

# **Arquitetura Multiagente Improvisacional: Transformando Planejamento em Improvisação e Introduzindo Improvisação nos Processos de Solução de Problemas**

**Márcia Cristina Moraes**

Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação  
Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática  
Porto Alegre, Brazil, 90619-900  
mmoraes@inf.ufrgs.br, mmoraes@inf.pucrs.br

e

**Antônio Carlos da Rocha Costa**

Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação  
Pontifícia Universidade Católica de Pelotas, Escola de Informática  
Pelotas, Brazil, 96010-000  
rocha@atlas.ucpel.tche.br

## **Abstract**

This paper proposes the understanding of improvisational agents as rational agents, through the presentation of an improvisational multi-agent architecture that incorporates improvisation processes in both, tactical level related to the building of agent's course of action, and strategic level related to problem-solving. In the first case, improvisation brings alternatives course of action executions, considering agent's plans as intentions. In the second case, improvisation brings the capability to handle problems that weren't anticipated in the planning of its course of action. The paper opposes this approach to probabilistic reasoning and re-planning, usually used in such situations. Improvisation allows agents to give rapid answers to unexpected situations, independently of having explicit knowledge directly applicable to such situations. To do that, agents use the resources readily available to them, using a process based on analogy by similarity.

**Keywords:** Artificial Intelligence, Autonomous Agents and Multi-Agent Systems, Planning and Scheduling, Agent Architectures, Improvisational Agents

## **Resumo**

Este artigo propõe o entendimento de agentes improvisacionais como agentes deliberativos, através da apresentação de uma arquitetura multiagente improvisacional que incorpora processos de improvisação tanto no nível tático da condução do curso de ação do agente quanto no nível estratégico da resolução de problemas. No primeiro caso, a improvisação visa proporcionar alternativas de execução a um determinado curso de ação, considerando os planos dos agentes como intenções. No segundo caso, ela traz a habilidade para tratar sub-problemas que não foram antecipados no planejamento do seu curso de ação. O artigo contrasta essa abordagem com as de raciocínio probabilístico e de re-planejamento normalmente utilizadas nessas situações. A improvisação possibilita aos agentes fornecerem respostas rápidas para situações inesperadas, independentemente de terem conhecimento explícito aplicável a tais situações. Os agentes usam os recursos diretamente disponíveis para eles, com base em um processo de analogia por similaridade.

**Palavras-Chave:** Inteligência Artificial, Agentes Autônomos e Sistemas Multiagentes, Planejamento, Arquiteturas de Agentes, Agentes Improvisacionais

## 1 Introdução

Tradicionalmente, um agente artificial tem as capacidades de raciocínio, aprendizagem e comunicação com outros agentes. O raciocínio é utilizado por agentes racionais para solucionar problemas de uma maneira correta e completa. Para fazer isto, os agentes racionais planejam seu curso de ação antecipadamente, tentando prever o futuro. Contudo, ambientes reais podem se modificar enquanto um agente está raciocinando sobre como alcançar algum objetivo, e estas mudanças podem invalidar as suposições sobre as quais o raciocínio do agente está fundamentado. Desta maneira, os agentes que operam em ambientes reais e dinâmicos precisam estar atentos a muitas situações inesperadas, que normalmente não surgem de maneira seqüencial, ou seja, os agentes precisam raciocinar sobre as suas ações [1]. Este raciocínio implica que os agentes devam saber quando novos fatos e oportunidades acontecem e como se adaptar à situação atual. Uma alternativa amplamente utilizada para dar conta desses ambientes dinâmicos é modelá-los através de recursos probabilísticos, ou então de replanejamento das ações, e realizar adaptação através de processos de aprendizagem [2]. Uma maneira alternativa, proposta por Hayes-Roth e Doyle [3], que é a adotada neste trabalho, é o uso de processos de improvisação. Em frente a situações inesperadas, que exigem respostas rápidas e espontâneas, as pessoas usam a sua capacidade de improvisação. Neste tipo de situação, muitas vezes é melhor improvisar do que re-planejar.

A capacidade de produzir respostas espontâneas é particularmente importante para agentes de interface que podem ser personificados através de personagens animados ou figuras humanas. Alguns pesquisadores têm mostrado que os usuários aplicam regras sociais aos computadores [4][5]. Deste modo, assim como os agentes humanos, os agentes racionais precisam apresentar comportamentos credíveis, interessantes e coerentes, trazendo a ilusão de vida para os agentes e fazendo o usuário suspender suas descrenças em relação aos agentes [6]. Para realizar isto, os agentes precisam agir de maneira credível tanto em frente a situações esperadas como em situações inesperadas.

Para serem credíveis no primeiro caso, os agentes podem usar o que chamamos de improvisação implícita. Por improvisação implícita, se entende a habilidade de incorporar cursos de ação previsíveis alternativos. Desta maneira, entendemos que planejamento se torna improvisação implícita, pois é mais razoável guiar agentes com um curso de comportamento abstrato do que dirigi-los através de um curso completo de ação. Improvisação implícita é tratada usando a noção de planos como intenção, ao invés de planos como programas, numa abordagem baseada em satisfação de restrições. No segundo caso, os agentes podem usar o que chamamos de improvisação explícita. Por improvisação explícita, se entende a habilidade de tratar um evento conhecido imprevisível. Com a improvisação explícita se tem o objetivo de introduzir o conceito de improvisação no processo de solução de problemas de agentes racionais, tratando improvisação, neste caso, como um processo baseado na analogia por similaridade.

Neste artigo apresentamos uma arquitetura multiagente improvisacional que incorpora as mudanças de planejamento para improvisação implícita e a introdução de improvisação explícita na solução de problemas de agentes racionais, baseados em uma arquitetura BDI. A arquitetura a ser apresentada possui seus conceitos de improvisação fundamentados no teatro improvisacional. Trabalhos relacionados com improvisação e teatro improvisacional, tais como Teatro Virtual [7], Oz [8] e Improv [9], enfocam somente eventos conhecidos previsíveis, usando abordagens como planejamento e improvisação implícita. Todos estes trabalhos enfocaram o papel dos atores e não examinaram o papel do diretor. Nossa proposta é inovadora, pois aborda improvisação explícita baseada nas habilidades do diretor em tratar problemas que não foram antecipados no momento do planejamento. No artigo, enfocamos o papel do diretor na arquitetura multiagente improvisacional, considerando sua capacidade de solucionar problemas que não foram previstos no momento da construção do seu curso de ação.

## 2 Improvisação, Satisfação de Restrições e Analogia por Similaridade

De acordo com Spolin [10], improvisação está relacionada à espontaneidade de agir em um mundo em constante movimento. Desta maneira, a idéia de improvisação e de teatro improvisacional está implicitamente apegada à representação como um comportamento informal e espontâneo que é realizado sem preparação prévia. O teatro tradicional, por outro lado, entende que a representação sempre é algo pensado, algo organizado antecipadamente. Chacra [11] salienta que representação improvisada e planejada são pólos diferentes de uma mesma matéria, determinadas por graus que fazem a apresentação teatral mais ou menos formalizada ou improvisada. Se os atores têm a intenção de usar improvisação eles estão implicitamente integrados no que é

chamado de Teatro Improvisacional. Desta maneira, eles não preparam antecipadamente todas suas falas e gestos, eles consideram o momento de espontaneidade. O momento de espontaneidade age em dois momentos de improvisação: implícita e explícita. A improvisação implícita envolve improvisação de texto (que ocorre através de um conjunto de frases e comportamentos não verbais que são deixados em aberto na peça para livre improvisação dos atores) e improvisação de personalidade (que está relacionada à maneira como o ator interpreta um personagem, considerando suas características físicas e psicológicas). Entendemos que este tipo de improvisação é realizado através da satisfação das restrições relacionadas ao texto (curso de ação) do agente e da sua personalidade. A improvisação explícita envolve improvisação de solução de problemas que ocorre quando um agente não tem um plano que pode ser imediatamente aplicado a uma situação inesperada. Neste caso, o diretor é responsável por auxiliar os atores a encontrar solução para problemas não inesperados. O diretor precisa de respostas rápidas para situações inesperadas e usa os recursos disponíveis para produzir estas respostas. Uma abordagem que fornece respostas rápidas e usa experiência passada (recursos) para produzir novas soluções para problemas inesperados é a analogia por similaridade [12].

## 2.1 Improvisação Implícita como Satisfação de Restrições

A improvisação implícita trata da questão da mudança de abordagem de planejamento para improvisação. Para entendermos como o planejamento se torna improvisação temos que considerar que o mundo real não é estático e completamente conhecido. Desta maneira, os agentes devem levar em conta as mudanças que podem ocorrer no ambiente enquanto eles raciocinam sobre como alcançar seus objetivos, ou seja, eles devem considerar a situação corrente no momento de agir. Vários autores apresentam abordagens diferentes do planejamento tradicional para tratar esta questão [13] [14] [15] [16] [17]. Para estes autores, o planejamento clássico não pode ser aplicado a ambientes dinâmicos e complexos, e a abordagem mais adequada é a improvisação.

Do ponto de vista do planejamento, existem duas maneiras de ver um plano. No planejamento clássico, um plano é visto como um programa e em uma abordagem alternativa um plano é visto como uma intenção. De acordo com Pfleger e Hayes-Roth [18] na visão de planos como programas, um plano é um programa executável composto por ações primitivas que um agente pode executar a fim de agir. Logo, o planejamento é um tipo de programação automática, e a aplicação do plano consiste simplesmente na sua execução direta. Na outra visão, um plano é um comprometimento a um objetivo que guia, mas não determina as ações específicas que um agente executa. Desta maneira, o agente não pode executar diretamente o seu plano, mas somente pode executar comportamentos, cada um dos quais podendo estar mais ou menos consistentes com os planos.

A visão de planos como programas possuem várias limitações como possibilidade de complexidade exponencial e inadequação para um mundo caracterizado por eventos imprevisíveis [13]. Planos como intenções superam estas limitações, pois um plano é um recurso que guia, através de ações abstratas, o que o agente deve fazer.

Nós iremos utilizar a segunda abordagem de planos como intenções para mostrar como planejamento se torna improvisação. Para alcançar suas intenções as pessoas podem ter uma idéia do que elas devem fazer e isto está relacionado a uma série de limitações e oportunidades que são chamadas de restrições. O agente tem a liberdade de improvisar, considerando as restrições presentes no momento. Esta visão não indica somente os objetivos do agente, mas também algum conjunto de comportamentos possíveis para alcançar estes objetivos [18].

Na nossa visão um agente tem alguma intenção e esta intenção poderia ser descrita como um *script* que informa algumas ações do que deve ser feito e alguns comportamentos que satisfazem estas ações. Estas ações são abstratas, descrevendo procedimentos gerais para se alcançar uma intenção. O agente somente pode escolher o que fazer e como fazer quando está frente a alguma situação. Uma intenção poderia ser entendida como um objetivo que é representado através de um *script* de alto nível que é instanciado com ações concretas de acordo com a situação atual. Este *script* de alto nível e as ações concretas são representados como improvisações que são realizadas através de satisfação de restrições.

De acordo com Marriott [19], restrições são formalizações matemáticas de relacionamentos que podem ocorrer entre objetos, elas são representadas através de um domínio de restrições que especifica as constantes, as funções e os relacionamentos de restrições que podem ocorrer entre objetos. A arquitetura proposta possui duas classes de restrições: restrições de ordem (relacionadas à ordem na qual algum conteúdo deve ser organizado e apresentado) e restrições de comportamento (relacionadas ao processo de selecionar comportamentos apropriados para executar).

A representação da intenção é baseada no modelo de regras de produção. De acordo com Stefik [20], cada regra de produção possui duas partes, chamada parte-se e parte-então. A parte-se da regra consiste de condições a serem testadas. Se todas as condições da parte-se de uma regra são verdadeiras, as ações da parte-então são

executadas. A diferença fundamental entre a nossa representação e as regras de produção tradicionais é que as ações da parte-se são comportamentos abstratos, ou ações de alto nível, que irão ser transformadas em ações primitivas durante a improvisação. Escolhemos utilizar o modelo de regras de produção, porque consideramos que esta é a representação mais apropriada para intenções considerando a abordagem do teatro improvisacional.

## 2.2 Improvisação Explícita como Analogia por Similaridade

Na improvisação explícita, o agente deve fazer uso dos recursos disponíveis para solucionar um problema de forma correta e rápida. Uma abordagem que fornece um mecanismo para efetivamente conectar um raciocínio passado com uma experiência presente, é a analogia. Na analogia os aspectos conhecidos de um novo caso são comparados com aspectos correspondentes de casos antigos. O caso antigo que possui a melhor combinação de aspectos correspondentes pode ser assumido como a melhor fonte de evidência para estimar os aspectos não conhecidos do novo caso. Quanto maior à concordância entre as alternativas dos aspectos não conhecidos, mais forte a evidência para uma conclusão do novo caso. De acordo com Russell [12], a analogia por similaridade fornece um mecanismo baseado na teoria de determinações [21][22] capaz de produzir respostas plausíveis e rápidas para problemas existentes em ambientes complexos.

De acordo com Russell [12] [21], um modelo de analogia simplificado em um banco de dados é o seguinte: existe um alvo T descrito por  $m$  pares de atributo-valor, para o qual se quer encontrar o valor de outro atributo Q. Existem várias fontes  $S_1, \dots, S_n$  (análogas) que têm valores para o atributo Q desejável bem como para os  $m$  atributos conhecidos para o alvo. A similaridade  $s$  é definida como o número de valores de atributos que combinam para um determinado alvo e fonte e a diferença é definida por  $d = m - s$ . Assumindo que existem  $r$  atributos relevantes para descobrir o valor de Q,  $p(d, r)$  é definida como sendo a probabilidade que a fonte S, diferindo do seu alvo em  $d$  atributos, se combina com os  $r$  atributos relevantes. A suposição de que não existe informação relevante significa que todos os atributos são igualmente importantes para serem relevantes. Pode-se calcular  $p(d, r)$  usando um argumento combinatório simples [12]. Seja  $N_m$  o número de escolhas de quais atributos são relevantes tais que S combina com T em tais atributos. Seja  $N$  o número total de escolhas de quais atributos são relevantes:

$$p(d, r) = N_m / N = \frac{\binom{m-d}{r}}{\binom{m}{r}} \quad (r \geq 1)$$

Esta probabilidade irá garantir que o análogo mais similar é a analogia mais apropriada.

## 3 Arquitetura Multiagente Improvisacional

Na arquitetura multiagente improvisacional proposta, os agentes são responsáveis pela criação e apresentação de algum conteúdo. A fim de realizar esta tarefa, eles assumem um dos três papéis: diretor, ator e diretor-ator. O diretor tem que coordenar os atores, informando aos mesmos seus cursos de ação, chamados *scripts*, e auxiliar os atores a resolver problemas inesperados. Os atores têm que seguir as instruções do diretor enquanto improvisam comportamentos apropriados a cada situação. O diretor-ator é um papel misto que os atores podem ter quando gerenciam ambas responsabilidades, de ator e de diretor. A arquitetura é improvisacional no sentido de que inclui improvisação tanto no ator quanto no diretor. Isto torna possível a incorporação de improvisação implícita (incluindo improvisação de texto e personalidade) relacionada a eventos previsíveis e conhecidos e improvisação explícita (significando improvisação para solução de problemas) relacionada a eventos conhecidos (através de experiências passadas) e não previstos no curso de ação atual do agente. Os atores e o diretor executam estes tipos de improvisação em diferentes níveis de abstração. O diretor está envolvido na fase preparatória da improvisação e o ator na fase de execução da improvisação.

Cada agente independente de seu papel está organizado em uma arquitetura de duas camadas. A camada de alto nível contém os processos relacionados às capacidades cognitivas dos agentes e a camada de baixo nível contém processos relacionados à percepção e ação no ambiente. A fim de suportar os dois tipos de improvisação mencionados anteriormente, a camada de alto nível é baseada nos processos do diretor improvisacional [10] e é composta por aquisição de conhecimento, construção das intenções e improvisação para solução de problemas, como mostra a figura 1.

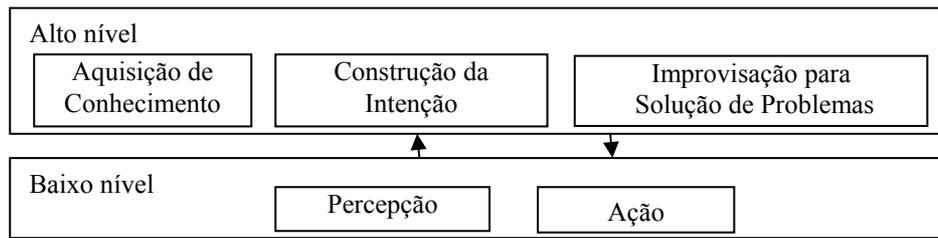


Figura 1. Arquitetura Multiagente Improvisacional

### 3.1 Intenção do Diretor

Para coordenar o comportamento dos atores, o diretor possui a sua própria intenção. O esquema da intenção do diretor pode ser visualizado na figura 2.

1. Executar processo de aquisição de conhecimento
2. Executar processo de construção das intenções para os atores
3. Enviar intenções (como scripts) para os atores
4. Esperar por uma percepção do ator solicitando ajuda
5. Enquanto a percepção é diferente de fim
  - Início
  - 5.1 Executa módulo de improvisação para solução de problemas para gerar a operação improvisada requisitada pelo ator.
  - 5.2 Enviar a intenção (como script) reformulada para o ator
  - 5.3 Esperar por uma nova percepção
  - Fim\_Enquanto
6. Fim\_Intenção

Figura 2: Intenção do Diretor

A improvisação implícita é realizada através dos módulos de aquisição de conhecimentos e construção da intenção, como satisfação de restrições. A improvisação explícita é realizada através do módulo de improvisação para solução de problemas, como analogia por similaridade. As próximas seções explicam o funcionamento dos módulos para o papel de diretor.

### 3.2 Módulo de Aquisição de Conhecimento

No processo de aquisição de conhecimento, o autor de uma apresentação fornece informação sobre a apresentação para o diretor. Esta informação pode ser uma seqüência de conteúdos que deve ser apresentada e falas relacionadas ao conteúdo. A seqüência pode ser informada em uma ordem completa ou parcial. Por exemplo, o autor pode dizer ao diretor que o ator tem que primeiro se apresentar dizendo alguma das seguintes falas: “*Olá, meu nome é Kira! Eu estou aqui para apresentar Porto Alegre para você.*” ou “*Olá! Eu sou Kira e estou aqui para falar de Porto Alegre para você.*”. Depois o ator pode escolher entre apresentar a história de Porto Alegre ou apresentar fatos sobre a localização de Porto Alegre. O autor também informa falas relacionadas a estes assuntos. A última ação é finalizar a apresentação dizendo uma das possíveis falas: “*Foi muito legal falar com você. Até uma próxima!*” ou “*Espero que tenha gostado da apresentação. Tchau!*”. No exemplo anterior, o autor humano informou uma ordem parcial das ações que o ator deve executar. O autor humano fixou a primeira e última ação e o ator tem que escolher a ordem das ações intermediárias. A figura 3 demonstra como a informação segue no sentido da aquisição de conhecimento e construção da intenção.

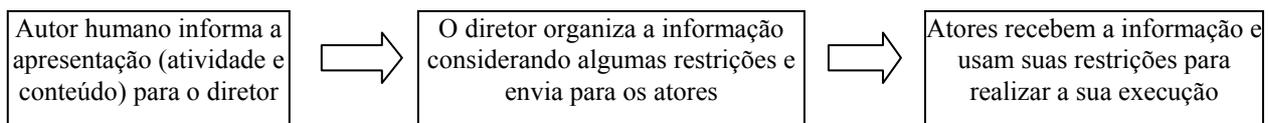


Figura 3: Fluxo de informação da aquisição de conhecimento para a construção da intenção

É importante observar que este é uma das maneiras na qual a informação flui. Em outros processos, o ator pode enviar informação para o diretor e o diretor para o ator humano.

Como mencionado da seção 2.1, pode-se pensar nos componentes do planejamento tradicional como algo que os agentes podem usar para guiar seu curso de ação e não como algo que será usado para planejar todo o curso de ação com antecedência. Desta maneira, os componentes de pré-condição, ação e efeitos, informados por um autor de uma apresentação, podem ser vistos como envolvidos na organização de um tipo de apresentação. Após receber a informação de um autor humano, o diretor reconhece esta informação como uma estrutura parcialmente ordenada representando ações e a usa para construir as intenções dinâmicas dos agentes.

### 3.3 Módulo de Construção da Intenção

Baseado nas informações obtidas pelo módulo de aquisição de conhecimento, o diretor executa o seu módulo de construção de intenções para construir as intenções para os atores. As intenções também são chamadas de *scripts* abstratos de comportamento, pois elas são planos abstratos que guiam, mas não determinam o comportamento do agente. O módulo é dividido em dois submódulos responsáveis pelo escalonamento das atividades e escalonamento dos comportamentos que irão compor uma intenção do agente (figura 4).

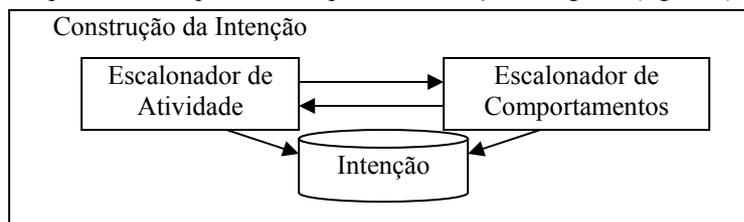


Figura 4: Módulo de Construção da Intenção

O módulo opera em um ciclo entre o escalonador de atividade e o escalonador de comportamento, que utilizam satisfação de restrições para construir a intenção do agente. O diretor opera sobre restrições de ordem para construir as intenções. Os tipos de restrições manipuladas são: pré-condições (indica as pré-condições relacionadas a ordenação de alguma execução); efeitos (que efeitos a execução de alguma atividade pode trazer e estado do script (que indica se o script está vazio ou não). A ativação do efeito dispara a satisfação de uma nova pré-condição, e assim o diretor vai construindo a intenção do agente. Alguns exemplos de restrições podem ser vistos na figura 5.

Ação	Exemplo de Restrições
<ul style="list-style-type: none"> <li>• Procure por atividade cuja pré-condição é nenhuma</li> <li>• Armazene um efeito intermediário, Armazene precedência no script, Chame o escalonador de comportamento e Armazene efeito no script</li> </ul>	<p>estado do script é igual a vazio</p> <p>estado do script é diferente de vazio e existe atividade cuja precedência é igual ao efeito corrente</p>

Figura 5: Exemplos de restrições utilizadas para construção da intenção

O escalonador de atividades é responsável por percorrer a estrutura de uma apresentação procurando por atividades que satisfazem as restrições de ordem apresentadas em um determinado momento. O escalonador de atividades funciona da seguinte maneira: procura por curso de ação igual ao objetivo informado; enquanto o efeito de uma atividade do curso de ação é diferente de fim, o diretor procura por atividades, que ainda não estejam na intenção, através de suas precedências. Ao encontrar, o diretor armazena a precedência na intenção, chama o escalonador de comportamento para armazenar a parte relativa à ação e armazena o efeito que será gerado após a execução da ação. Ao finalizar a construção da intenção como *script*, o diretor relaciona um ator ao *script* e grava a intenção em uma biblioteca de *scripts*.

O escalonador de comportamentos relaciona um conjunto de comportamentos possíveis para uma determinada atividade, considerando as restrições de comportamento que o agente possui. O ator, no momento da execução irá considerar suas restrições para escolher dentro do conjunto de comportamentos, o comportamento mais apropriado para uma determinada situação.

Através da interação entre o escalonador de atividades e o escalonador de comportamentos, é construída a intenção do agente como um conjunto de regras de decisão, que serão satisfeitas de acordo com a execução da intenção pelo ator. A estrutura da intenção pode ser visualizada na figura 6.

```
if <precedência1, ..., precedênciaN>  
then <conteúdo>  
    <efeito>
```

Figura 6: Estrutura de uma intenção

Precedência é a precedência de alguma atividade. Conteúdo é uma indicação de um processo que deve ser chamado no ator para escolher que conteúdo e ação específica deve ser executado. O ator tem que usar suas restrições para escolher quais conteúdos e ações executar, porque o diretor somente informa alguma ordem e classes de ações para o ator. Efeito é o efeito ou efeitos relacionados à execução de uma atividade. A ativação de um efeito influencia na satisfação de uma ou mais precedências. Em alguns casos, pode existir mais de uma precedência que seja satisfeita em um determinado momento. Quando um ator está executando seu *script* e isto acontece, ele deve escolher qual é a melhor opção de acordo com as suas restrições (limitações e oportunidades). O ator tem que improvisar a execução do seu *script* considerando as restrições de uma determinada situação.

Como se pode observar, a intenção representada como um *script* abstrato de comportamento é uma descrição abstrata do que e de como o ator irá executar uma determinada atividade. O diretor não dita o comportamento do ator. Ele fornece informações gerais e deixa o ator livre para agir de acordo com as suas restrições. O diretor e o ator estão trabalhando em conjunto para apresentar algum conteúdo ao usuário.

### 3.4 Módulo de Improvisação de Solução de Problemas

Após adquirir conhecimento, construir intenções como *scripts* abstratos de comportamentos e enviá-las ao ator, o diretor espera por uma percepção. A estrutura da percepção é composta por: um tipo de percepção, identificação do agente, e uma descrição de objeto. Se o tipo de percepção é *falha*, a identificação do agente indica qual agente solicitou a solução de problema e qual intenção (*script* abstrato de comportamento) necessita de improvisação. A descrição de um objeto tem o mesmo conjunto de atributos de um objeto meio (ver seção 3.4.1). A ação de *falha* é gerada por um ator quando ocorre um evento que não foi previsto no seu curso de ação. Este evento pode gerar duas situações diferentes. Na primeira situação, o agente pode ter conhecimento que pode ser diretamente aplicável para resolver o problema. Desta maneira, ele poderia usar re-planejamento, embora algumas vezes esta não seja a solução mais apropriada, como mencionada anteriormente. No outro caso, o re-planejamento não é aplicável, pois o agente não tem conhecimento de como agir na situação. Consideramos que em ambos os casos, o problema pode ser resolvido por improvisação baseada em analogia por similaridade.

#### 3.4.1 Objetos Fim e Objetos Meio

Para trabalhar com analogia por similaridade, assumimos que na analogia tanto os objetos fonte quanto os objetos alvo são descritos através de um modelo genérico de objetos. Neste modelo, os objetos são de dois tipos: fim e meio. Cada tipo de objeto é descrito usando um conjunto de atributos. Objetos classificados como fim são relacionados ao objetivo do agente, em um curso de ação específico. Objetos classificados como meio estão relacionados à maneira como um curso de ação específico é executado. O conjunto de atributos para objetos fim são: objetivo, efeito e características particulares. O conjunto de atributos para objetos meio são: usos possíveis e características particulares. O grupo de usos possíveis tem uma lista de usos para um objeto. O grupo de características particulares tem um conjunto de pares atributo-valor que especificam as características de um objeto. Os valores para os atributos (objetivo, efeito, lista de usos possíveis e características particulares) dependem do domínio da aplicação. Um objeto fim pode estar relacionado a objetos meio através das características particulares. Por exemplo, um objeto fim pode ter como característica particular o nome do objeto meio que poderia ser parte do seu curso de ação.

#### 3.4.2 Tipos de Improvisação para Solução de Problemas

Improvisação para solução de problemas pode ocorrer de duas maneiras: usando um *meio improvisado* como uma maneira alternativa para alcançar o objetivo atual ou usando um *fim improvisado* como um objetivo alternativo para alcançar o objetivo atual através de algum efeito do objetivo alternativo que seja compatível com o objetivo atual. No primeiro caso se possui improvisação no nível de meios e no segundo caso, improvisação no nível de objetivos.

### 3.4.3 Arquitetura do Módulo de Improvisação para Solução de Problemas

A figura 7 mostra a arquitetura do módulo de improvisação para solução de problemas, com suas entradas e saídas. O módulo recebe como entrada a descrição de um objeto que provoca a falha, os objetos que representam o conhecimento do agente e a intenção que deve ser reconsiderada. A descrição de um objeto problema segue o mesmo padrão de um objeto do tipo meio, logo é composta por usos possíveis e características particulares. Baseado nestas entradas, o módulo executa a reconsideração da intenção usando um processo de analogia por similaridade. Para realizar isto, a arquitetura do módulo é composta de três submódulos: identificação do tipo de improvisação, construção da analogia e transformação da analogia em intenção.

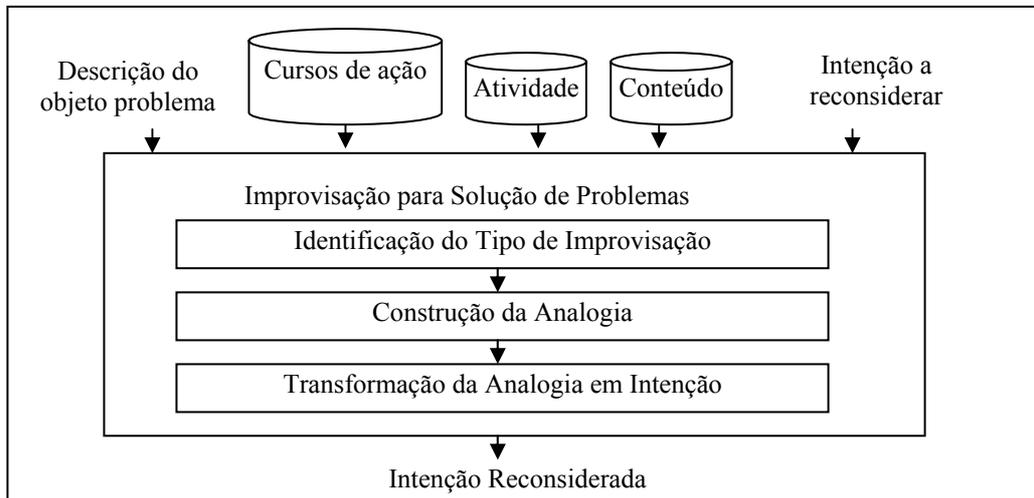


Figura 7: Módulos da Arquitetura de Improvisação para Solução de Problemas

#### 3.4.3.1 Módulo de Identificação do Tipo de Improvisação

Este submódulo recebe a descrição do objeto que ocasionou o problema e envia para o submódulo de construção da analogia, o tipo de improvisação a ser executada, uma lista de possíveis fontes análogas e a descrição do objeto problema. Para identificar o tipo de improvisação a ser executado, o submódulo busca por objetos fim (na base de dados de curso de ação) que tem um objetivo compatível com um dos usos possíveis presentes na descrição do objeto problema. Se um ou mais objetos compatíveis são encontrados, o tipo de improvisação é de meio. Neste momento, o submódulo procura por objetos meio (na base de dados de conteúdos) que tenham usos possíveis e características particulares compatíveis com a descrição do objeto problema. Este processo resulta em uma lista de possíveis objetos análogos.

Se nenhum objeto fim é encontrado, o tipo de improvisação é de fim. Neste caso, o submódulo procura por objetos fim (na base de dados do curso de ação) que tenham efeito compatível com os usos possíveis da descrição do objeto problema, construindo uma lista de objetos análogos possíveis. Se esta lista é vazia, ou seja, não existe objeto fonte com efeitos compatíveis com os usos possíveis da descrição do objeto problema, existe o que chamamos de improvisação sem efeitos conhecidos. Neste caso, o submódulo procura por objetos meio (na base de dados de conteúdos) que tenham características particulares semelhantes às características particulares do objeto problema. Este processo constrói a lista de objetos análogos possíveis.

Desta maneira, o submódulo implementa improvisação de meio e dois tipos de improvisação de fim: com efeito conhecido e sem efeito conhecido. Tanto a improvisação de meio quanto a improvisação de fim com efeitos conhecidos são resultado de uma analogia forte, enquanto improvisação de meio sem efeito conhecido é resultado de uma analogia fraca. Analogia é chamada de fonte quando existe um relacionamento entre os usos possíveis da descrição do objeto problema e os efeitos dos objetos fim fonte. Analogia é chamada de fraca quando não existe tal relacionamento, mas existe alguma analogia entre as características particulares. Estes dois tipos de analogias sempre possibilitam ao agente produzir improvisações (achar uma solução para o problema), mesmo quando a analogia encontrada não é a mais adequada.

#### 3.4.3.2 Módulo de Construção da Analogia

Este submódulo usa as idéias apresentadas por Russell [12] para selecionar o objeto análogo fonte mais adequado, dada a descrição de um objeto alvo. Deve-se definir a similaridade  $s$ , a diferença  $d$ , os atributos

relevantes  $r$  e a probabilidade  $p(d,r)$  para cada uma das fontes análogas encontradas pelo submódulo de identificação do tipo de improvisação. A similaridade  $s$ , os atributos alvo  $m$  e os atributos relevantes  $r$  são calculados para improvisação de meio e de objetivo sem efeito conhecido considerando o conjunto de atributos de características particulares e usos possíveis. Na improvisação com efeito conhecido somente o conjunto de atributos do tipo efeito é considerado. Isto ocorre porque quando um objetivo muda, as características particulares não necessariamente são compatíveis, mas os usos possíveis são compatíveis. Na improvisação de meio, o objetivo não muda, mas os meios podem ser diferentes do objetivo atual, desta maneira, o submódulo considera os usos possíveis para buscar por similaridades dentro dos objetos meio.

Após calcular a similaridade da fonte análoga, deve ser calculada a diferença entre os  $m$  atributos que descrevem o alvo e a similaridade  $s$ , como  $d = m - s$ . São assumido  $r$  atributos relevantes como um argumento de entrada e é calculada a probabilidade  $p(d,r)$  como a probabilidade de uma fonte  $S$ , diferindo do seu alvo em  $d$  atributos, combinar com os  $r$  atributos relevantes usando a fórmula apresentada na seção 2. A suposição de que não existe informação relevante, significa que todos os atributos são igualmente relevantes. O submódulo aplica a fórmula de probabilidade ( $p(d,r) = N_m/N$ ) para cada um dos objetos análogos possíveis presentes no conjunto construído pelo submódulo de identificação do tipo de improvisação. O objeto análogo escolhido será aquele que tem a maior probabilidade de ser similar ao alvo. Ao final da sua execução, o submódulo envia para o submódulo que transforma a analogia em intenção o tipo de improvisação e a fonte análoga selecionada como mais similar.

#### 3.4.3.3 Transformação da Analogia em Intenção

O submódulo recebe o tipo de improvisação e o objeto fonte que é a analogia mais apropriada e transforma esta analogia em intenção. Se o tipo de improvisação é de meio ou fim sem efeito conhecido, a intenção a ser desenvolvida é uma atualização da intenção atual. Esta atualização irá conter o objeto fonte mais similar ao objeto que ocasionou o problema. Se o tipo de improvisação é de fim com efeito conhecido, o submódulo constrói uma nova intenção baseada no objeto fim escolhido. Como mencionado anteriormente, um objeto fim tem um relacionamento com objetos meio através das suas características particulares. O submódulo ativa o módulo de construção da intenção, para que iniciando do objeto fim e dos seus relacionamentos a objetos meio, construa uma nova intenção e envie-a de volta para o submódulo de transformação da analogia em intenção.

## 4 Estudo de Caso: Museu Virtual SAGRES

O museu virtual SAGRES é um sistema que visa suportar o processo de aprendizagem através da interação entre visitantes e entre um visitante e os recursos do museu [23]. O SAGRES facilita a organização de visitas ao museu, apresentando as bases de informações do museu de maneira adaptada às características dos visitantes. A arquitetura proposta está sendo usada para criar guias virtuais e assistentes pessoais no SAGRES. Até o momento, realizamos três simulações do funcionamento da arquitetura para a apresentação de três conteúdos que não foram previstos no curso de ação inicial do agente. Analisamos em detalhe o módulo de improvisação de solução de problemas para os diferentes tipos de improvisação (meio, fim com objetivo conhecido e de fim sem objetivo conhecido) e verificamos que o módulo produz respostas coerentes para os testes aplicados, sempre escolhendo o objeto fonte análogo que corresponde à analogia mais similar.

Como considerado na seção 3.4.3.1, a improvisação de fim sem efeito conhecido pode produzir uma analogia fraca e isto aconteceu em uma das simulações. Embora a resposta produzida não fosse diretamente relacionada ao conteúdo solicitado, existia um relacionamento entre as características particulares do conteúdo, indicando que eles tem um mesmo super conjunto de conteúdos. Isto mostra que mesmo que os agentes não tenham conhecimento completo do conteúdo, eles podem produzir respostas plausíveis, prevenindo a sua falha de execução. Neste caso, os agentes podem expressar sua falta de conhecimento específico explicando que eles não tem conhecimento do conteúdo, mas conhecem um outro que é semelhante, em certas características, ao que foi requisitado. Fazendo isto, os agentes podem agir como os humanos agem na mesma situação, mostrando uma analogia a uma situação conhecida e apresentando um comportamento credível e esperto, produzindo a ilusão de vida tão desejada nos agentes computacionais.

## 5 Considerações Finais

A arquitetura multiagente improvisacional se baseia em uma arquitetura BDI tradicional [24] e pode ser mapeada para uma arquitetura BDI diagramática como apresentada em Wooldridge [25] com uma extensão. O processo de aquisição de conhecimento representa a função de revisão de crenças. A construção da intenção contém as funções de filtro e opções. Percepção representa o sensor de entradas e ação representa a função de ação. A extensão está presente no processo de improvisação para solução de problemas que contém as

características das funções opções e filtro para construir um curso de ação alternativo através de analogia por similaridade quando algo inesperado acontece por acidente e o agente não sabe como resolvê-lo.

Mostrar o mapeamento entre a arquitetura proposta e a arquitetura BDI diagramática de Wooldridge é importante a fim de inserir improvisação como uma característica fundamental de agentes racionais, aperfeiçoando suas capacidades de tratar problemas que não foram antecipados no planejamento do seu curso de ação. A escolha da arquitetura BDI para mostrar a possibilidade de usar improvisação em agentes racionais foi realizada devido à significância e reconhecimento do BDI para descrever comportamento de agentes racionais. Este mapeamento supera o vazio existente entre improvisação e inteligência artificial e mostra que improvisação é uma técnica natural de resolução de problemas.

Neste artigo apresentamos uma abordagem para incluir improvisação no processo de resolução de problemas em agentes racionais, considerando analogia e raciocínio baseado em casos como sinônimos. A analogia por similaridade foi escolhida porque acreditamos que ela é idéia mais próxima à improvisação. Improvisação necessita de respostas rápidas para situações inesperadas e usa os recursos disponíveis para produzir estas respostas. A analogia por similaridade fornece respostas rápidas [12] e utiliza experiências passada (recursos) para produzir novas soluções para problemas inesperados.

Com o módulo de improvisação de solução de problemas estendemos os trabalhos anteriormente realizados em improvisação, propondo uma abordagem para tratar eventos inesperados através de um agente diretor. Trabalhos anteriores somente consideraram agentes atores capazes de tratar eventos relacionados a situações esperadas através de planejamento e improvisação implícita. Estendemos o uso da improvisação para a geração de comportamento improvisado em resposta a situação inesperada que pode ocorrer quando o agente está executando o seu curso de ação. A habilidade de produzir respostas rápidas e coerentes para situações inesperadas é fundamental para os agentes, especialmente aqueles responsáveis pela interação com usuário, e o uso de improvisação na solução de problemas aperfeiçoa esta habilidade.

As simulações executadas no SAGRES mostram que o módulo de improvisação para solução de problemas pode produzir respostas relevantes para eventos não previstos. Contudo, com estas simulações não é possível realizar considerações sobre a eficiência do módulo. Estamos realizando novos testes a fim de comparar a performance da arquitetura de guias virtuais com e sem o módulo de improvisação e observar resultados mais completos e adequados em relação à capacidade do agente em resolver problemas inesperados.

## Referências

- [1] Pollack, M. E. The use of plans. *Artificial Intelligence*, Vol. 57. Elsevier Science Publishers (1992) 43-68
- [2] Russell, S.; Norvig, P. *Artificial Intelligence: a modern approach*. Upper Saddle River. (2003)
- [3] Hayes-Roth, B., Doyle, P. Animated Characters. In: *Autonomous Agents and Multi-Agent Systems*, Vol. 1, Kluwer Academic Publishers, (1998) 195-230
- [4] Ball, G.; Ling, D.; Kurlander, D.; Miller, J.; Pugh, D.; Skelly, T.; Stankosky, A.; Thiel, D.; Van Dantzich, M.; Wax, T.: *Lifelike Computer Characters: The Persona Project at Microsoft*. In: *Software Agents*. Menlo Park, California: AAAI Press. (1997)
- [5] Koda, T. *Agents with Faces: A Study on the Effects of Personification of Software Agents*. Master Thesis, MIT Program in Media Arts and Sciences. (1996)
- [6] Loyall, B.: *Believable Agents: Building Interactive Personalities*. PhD Thesis Carnegie Mellon University, Technical Report CMU-CS-97-123. (1997)
- [7] Hayes-Roth, B.; Browston, L.; Sincoff, E. *Directed Improvisation by Computer Characters*. Technical Report KSL-95-04 – Stanford University. (1995)
- [8] Bates, J. *The Nature of Characters in Interactive Worlds and The OZ Project*. Technical Report CMU-CS-92-200 – Carnegie Mellon University. (1992)
- [9] Perlin, K.; Golberg, A. *Improv: A System for Scripting Interactive Actors in Virtual Worlds*. In: *Computer Graphics*, vol. 29. (1996)
- [10] Spolin, V. *Improvisation for the Theater: A Handbook of Teaching and Directing Techniques*. First Edition. Northwestern University Press. (1963)
- [11] Chacra, S.: *Natureza e Sentido da Improvisação Teatral*. Editora Perspectiva. (1983)

- [12] Russell, S.: *Analogy by Similarity*. In David Helman (Ed.), *Analogical Reasoning*, Boston, MA: D. Reidel. (1988)
- [13] Agre, P. E. *The Dynamic Structure of Everyday Life*. Phd Thesis, MIT Artificial Intelligence Laboratory, Technical Report 1085 (1988)
- [14] Agre, P. E.: *Computation and Human Experience*. Cambridge University Press (1997)
- [15] Agre, P. E., Chapman, D.: *Pengi: An Implementation of a Theory of Activity*. Sixth National Conference on Artificial Intelligence. Morgan Kaufmann Publishers. Vol. (1987) 268-272
- [16] Anderson, J. E.: *Constraint Directed Improvisation for Everyday Activities*. Phd Thesis University of Manitoba. (1995)
- [17] Suchman, L. A.: *Plans and Situated Actions: The problem of human machine communication*. Cambridge: Cambridge University Press (1987)
- [18] Pflieger, K., Hayes-Roth, B.: *Plans Should Abstractly Describe Intended Behavior*. In Alex Meystel, Jim Albus, and R. Quintero (eds.): *Intelligent Systems: A Semiotic Perspective*, Proceedings of the 1996 International Multidisciplinary Conference, Vol. 1 (1996) 29-34.
- [19] Marriott, K., Stuckey, P. J.: *Programming with Constraints: An Introduction*. MIT Press, Cambridge Massachusetts (1998)
- [20] Stefik, M.: *Introduction to Knowledge Systems*. Morgan Kaufmann Publishers, Inc. San Francisco (1995)
- [21] Russell, S.: *A Quantitative Analysis of Analogy by Similarity*. In Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelphia, PA: Morgan Kaufmann. (1986)
- [22] Davies, T. R.; Russell, S.: *A Logical Approach to Reasoning by Analogy*. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, Milan, Italy: Morgan Kaufmann,. (1987)
- [23] Bertolotti, A. C.; Costa, A. C. R. *SAGRES – A Virtual Museum*. In: *Museums and The Web 1999 Conference*. USA: New Orleans, Louisiana. (1999)
- [24] Bratman, Michael E.; Israel, David J.; Pollack, Martha E. *Plans and Resource-Bounded Practical Reasoning*. *Computational Intelligence*, 4 (4). (1988) 349-355
- [25] Wooldridge, Michael. *Intelligent Agents*. In: *Multiagent systems: A Modern Approach to Distributed Artificial Intelligence*. Gerhard Weiss (editor). The MIT Press, Cambridge, Massachusetts. (1999) 27-77