linguagem de comunicação usada pelos agentes;

$\langle \text{decision capabilities} \rangle$ são as capacidades que um AA possui que o tornam apto a decidir qual agente deve ter o controle de um recurso em um determinado momento, baseadas, entre outros, na estrutura de autoridade.

Esta definição é usada para modelar todos os AAs em um sistema, não importando o seu comportamento. Outra característica essencial relacionada aos agentes em um ambiente multi-agentes é a sua capacidade de comunicação. Isto torna a definição da linguagem de comunicação a ser usada por eles um tópico fundamental.

2.3. A Linguagem de Comunicação dos AAs

Modificando-se a linguagem de comunicação descrita em Boissier e Demazeau [7], definiu-se a seguinte linguagem de interação entre os agentes autônomos do sistema:

$\langle \text{interaction} \rangle ::= \langle \text{type} \rangle \ \langle \text{content} \rangle$

onde:

$\langle \text{type} \rangle$ é o tipo da comunicação;

$\langle \text{content} \rangle$ é o conteúdo da comunicação;

Os elementos da linguagem de comunicação, assim como sua implementação, dependem das regras de comportamento da sociedade de AAs de um sistema; porém existem alguns tipos básicos necessários, resumidos na Tabela 1.

3. Uma implementação utilizando a arquitetura proposta

Um sistema realizando tarefas de montagens simples e visualmente guiadas baseado na arquitetura aqui proposta está sendo implementado na Célula Flexível de Montagem [9] da Divisão de Automação e Inteligência Artificial do Laboratório de Sistemas Integráveis da Escola Politécnica da USP. Esta Célula de Montagem é composta por várias estações de trabalho, dois manipuladores robóticos de 5 graus de liberdade, câmeras de vídeo e placas digitalizadoras, com todos os computadores interligados por uma rede ethernet local.

<type>	<content>	descrição
addAgent	<nome do novo agente> <nova estrutura de autoridade>	adiciona um agente à sociedade
deleteAgent	<nome do agente>	retira um agente da sociedade
inform	{<recurso> <ocupado livre> <agente controlador>}	envia informações sobre a alocação dos recursos para um novo agente
acknowledge	<nome do agente>	confirma o recebimento de mensagem de adição ou remoção de agentes
request	<nome do agente> <recurso>	requisita o controle de um recurso do sistema a outro agente
transfer	<agente destino> <recurso> [estado]	transfere o controle de um recurso de um AA para outro
free	<recurso>	avisa aos agentes que o controle de um recurso foi liberado

Tabela 1 - A Linguagem de Comunicação.

Este sistema tem como propósito a realização de uma montagem simples. Para atingir este objetivo, foram definidos três Agentes Autônomos que se encontram totalmente conectados (figura 1) e utilizam uma das câmeras da célula de montagem. Atualmente, o único recurso que o sistema possui é um dos manipuladores robóticos da célula.

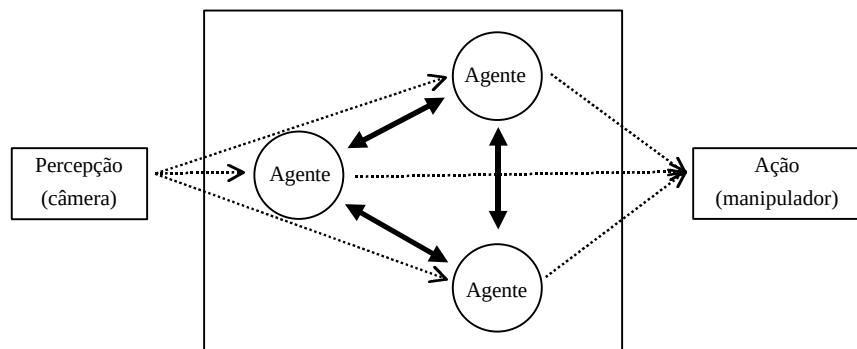


Figura 1: Esboço da sociedade implementada.

Os Agentes Autônomos foram definidos como:

1. *Agente Montador*: realiza uma atividade de montagem, pegando peças conhecidas da área de trabalho com o manipulador e colocando-as nas posições desejadas. O objetivo da montagem e a forma e tipo das peças podem mudar, por exemplo, de uma montagem de um objeto conhecido para a seleção de peças por forma ou cor. Para isso, o agente tem que ser capaz de planejar as atividades envolvidas na montagem. Este planejamento pode ser simples ou complexo, reativo ou temporal, dependendo da complexidade do sistema. No estado atual, o plano é previamente definido pois o agente não possui capacidades de planejamento automático de atividades.

2. *Agente Desobstrutor*: desobstrui a área de trabalho. A área de trabalho é um espaço, previamente definido como importante para a montagem. Assim, objetos não desejados que uma pessoa ou outro manipulador possam ter colocado nesta área devem ser retirados pelo agente.

3. Agente *Evitador de Colisões*: evita colisões do manipulador com objetos que se movam na área de trabalho, com o objetivo de preservar a integridade física do recurso.

A *estrutura de autoridade* foi definida para os agentes da seguinte maneira: o agente *Evitador de Colisões* é o agente com maior autoridade, o *Desobstrutor* é o segundo e o *Montador* é o agente com menor autoridade. Pode-se notar que a estrutura de autoridade tem por objetivo principal a preservação da integridade física do sistema e, secundária, a tarefa global a ser realizada. Quando um sistema cresce, a estrutura é modificada para incorporar outros agentes.

Ainda, a sociedade dos agentes possui três *regras de comportamento*, que fazem os mesmos competir pelo manipulador:

Regra # 1: somente um agente pode controlar o manipulador em um determinado momento.

Regra # 2: qualquer agente pode requisitar o controle do manipulador para um agente com menor autoridade que ele, em qualquer momento.

Regra # 3: um agente só pode requisitar o controle do manipulador para um agente de maior autoridade se este estiver cedendo o controle.

Finalmente, a descrição de *estado* que cada agente possui se refere a alocação e ao estado de trabalho do recurso, que descreve, por exemplo, se o manipulador está com uma peça na garra, sua posição atual, etc. As *capacidades de decisão* referem-se, no caso, somente a uma função que decide, com base no conhecimento da estrutura de autoridade, se transfere ou não o recurso para outro agente.

No exemplo abaixo é apresentado o funcionamento do sistema, descrevendo, em uma linguagem baseada em LISP, a sequência de mensagens que são trocadas quando algum objeto é colocado na área de trabalho durante uma montagem. Para evitar uma colisão, o agente *Evitador* requisita o controle do recurso ao agente *Montador*; o agente *Montador* transfere o controle ao agente *Evitador*; o agente *Evitador de Colisão* cede o controle do recurso ao final de perigo de colisão, controle este que é requisitado pelos agentes *Desobstrutor* e *Montador*; o agente *Evitador* transfere então o controle ao de maior autoridade, o *Desobstrutor*; ao terminar de desobstruir a área de trabalho, o *Desobstrutor* cede o controle do recurso; finalmente, o agente *Montador* requisita o controle, que é transferido do *Desobstrutor* para ele, que pode então retornar ao seu trabalho.

```
comentário: o controle está com Assembler.
(request (collisionAvoider)(manipulator))
(transfer (collisionAvoider) (manipulator) (piece in grip))
comentário: a colisão é evitada.
(free (manipulator))
(request (cleaner) (manipulator))
(request (assembler) (manipulator))
(transfer (cleaner) (manipulator) (piece in grip))
comentário: o objeto é removido.
(free (cleaner) (manipulator))
(request (assembler) (manipulator))
(transfer (assembler) (manipulator) (piece in grip))
```

Exemplo 1 - Troca de mensagens durante a interrupção de uma montagem.

4. Conclusão

A abordagem de sistemas Multi-Agentes apresenta-se como uma base interessante para o modelamento de arquiteturas para sistemas de visão computacional propositada, facilitando o mapeamento dos propósitos do sistema em comportamentos e explicitando a interação entre os comportamentos.

A arquitetura proposta possui várias características que a tornam eficiente e distinta, citando-se a possibilidade de cooperação entre processos, modularidade, flexibilidade e autonomia do sistema, permitindo a adição e remoção de agentes e facilitando a integração entre sistema e ambiente.

5. Referências

- [1] ALOIMONOS, Y. “What I have learned”, **CVGIP: Image Understanding**, v.60, n.1, p.74-85, July 1994.
- [2] RILLO, A. H. R. C., BIANCHI, R. A. C., MOREIRA Jr, B., FERRAZ, F. “Integrando Visão e Comportamento: Uma aplicação de reconstrução propositiva”. **XI. Congresso Brasileiro de Automática**, São Paulo, 2-6 Set., 1996. SBA, pp. 573-578.
- [3] BIANCHI, R. A. C.; RILLO, A. H. R. C. “A distributed control architecture for a purposive computer vision system”. In: IEEE SYMPOSIUM ON IMAGE, SPEECH AND NATURAL LANGUAGE SYSTEMS (ISNL) - IEEE INTERNATIONAL JOINT SYMPOSIA ON INTELLIGENCE AND SYSTEMS, 2nd., Rockville, Maryland, 1996. **Proceedings**. Los Alamitos, CA, IEEE Computer Society Press, 1996. P. 288-294.
- [4] BROOKS, R. A. “Intelligence without representation”. **Artificial Intelligence**, v.47, p.139-59, 1991.
- [5] ELFES, A. “A distributed control architecture for an autonomous mobile robot”. **Artificial Intelligence**, Computational Mechanics Publications, v.1, n.2, p 135-44 , 1986
- [6] BOISSIER, O., DEMAZEAU, Y. “A distributed artificial intelligence view on general purpose vision systems”. **Decentralized AI-3**, Amsterdam, p.311-330, 1992.
- [7] BOISSIER, O, DEMAZEAU, Y. “ASIC: An architecture for social and individual control and its application to Computer Vision”. European Workshop on Modeling Autonomous Agents in a Multi-Agent World, p.107-18, 1994.
- [8] RILLO, M., RILLO, A.H.R.C., COSTA, L.A.R. “The LSI assembly cell”. 7th. IFAC/IFIP/IFORS/IMACS/ISPE Symposium on information control problems in manufacturing technology, Toronto, p. 361-5, 1992.