

Extensão do Cálculo de Situações Aplicada à Gestão de Projetos

Fabrizio Jailson Barth e Edson Satoshi Gomi

Laboratório de Técnicas Inteligentes

Escola Politécnica da Universidade de São Paulo

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

fabrizio.barth,edson.gomi@poli.usp.br

Resumo. O Cálculo de Situações tem sido utilizado como uma ferramenta para desenvolver sistemas de planejamento. Entretanto, o Cálculo de Situações tradicional não prevê mecanismos para representar ações que consomem recursos. Essa característica é importante em aplicações na área de Gestão de Projetos. Neste trabalho é apresentada uma extensão do Cálculo de Situações que permite representar ações que consomem recursos humanos e materiais. O uso dessa extensão do Cálculo de Situações é exemplificado através de uma aplicação de planejamento de projetos na área de Telecomunicações.

1 Introdução

Um agente inteligente deve ser capaz de planejar suas ações com antecedência. Tal habilidade é essencial ao comportamento inteligente e sua implementação é extremamente importante em aplicações práticas, e.g. gestão de projetos, robótica, manufatura, logística, missões espaciais, etc.

Vários algoritmos de planejamento foram propostos nos últimos anos. Aqueles provados corretos possuem grandes limitações, em particular, quanto à representação de ações e desempenho computacional, conseqüentemente, não podem ser usados facilmente para resolver problemas do mundo real. Por outro lado, os planejadores práticos, capazes de resolver problemas grandes, em geral, foram construídos de maneira *ad-hoc*, sendo difícil adaptá-los para aplicações para as quais eles não foram construídos.

Um exemplo de aplicação prática é aquela voltada a Gestão de Projetos. A gestão de projetos é um domínio que exige, entre outras características, a capacidade de representar ações concorrentes, ações que consomem recursos e ações que possuem duração.

Através do uso de lógica formal é possível construir planejadores corretos, fundamentados em princípios bem conhecidos, que podem ser facilmente validados, mantidos e modificados. Resta investigar, se tais planejadores podem ser executados em domínios reais, ou seja, possam representar todas as informações relevantes ao problema, como por exemplo, para o domínio de gestão de projetos, ações que consomem recursos. Também, é necessário verificar a eficiência de planejadores baseados em lógica aplicados a problemas reais, ou seja, avaliar a sua performance.

Com o objetivo de representar ações e seus efeitos, *McCarthy e Hayes* [6] propuseram uma linguagem baseada em lógica de primeira ordem, chamada de Cálculo de Situações. O

cálculo de situações possui algumas restrições que impedem a representação de ações de um domínio prático, por exemplo, a incapacidade de representar ações que consomem recursos, ações concorrentes e o tempo.

É com o objetivo de construir uma arquitetura, baseada em lógica formal, capaz de planejar e fornecer prognósticos em um domínio prático de Gestão de Projetos, que este trabalho tende, inicialmente, extender o cálculo de situações para representar ações que consomem recursos.

Este trabalho é organizado do seguinte modo: na seção 2 é apresentado conceitos sobre Gestão de Projetos; na seção 3 é apresentado o cálculo de situações; na seção 4 é apresentada a proposta de adição da representação de ações que consomem recursos ao cálculo de situações; na seção 5 é demonstrado como este formalismo estendido pode ser utilizado como um programa lógico; na seção 6 é apresentado um exemplo de implementação simplificada no domínio de gestão de projetos, e; na seção 7 são apresentadas as conclusões deste trabalho.

2 Gestão de Projetos

As organizações executam trabalhos que envolvem, geralmente, projetos [3]. Os projetos são caracterizados por serem executados por pessoas, serem restringidos por recursos limitados, e são planejados, executados e controlados.

Um projeto pode assim ser definido: *um projeto é uma empreitada temporária criada para gerar um produto ou um serviço único*. Sendo que, *temporário* significa que cada projeto tem um começo e um fim definidos. E, *único* significa que o produto ou o serviço é diferente, de alguma maneira, de todos os produtos ou serviços similares já produzidos.

Os projetos são realizados em todos os níveis da organização. Podem envolver uma única pessoa ou milhares, podem envolver uma única unidade de uma organização ou podem cruzar os limites da organização. Os projetos são frequentemente componentes críticos da estratégia de negócio uma organização. Os exemplos dos projetos incluem: desenvolver um novo produto ou serviço, efetuar uma mudança na estrutura da organização, projetar um veículo novo, entre outros.

A Gestão de Projetos é a aplicação do conhecimento, das habilidades, das ferramentas, e das técnicas para projetar atividades a fim de resolver ou exceder as necessidades dos interessados no projeto. A atividade de resolver as necessidades dos interessados pelo projeto envolve balancear demandas competitivas entre: escopo, tempo, custo e qualidade; pessoas com diferentes necessidades e expectativas; exigências identificadas (necessidades) e exigências não identificadas (expectativas), entre outras.

Uma tarefa importante na gestão de projetos é a realização de um plano para a execução do projeto. Muitas vezes esta tarefa de planejamento não é uma tarefa trivial, por se tratar de um domínio complexo com muitas variáveis.

Um exemplo de plano de um projeto é o apresentado na figura 1. Este plano representa a construção de uma estação rádio base. Uma estação rádio base é um produto desenvolvido pelas empresas de telecomunicações, e tem como objetivo tornar possível a comunicação entre os aparelhos da telefonia móvel.

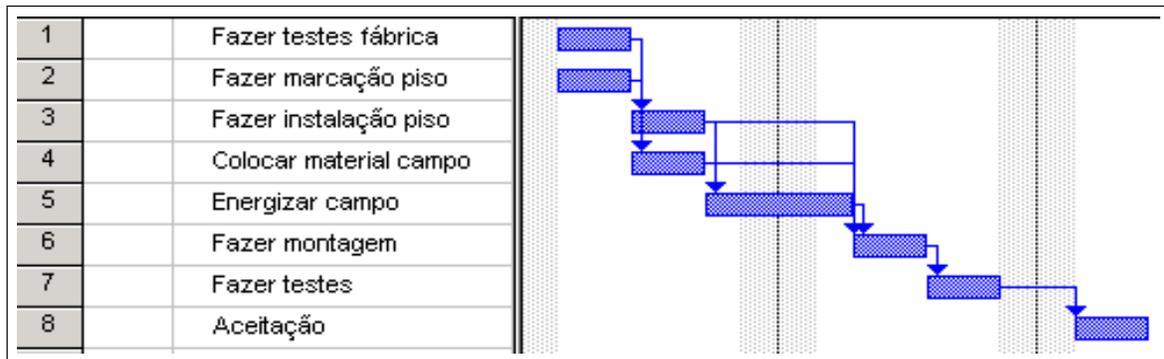


Figura 1: Gantt Chart de um exemplo simplificado da construção de uma estação rádio base

O exemplo apresentado na figura 1 é um exemplo simplificado e representa um conjunto de tarefas e seus relacionamentos de precedência.

3 Cálculo de Situações

Shanahan [8], apresenta o cálculo de situações como sendo um formalismo lógico cujos os elementos são ações, efeitos e situações:

- *situações*: uma descrição completa de uma determinada situação do universo;
- *fluentes*: são funções que denotam propriedades ou relacionamentos que podem mudar de uma situação para outra;
- *ações*: que transforma uma situação em outra;
- função $Result(\alpha, s)$: denota a situação resultante da execução da ação α em uma determinada situação s ;
- predicado $Hold(\delta, s)$: estabelece que o fluente δ vale na situação s ¹.

Além destes elementos, o cálculo de situações possui as seguintes classes de fórmulas: *axiomas de efeito*, *restrições de domínio* e *sentença de observação*.

Os *axiomas de efeito* possuem a forma $\forall s[Hold(\delta, Result(\alpha, s))]$ ou a forma $\forall s[Hold(\delta, Result(\alpha, s)) \leftarrow Hold(\delta_i, s)]$ e tem o objetivo de descrever os efeitos das ações. Por exemplo, para declarar que ao realizar a tarefa de “fazer testes fábrica” os testes da fábrica estarão feitos, deve-se utilizar o axioma de efeito para a tarefa de “fazer testes fábrica” da seguinte maneira: $Hold(teste_fabrica_feito, Result(fazer_testes_fabrica, S))$.

Uma *restrição de domínio* é uma fórmula que possui a forma $\forall s[Hold(\delta, s) \leftarrow Hold(\delta_i, s)]$ e tem o objetivo de descrever eventuais restrições de domínio.

Uma coleção de axiomas de efeito e restrições de domínio é chamada de *descrição do domínio*. Na descrição de domínio deve-se ter um axioma de efeito para cada ação.

As *sentenças de observação* são fórmulas que possuem a seguinte forma $Hold(\delta, \sigma)$, onde σ é um termo situação não variável, por exemplo: $Hold(pronto_para_comecar, S0)$.

¹Suponha que queira-se representar o fato que está chovendo na situação $S0$. Um caminho para isto, é usar a notação $Chovendo(S0)$. Outra alternativa é expressar o mesmo fato como $Hold(Chovendo, S0)$.

Este tipo de fórmula costuma aparecer em situações iniciais e descrevem o que é verdade em uma situação já instanciada.

O cálculo de situações possui como características, entre outras, a não capacidade de representar o tempo, isto significa dizer que informação do tipo *está chovendo ao meio dia* não é possível de ser declarada. Também, não representa ações concorrentes e não expressa ações com duração [8, 1, 2].

Desta forma, para poder aplicar o cálculo de situações como formalismo para representar as ações e mudanças em um domínio de gestão de projetos, faz-se necessária a extensão do mesmo.

4 Representação de recursos e ações que consomem recursos

A palavra *recursos* em planejamento e escalonamento se refere a objetos que podem ser produzidos, consumidos, ou “pedidos” durante a execução de planos, restringindo possíveis ações [7, 4, 5]. E.g., dirigir um carro requer e consome combustível, construir uma casa requer e consome tijolos, etc.

Para representar recursos no domínio modelado, é necessário que a linguagem utilizada possa expressar uma pré-condição do tipo $ter(R\$2, 50)$ e o algoritmo de planejamento utilizado possa executar isto eficientemente.

Recursos são normalmente representados como variáveis que podem assumir valores reais ou inteiros [7, 4, 5]. Neste trabalho, assumir-se-á que todos os recursos são representados através de valores monetários. Visto que, em um domínio de Gestão de Projetos, todos os recursos são facilmente convertidos para valores monetários, por exemplo: uma tarefa para ser executada necessita de uma quantidade de material, esta quantidade é facilmente convertida em moeda.

Para representar a quantidade de recurso que uma ação α necessita para ser executada, tem-se a fórmula

$$\forall \alpha, \exists \xi [Resource(\alpha, \xi)] \quad (1)$$

Onde $\xi \geq 0$ e $\xi \in \mathbb{R}^2$.

Esta nova fórmula é adicionada ao axioma de efeito, fazendo com que o axioma de efeito possua uma nova forma:

$$\forall s [Holds(Total, \delta, Result(\alpha, s)) \leftarrow Holds(\beta, \delta_i, s) \wedge Resource(\alpha, \xi)] \quad (2)$$

Onde $Total$ é uma variável que informa sobre o recurso que está sendo gasto para a execução na situação s . O resultado de $Total$ é obtido através da soma de β com ξ .

A fórmula (2) diz que a ação α para ser executada *requer* o recurso ξ e durante a execução *consome* o recurso ξ . Por exemplo: $Holds(Ter(x), Result(Comprar(x), s)) \leftarrow Resource(Comprar(x), y)$.

Isto quer dizer que, para comprar um objeto x , deve-se ter o recurso suficiente para executar tal ação.

Sendo assim, a adição desta nova fórmula faz com que a teoria, além de possuir axiomas de efeito, restrições de domínio e sentenças de observação, possua também *sentenças de observação sobre os recursos*, cuja as fórmulas possuem a forma apontada pela fórmula (1).

² \mathbb{R} significa o conjunto dos números reais

5 Planejador baseado no Cálculo de Situações Estendido

Utilizando o cálculo de situações, pode-se construir programas lógicos que podem ser usados para representar efeitos de ações, para planejar ou para realizar prognósticos.

Shanahan [8] descreve um programa de cálculo de situações como sendo uma conjunção de: um *axioma de quadro universal*³, na forma $Holds(\delta, Result(\alpha, s)) \leftarrow Holds(\delta, s) \wedge notAffects(\alpha, \delta, s)$, onde o predicado $Affects(\alpha, \delta, s)$ define que o fluente δ é afetado pela ação α na situação s ; um conjunto finito de sentenças de observação, com o formato: $Holds(\delta, S0)$; um conjunto finito de axiomas de efeito, com o formato: $Holds(\delta, Result(\alpha, s)) \leftarrow Holds(\delta_i, s)$; um conjunto finito de cláusulas $Affects$; e um conjunto finito de sentenças de fundo que não mencionam os predicados $Holds$ ou $Affects$.

Assim sendo, tem-se:

Definição 1. *Um programa de cálculo de situações que implementa ações consumidoras de recursos é uma conjunção de:*

- um conjunto finito de sentenças de observação, com o seguinte formato:

$$Holds(\beta, \delta, S0) \quad (3)$$

onde β é o valor dos recursos gastos na situação inicial sem a aplicação de nenhuma ação.

- um conjunto finito de axiomas de efeito, com o seguinte formato:

$$Holds(Total, \delta, Result(\alpha, s)) \leftarrow \Pi \wedge Resource(\alpha, \xi) \quad (4)$$

onde Π não menciona o predicado $Affects$ e toda ocorrência do predicado $Holds$ em Π é na forma $Holds(\beta, \delta_i, s)$, onde $\beta + \xi = Total$.

- um conjunto finito de cláusulas $Affects$, com o seguinte formato:

$$Affects(\alpha, \delta, s) \quad (5)$$

- o axioma de quadro universal:

$$Holds(Total, \delta, Result(\alpha, s)) \leftarrow Holds(\beta, \delta, s) \wedge notAffects(\alpha, \delta, s) \wedge Resource(\alpha, \xi) \quad (6)$$

onde $\beta + \xi = Total$.

- um conjunto finito de sentenças de fundo que não mencionam os predicados $Holds$ ou $Affects$, e;
- um conjunto finito de sentenças de observação sobre os recursos:

$$Resource(\alpha, \xi) \quad (7)$$

onde α são todas as ações existentes no domínio.

³Quando o cálculo de situações é usado para descrever efeitos de ações, além de descrever o que muda é necessário descrever também aquilo que permanece inalterado (problema do quadro). A fim de descrever o que permanece inalterado, o cálculo de situações possui o axioma de quadro universal [8].

Estas fórmulas podem ser facilmente transformadas em cláusulas de um programa lógico, por exemplo, para a fórmula (3), tem-se uma codificação que pode ser visualizada na figura 2. Onde, *pronto_para_comecar* é um fluente válido na situação inicial *s0*.

```
1 holds(0,pronto_para_comecar,s0).
```

Figura 2: Codificação da Sentença de Observação (estado inicial)

Um outro exemplo é a codificação dos axiomas de efeito (4), que pode ser visualizado na figura 3. Onde *fazer_instalacao_piso* é uma ação que somente pode ser executada na situação *S* se os fluentes *teste_fabricafeito* e *marcacao_pisofeito* forem válidos e, também, gasta um total de recursos que é a soma dos recursos gastos para tornarem os fluentes *teste_fabricafeito* e *marcacao_pisofeito* verdadeiros (*Y*) mais a quantidade de recurso gasto para executar a ação *fazer_instalacao_piso* (*X*).

```
1 holds(Total,instalacao_pisofeito,result(fazer_instalacao_piso,S)):-
2     holds(Y,teste_fabricafeito,S),
3     holds(Y,marcacao_pisofeito,S),
4     resource(fazer_instalacao_piso,X),
5     Total is Y + X.
```

Figura 3: Codificação do Axioma de Efeito

6 Implementação do Cálculo de Situações extendido no domínio de Gestão de Projetos

Com o objetivo de esclarecer a aplicabilidade de tal extensão do cálculo de situações, nesta seção será demonstrado a representação das ações aplicadas em um projeto de construção de estação rádio base simplificado, como visto na figura 1.

Assim sendo, as informações sobre as ações do projeto que devem ser representadas são: a ordem de precedência entre as ações, ou seja, as pré-condições de cada ação; a descrição dos efeitos que as ações podem gerar, seja ele positivo⁴ ou negativo⁵; e o recurso (expresso em unidades monetárias) que cada ação necessita. Estas informações podem ser visualizadas na tabela 1.

Tabela 1: Descrição das ações do domínio

| Ação | Pré-condições | Efeitos | Recursos |
|------------------------|---|--|----------|
| fazer_testes_fabrica | pronto_para_comecar | teste_fabricafeito | 10 UM |
| fazer_marcacao_piso | pronto_para_comecar | marcacao_pisofeito | 15 UM |
| fazer_instalacao_piso | teste_fabricafeito marcacao_pisofeito | instalacao_pisofeito ¬ marcacao_pisofeito | 10 UM |
| colocar_material_campo | teste_fabricafeito | material_em_campo | 20 UM |
| energizar_campo | instalacao_pisofeito | energia_pronta | 30 UM |
| fazer_montagem | material_em_campo, energia_pronta instalacao_pisofeito | montada | 20 UM |
| fazer_testes | montada | testes_ok | 15 UM |
| aceitacao | testes_ok | estacao_radio | 10 UM |

⁴Efeitos que modificam o valor dos fluentes para verdadeiro.

⁵Efeitos que modificam o valor dos fluentes para falso.

Através dos axiomas de efeito, pode-se descrever os efeitos positivos de uma ação e quais pré-condições são necessárias para a sua execução, e através do predicado *Affects* pode-se descrever os efeitos negativos. Um exemplo é o trecho de código apresentado na figura 3 e na figura 4.

```
1 affects(fazer_instalacao_piso,marcacao_piso_feito,S).
```

Figura 4: Codificação do predicado Affects

Além disto, é necessário declarar qual o custo para cada ação, figura 5.

```
1 resource(fazer_testes_fabrica,10).
2 resource(fazer_marcacao_piso,15).
3 resource(fazer_instalacao_piso,10).
4 resource(colocar_material_campo,20).
5 resource(energizar_campo,30).
```

Figura 5: Codificação do predicado Resource

Após codificadas as cláusulas que correspondem as sentenças de observação, os axiomas de efeito, o axioma de quadro universal, as cláusulas *Affects* e as sentenças de observação sobre os recursos, de acordo com a definição 1. Tem-se uma base de conhecimento que aplicando sobre ela um mecanismo dedutivo de inferência, consegue-se gerar planos, como também fornecer prognósticos, dependendo da cláusula objetivo que for fornecida.

Por exemplo, se a cláusula objetivo for $holds(Total, W, Result(fazer_testes_fabrica, s0))$, o mecanismo de inferência dedutivo aplicado sobre a base de conhecimento é capaz de retornar um prognóstico da situação resultante. Na figura 6 é possível visualizar a árvore de busca da cláusula objetivo $holds(Total, W, Result(fazer_testes_fabrica, s0))$. A resposta para esta cláusula objetivo é $Total = 10$ e $W = teste_fabrica_feito$.

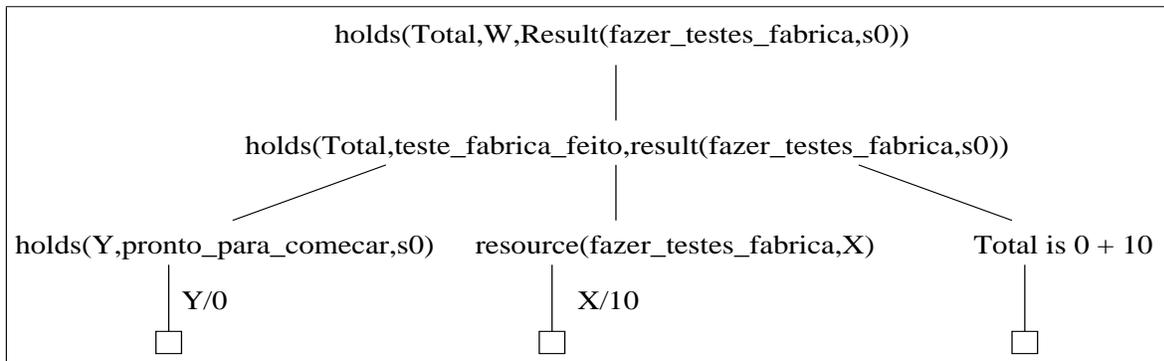


Figura 6: Exemplo da árvore de busca do programa

Como planejador, pode-se montar a cláusula da seguinte maneira: $holds(Total, energia_pronta, X)$, onde pergunta-se por um plano que torne verdadeiro o fluente *energia_pronta*, e este plano é retornado no formato $result(\alpha_1, \dots, result(\alpha_m, S0))$ unificado com a variável *X*. E a variável *Total* retorna o valor total de recursos gastos pelo plano.

7 Conclusões

Neste trabalho foi apresentada uma extensão do cálculo de situações, que torna possível a representação de ações consumidoras de recursos.

Através do exemplo implementado, constatou-se, empiricamente, que o Cálculo de Situações estendido pode ser utilizado em domínios que exijam a representação de ações que consomem recursos, em especial na área de Gestão de Projetos.

Devido a incapacidade de representação do tempo no cálculo de situações, esta extensão não possibilita a representação de quantos recursos são gastos em determinado instante de tempo. Apenas, possibilita a representação dos recursos gastos para a completa execução do plano.

Com os resultados deste trabalho, ainda não é possível representar ações ou construir um planejador para o domínio de gestão de projetos. É necessário, além desta extensão, representar ações concorrentes, ações com duração, entre outras.

Assim sendo, como trabalhos futuros, tem-se o objetivo de fazer com que o cálculo de situações possa representar o tempo, represente ações com duração e ações concorrentes, com a finalidade de construir uma arquitetura, baseada em um formalismo lógico, capaz de executar tarefas de planejamento no domínio de Gestão de Projetos.

Referências

- [1] José Júlio Alferes, Renwei Li, and Luís Moniz Pereira. Concurrent actions and changes in the situation calculus. *Proc. of IBERAMIA*, pages 93–104, 1994.
- [2] Jorge Baier and Javier Pinto. Non-instantaneous actions and concurrency in the situation calculus. *In 10th European Summer School in Logic, Language and Information*, 1998.
- [3] Project Management Institute Standards Committee. *A Guide to the Project Management Body of Knowledge*. Automated Graphic Systems, Maryland, USA, 1996.
- [4] K. Currie and A. Tate. O-plan: the open planning architecture. *Artificial Intelligence*, 52(1):49–86, 1991.
- [5] A. El-Kholy and B. Richards. Temporal and resource reasoning in planning: the parplan approach. *Proc. ECAI-96*, pages 614–618, 1996.
- [6] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.
- [7] Stuart J. Russel and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice-Hall, 1995.
- [8] Murray Shanahan. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. The MIT Press, 1997.