

# ***Atena: um sistema para suporte ao planejamento na área de Gestão de Projetos***

**Fabício Jailson Barth , Edson Satoshi Gomi**

<sup>1</sup>Laboratório de Engenharia de Conhecimento (KNOMA)  
Departamento de Computação e Sistemas Digitais  
Escola Politécnica da Universidade de São Paulo  
Av. Prof. Luciano Gualberto, 158, tv. 3 – 05508-900 São Paulo, SP

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

**Abstract.** *This paper presents a system aimed at supporting project managers in the design of activities' networks. The system framework uses a partial order planner that builds project plans starting from enterprise activities. These activities are modeled according to the definitions of the TOVE Ontology and are formally represented using STRIPS notation.*

**Resumo.** *Este artigo apresenta um sistema que, com base no conhecimento da empresa, visa auxiliar o gerente de projetos na construção de modelos de redes de atividades. O arcabouço construído utiliza um algoritmo de ordem parcial que manipula informações sobre as atividades da empresa, codificadas utilizando uma notação desenvolvida a partir da representação STRIPS e da definição de atividade encontrada na ontologia TOVE.*

## **1. Introdução**

Ao longo dos anos, vem se percebendo que a vitalidade de uma empresa depende muito da sua capacidade de inovação, produção, qualidade dos seus produtos, entre outros fatores. Cada vez mais, novos produtos vem surgindo com mais rapidez, o ciclo de vida destes produtos vem diminuindo e o cliente vem se tornando cada vez mais exigente quanto a qualidade do produto. Logo, as empresas são forçadas, a diminuir o seu tempo de projeto dos produtos, aumentar o controle da qualidade, assim como aumentar o controle dos custos, entre outros fatores.

A definição dos métodos de Gestão de Projetos começa com a definição conceitual dos processos. Além destas definições conceituais, desenvolveu-se ao longo dos anos um conjunto de ferramentas para que os processos de Gestão de Projetos fossem apoiados por computador, com o objetivo de melhorar a performance das atividades de Gestão de Projetos. Da mesma forma, ao longo dos anos vem sendo desenvolvidos sistemas baseados em conhecimento, que tentam atribuir aos sistemas computacionais um maior poder de raciocínio, e portanto, capacitando tais sistemas na realização de tarefas mais complexas dentro de uma empresa [O'Leary, 1998].

O objetivo deste trabalho é desenvolver um sistema que, com base no conhecimento da empresa, histórico de projetos já executados, auxilie o gerente de projetos no desenvolvimento de modelos de redes de atividades (cronograma).

Esta ferramenta deve atuar sobre uma base de conhecimento que descreve o conjunto de atividades que uma empresa é capaz de executar. O sistema recebe como entrada uma descrição da situação atual e uma descrição dos objetivos a serem atingidos, e devolve um ou mais planos possíveis, descrevendo o conjunto de atividades que devem ser executadas, os recursos necessários (pessoas, máquinas, etc), e as relações de dependência entre as atividades.

Os planos retornados devem ser consistente com os objetivos fornecidos . Algumas das soluções obtidas podem ser diferentes das soluções tradicionais - essa expectativa é baseada no fato do ser humano, na maioria dos casos, utilizar sempre a mesma solução para determinado conjunto de problemas.

Este trabalho está organizado da seguinte forma: na seção 2 são apresentados alguns conceitos sobre Gestão de Projetos; na seção 3 é definida a forma de representação das atividades da empresa; na seção 5 é descrito o algoritmo de ordem parcial adotado; na seção 6 é apresentado os resultados experimentais obtidos, e; na seção 7 são apresentadas as considerações finais deste trabalho.

## 2. Gestão de Projetos

As organizações executam trabalhos que podem ser representados pelo modelo de projetos [PMI, 2000]. Os projetos são caracterizados por serem executados por pessoas, serem restringidos por recursos limitados, e são planejados, executados e controlados.

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 em uma organização. Exemplos de 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 atender 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.

A primeira tarefa do planejamento é criar a *rede de atividades* do projeto. Nesta rede os relacionamentos entre as atividades devem refletir a seqüência de execução do trabalho<sup>1</sup>. O seu objetivo é descrever quais atividades podem ser feitas em paralelo e quais precisam ser feitas em série.

A elaboração do conjunto de atividades para um determinado projeto é realizada de duas maneiras: ou iniciando um plano totalmente novo ou utilizando algum tipo de modelo já existente.

---

<sup>1</sup>As atividades devem ser encadeadas levando-se em consideração as relações de dependência: uma termina e a outra começa (*finish-to-start*), uma começa junto com a outra (*start-to-start*), uma termina junto com a outra (*finish-to-finish*), uma começa e a outra termina(*start-to-finish*).

Existem WBS's padrões para determinados tipos de projetos, que podem servir como ponto de partida para a criação da WBS específica do projeto. Por exemplo, existem metodologias de desenvolvimento de software, como o *Unified Process*, que prevêem um conjunto padrão de atividades que deveriam existir em todos os projetos de desenvolvimento de software [Pressman, 2002].

Na maioria das vezes, com o desenrolar dos projetos, as empresas começam a manter o cronograma dos projetos já executados para futura consulta e reutilização em projetos similares.

Independente da maneira como é definido o conjunto de atividades para um determinado projeto, seja através de modelos ou iniciando um plano totalmente novo, em ambos os casos tem-se os seguintes problemas:

- i. *falta de conhecimento*: gerentes de projetos nem sempre possuem conhecimento sobre o domínio do problema ou sobre como planejar um projeto específico;
- ii. *complexidade do projeto*: muitos projetos, dado a sua complexidade, são naturalmente difíceis de serem planejados, mesmo com o auxílio de modelos que tentam retratar a experiência passada do gerente do projeto ou da empresa, e;
- iii. *falta de alternativas*: dado a forma como hoje em dia é construído um plano, o gerente de projetos tende a visualizar sempre *uma única solução* para determinado conjunto de problemas.

Na próxima seção é discutido como armazenar o conhecimento de atividades que a empresa pode executar, em uma base de conhecimento.

### 3. Representação do Conhecimento das Empresas

A primeira tarefa necessária para a construção de sistemas baseados em conhecimento é a definição de uma representação do mundo coerente com o senso comum e suficientemente precisa para permitir que a ferramenta implementada apresente um comportamento interessante tomando como base essa representação.

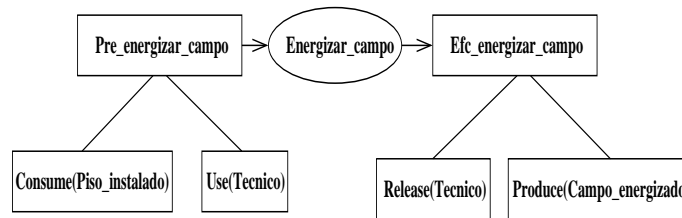
#### 3.1. STRIPS

A representação STRIPS, proposta por *Fikes e Nilsson* em [Fikes and Nilsson, 1971], tem sido amplamente utilizada na descrição de ações nos sistemas de planejamento.

Um operador STRIPS consiste de um nome, uma lista de pré-condições, uma lista de efeitos positivos (*add list*) e uma lista de efeitos negativos (*delete list*) [Fikes and Nilsson, 1971].

Não existe variável explícita do estado do mundo (situação). Tudo o que for declarado na *pré-condição* do operador se refere à situação imediatamente *antes* da ação a ser executada. E tudo o que for declarado no *efeito* se refere à situação imediatamente *depois* da ação a ser executada [Russel and Norvig, 2003].

Quando uma ação é executada ela muda a descrição do mundo. Todos os literais contidos na lista de efeitos positivos são adicionados na descrição do estado, enquanto que todos os literais contidos na lista de efeitos negativos são removidos [Lifschitz, 1990].



**Figura 1: Exemplo de conjunto de atividade e estado**

### 3.2. Atividade

Uma atividade é o objeto básico com que os processos e as operações de uma empresa podem ser representados, ou seja, é através do objeto atividade que especifica-se como o mundo é mudado para a empresa [Grüninger and Fox, 1994].

Segundo a ontologia TOVE (**TO**ronto **V**irtual **E**nterprise) o objeto atividade de empresa é definido através da relação com um objeto estado *enabled* e com um objeto estado *caused* [Enterprise Integration Laboratory, 2002, Fox and Grüninger, 1998].

Um estado *enabled* define o que deve ser verdadeiro no mundo para que a atividade possa ser executada (pré-condições). Um estado *caused* define o que é verdadeiro no mundo após a execução da atividade (efeitos).

Existem dois tipos de estados: *terminal* e *não terminal*. O estado não-terminal é definido como uma conjunção de estados terminais. Existem quatro tipos de estados terminais representados pelos seguintes predicados:  $Use(r, \alpha)$ ,  $Consume(r, \alpha)$ ,  $Release(r, \alpha)$ ,  $Produce(r, \alpha)$ . Estes predicados relatam se o recurso é *usado*, *consumido*, *liberado* ou *produzido* por uma determinada atividade, respectivamente.

O estado *enabled* é um estado não-terminal que é definido pela conjunção de estados terminais representados pelos predicados  $Use(r, \alpha)$  e  $Consume(r, \alpha)$ , enquanto que o estado não-terminal *caused* é definido pela conjunção de estados terminais representados pelos predicados  $Release(r, \alpha)$  e  $Produce(r, \alpha)$ .

O recurso é *usado* e *liberado* por uma atividade se nenhuma das propriedades do recurso são mudadas quando uma atividade é terminada com sucesso. O recurso é *consumido* ou *produzido* se alguma das propriedades do recurso é alterada depois de terminada a atividade; isto inclui a existência e quantidade de recursos, ou alguma propriedade arbitrária, como cor. Desta forma,  $Consume(r_1, \alpha_1)$  significa que o recurso  $r_1$  será utilizado pela atividade  $\alpha_1$  e depois da atividade completada não existirá mais. E  $Produce(r_2, \alpha_2)$  significa que um recurso  $r_2$ , que não existia antes da execução da atividade  $\alpha_2$  é criado após a execução da mesma.

Na figura 1  $Pre\_energizar\_campo$  é o estado não terminal que habilita a atividade  $Energizar\_campo$ , enquanto que  $Efc\_energizar\_campo$  é o estado não-terminal resultante da execução desta atividade. A conjunção de sub-estados terminais de  $Pre\_energizar\_campo$  são  $Consume(Piso\_instalado)$ , e  $Use(Tecnico)$ . A conjunção de sub-estados terminais de  $Efc\_energizar\_campo$  são  $Release(Tecnico)$  e  $Produce(Campo\_energizado)$ .

Isto significa dizer, que para executar a atividade  $Energizar\_campo$  é necessário *usar* o perfil de um *Tecnico* e *consumir* o material  $Piso\_instalado$ . Após a execução da

atividade *Energizar\_campo* é liberado o perfil de um *Tecnico* e produzido o material *Campo\_energizado*.

Na próxima seção é apresentada a forma como são codificadas as atividades de projetos na base de conhecimento. Nesta estrutura, as pré-condições, efeitos positivos e efeitos negativos da atividade são determinados com base nos predicados que definem os estados terminais de uma atividade de empresa, ou seja, com base nos seus recursos.

#### 4. Estrutura de representação de atividades

Usando a notação STRIPS e a definição de atividade segundo a ontologia TOVE, foi desenvolvida uma estrutura que define como uma atividade de projeto deve ser representada na base de conhecimento deste sistema.

A notação STRIPS define uma estrutura que permite com que algoritmos de planejamento possam manipular informações sobre ações (pré-condições, efeitos positivos e negativos), enquanto que a ontologia TOVE, fornece a estrutura de uma atividade de projeto, baseada em recursos que são usados, consumidos, liberados e produzidos.

Optou-se em utilizar a estrutura da notação STRIPS atribuindo um significado a cada elemento do operador, com base no que está descrito na ontologia TOVE de atividade. Desta forma, pode-se definir o operador que irá compor a base de conhecimento como:

$$Op = \langle N, P, A, D, E \rangle \quad (1)$$

Onde:

- i.  $N$  é o nome da atividade;
- ii.  $P$  é um conjunto de recursos, *usados* ou *consumidos*, que são pré-condições para a execução da atividade  $N$ ;
- iii.  $A$  é um conjunto de recursos *produzidos* pela atividade  $N$ ;
- iv.  $D$  é um conjunto de recursos *consumidos* pela atividade  $N$ , e;
- v.  $E$  é um número natural que determina o esforço realizado para a execução da atividade  $N$ .

Considerando  $\alpha$  como um elemento pertencente ao conjunto de atividades e  $R$  como um elemento pertencente ao conjunto de recursos, pode-se definir formalmente o significado dos elementos do operador como:

- $N = \alpha$ ;
- $P = \{R \mid use(R, \alpha) \cup release(R, \alpha) \cup consume(R, \alpha)\}$ ;
- $A = \{R \mid produce(R, \alpha)\}$ ;
- $D = \{R \mid consume(R, \alpha)\}$ ;
- $E = \{X \mid X \in N\}$

A semântica deste operador segue as regras semânticas do operador STRIPS.

Como exemplo da codificação de uma atividade na base de conhecimento, na figura 2 é possível visualizar a codificação da atividade apresentada na figura ??.

```

Op(energizar_campo,
   [tecnico,piso_instalado],
   [campo_energizado],
   [piso_instalado],
   32
  ).

```

**Figura 2: Exemplo de operador STRIPS codificado na base de conhecimento**

Onde, *energizar\_campo* é o nome da atividade, o recurso *técnico* é usado e liberado, o recurso *piso\_instalado* é consumido, o recurso *campo\_energizado* é produzido e 32 horas é o esforço necessário para a execução da atividade *energizar\_campo*.

Uma vez definida a linguagem de representação das atividades, o segundo passo é a escolha do algoritmo que irá interpretar o conhecimento existente na base. O algoritmo de planejamento escolhido é apresentado na próxima seção.

## 5. Algoritmo Planejador

Neste trabalho foi utilizado o algoritmo POP (*Partial Order Planning*) [Weld, 1994, Nguyen and Kambhampati, 2001], que é um algoritmo de planejamento regressivo de ordem parcial que realiza busca no espaço de planos.

No planejamento de ordem parcial [Weld, 1994], planos são representados por uma tupla  $\langle \omega, \beta, \gamma \rangle$ , onde:

- $\omega$  é um conjunto de *atividades do plano*;
- $\beta$  é um conjunto de *restrições de ordenação* sobre  $\omega$ :  $S_i < S_j$ , que deve ser lido “ $S_i$  ocorre antes de  $S_j$ ”, que significa:  $S_i$  deve ocorrer antes de  $S_j$  mas não necessariamente imediatamente antes, e;
- $\gamma$  é um conjunto de *vínculos causais*:  $(S_i \xrightarrow{Q} S_j)$ , que deve ser lido “*realiza Q para  $S_j$* ”. As ligações causais servem para registrar o propósito das atividades no plano, onde o propósito de  $S_i$  é realizar a pré-condição  $Q$  de  $S_j$ .

No algoritmo POP, apresentado no algoritmo 1, a idéia está em identificar uma atividade com pré-condição não satisfeita, introduzir uma atividade cujo efeito é satisfazer esta pré-condição, atualizar o conjunto de restrições de ordenação, atualizar o conjunto de vínculos causais e verificar se há ameaças e corrigir o plano se for o caso.

Uma atividade é uma *ameaça* ao plano, se quando adicionada ao plano ela interfere nas decisões passadas. Mais precisamente, suponha que  $\langle \omega, \beta, \gamma \rangle$  é um plano e  $(Ap \xrightarrow{Q} Ac)$  é uma ligação causal em  $\gamma$ . Sendo  $At$  uma atividade diferente em  $\omega$ , sabe-se que  $At$  ameaça  $(Ap \xrightarrow{Q} Ac)$  quando os seguintes critérios são satisfeitos:  $\beta \cup \{Ap < At < Ac\}$  é consistente, e;  $At$  possui  $\neg Q$  como um efeito.

O algoritmo POP é correto, completo, sistemático (sem repetição), não determinístico e a inserção de uma atividade só é considerada se atender uma pré-condição não atingida.

O plano final deve ser *completo* (toda pré-condição é realizada por alguma atividade do plano e uma pré-condição é realizada se e somente se ela é efeito de uma

---

**Algoritmo 1** Algoritmo POP( $\Sigma, \Delta, \Gamma, \langle \omega, \beta, \gamma \rangle$ ) [Weld, 1994]

---

**Entrada:**  $\Sigma$  = conjunto de atividades,  $\Delta$  = estado inicial e  $\Gamma$  = objetivos.

**Saída:** Plano parcialmente ordenado =  $\langle \omega, \beta, \gamma \rangle$ .

**1. Finalização:** se  $\Gamma$  = vazio então devolve  $\langle \omega, \beta, \gamma \rangle$ .

**2. Seleção do objetivo:** escolher  $(Q, A_{need})$ , onde  $A_{need} \in \omega$  e  $Q$  é uma pré-condição de  $A_{need}$ .

**3. Seleção da atividade:** escolher uma atividade que adiciona  $Q$  (ou uma nova atividade instanciada a partir de  $\Sigma$ , ou uma atividade já existente em  $\omega$ ); se nenhuma atividade adiciona  $Q$  então devolve *falha*;  $\gamma' = \gamma \cup \{A_{add} \xrightarrow{Q} A_{need}\}$ ;  $\beta' = \beta \cup \{A_{add} < A_{need}\}$ ; se  $A_{add}$  é uma nova atividade instanciada então  $\gamma' = \gamma \cup \{A_{add}\}$  e  $\beta' = \beta \cup \{A_i < A_{add} < A_p\}$

**4. Modificar lista de objetivos:**  $\Gamma' = \Gamma - \{(Q, A_{need})\}$ ; se  $A_{add}$  é uma nova atividade então para cada  $Q_i \in A_{add}$  adicionar  $\langle Q_i, A_{add} \rangle$  em  $\Gamma'$ .

**5. Proteção de ligações causais:** para toda ação  $A_t$  que pode ameaçar uma ligação causal  $A_p \xrightarrow{Q} A_c \in \gamma$  escolha uma ordenação consistente, entre: adicionar em  $\beta'$  :  $A_t < A_p$  (Demoção), ou; adicionar em  $\beta'$  :  $A_c < A_t$  (Promoção); se nenhuma ordenação é consistente então devolve *falha*.

**6. Invoque** POP( $\Sigma, \Delta, \Gamma', \langle \omega', \beta', \gamma' \rangle$ ).

---

atividade e nenhuma atividade intermediário a desfaz) e *consistente* (não há contradições nas ordenação das atividades) [Russel and Norvig, 2003, Weld, 1994].

### 5.1. Conversão de um plano parcialmente ordenado em um modelo de rede de atividades

Para transformar as informações de um plano parcialmente ordenado em um modelo de rede de atividades, deve-se aplicar um algoritmo de conversão que tem como entrada um plano parcialmente ordenado (passos do plano, restrições de ordenação e vínculos causais) e como saída uma rede de atividades (conjunto de atividades, conjunto de recursos, relação de seqüência entre as atividades e conjunto de números naturais representando a duração das atividades).

A idéia do algoritmo de conversão está em listar, para todas as atividades pertencentes ao conjunto de passos do plano ( $\omega$ ), seus respectivos predecessores através das informações contidas no conjunto de restrições de ordenação ( $\gamma$ ), no formato  $S_i < S_j$ , e não atribuir nenhum predecessor para cada elemento em  $S_j$  que não estiver em  $S_i$ . Após listada todas as atividades e seus respectivos predecessores o algoritmo realiza uma consulta na base de conhecimento para estabelecer qual a duração e respectivos recursos utilizados por cada atividade.

Com o objetivo de ilustrar a funcionalidade desta arquitetura, na seção seguinte é descrito um exemplo, juntamente com os resultados experimentais obtidos.

## 6. Resultados Experimentais

Considere um projeto onde o objetivo é construir uma estação rádio base. Para isso é dada a descrição de uma empresa orientada a projetos, com estrutura matricial, que possui um conjunto de equipes - e.g. pré-montagem (ADM), desenvolvimento (DEV), implantação (IMP), manutenção (MAT) e serviços auxiliares (AUX) - e cujo o objetivo é desenvolver

**Tabela 1: Descrição das atividades**

Atividade	Recursos			Duração
	Usados e Liberados	Consumidos	Produzidos	
procurar_area2	ADM		area_ok	2 UT
pedido_prefeitura2	ADM area_ok		pedido_ok	1 UT
fazer_testes_fabrica3	DEV		eq_testado	3 UT
colocar_material_campo1	AUX pedido_ok eq_testado	eq_testado	material_campo	1 UT
fazer_marcacao_piso2	IMP		piso_marcado	2 UT
fazer_instalacao_piso4	material_campo piso_marcado IMP	piso_marcado	piso_instalado	4 UT
energizar_campo4	piso_instalado IMP		energia_pronta	4 UT
fazer_marcacao_piso5	AUX		piso_marcado	5 UT
contratar_montagem20	prestador_servico ADM		radio_base_montada	20 UT

produtos na área de telecomunicações - e.g construção de estações rádio base, desenvolvimento de celulares, entre outros. O estado inicial é representado pelo conjunto de todas as equipes da empresa  $\{ADM, DEV, IMP, MAT, AUX\}$  e o conjunto de atividades  $\Sigma$  é composto, entre todas as outras atividades da empresa, pelas atividades descritas na tabela 1.

Dado a entrada  $(\Sigma, \{ADM, DEV, IMP, MAT, AUX\}, radio\_base\_montada)$ , o algoritmo de planejamento deve devolver os planos parcialmente ordenados da figura 3. Onde, as caixas são as atividades obtidas através de  $\omega$ , as setas indicam as relações de precedência obtidas através de  $\beta$  e os nomes ao lado das caixas indicam os recursos utilizados.

A diferença entre os planos (1) e (2) da figura 3 é que no plano (1) a atividade *fazer\_marcacao\_piso* possui uma duração de 2 unidades de tempo, enquanto que no plano (2) a mesma atividade possui uma duração de 5 unidades de tempo, isto porque a segunda atividade utiliza um grupo de recursos menos especializado. O plano número (3) demonstra que uma das opções da empresa para alcançar o objetivo determinado, além da própria empresa desenvolver o projeto, é terceirizar o projeto.

Aplicando o algoritmo de conversão sobre o plano (2) da figura 3 é possível gerar o modelo de rede de atividade da figura 4.

Para a implementação do arcabouço descrito neste artigo foi utilizado a linguagem de programação Prolog na implementação do algoritmo POP e a linguagem de programação Java para implementar a interface com o usuário. Para comunicar os módulos implementados em Java com o módulo implementado em Prolog foi utilizado a API JPL.

O resultado final é convertido para o formato de uma das ferramentas mais po-



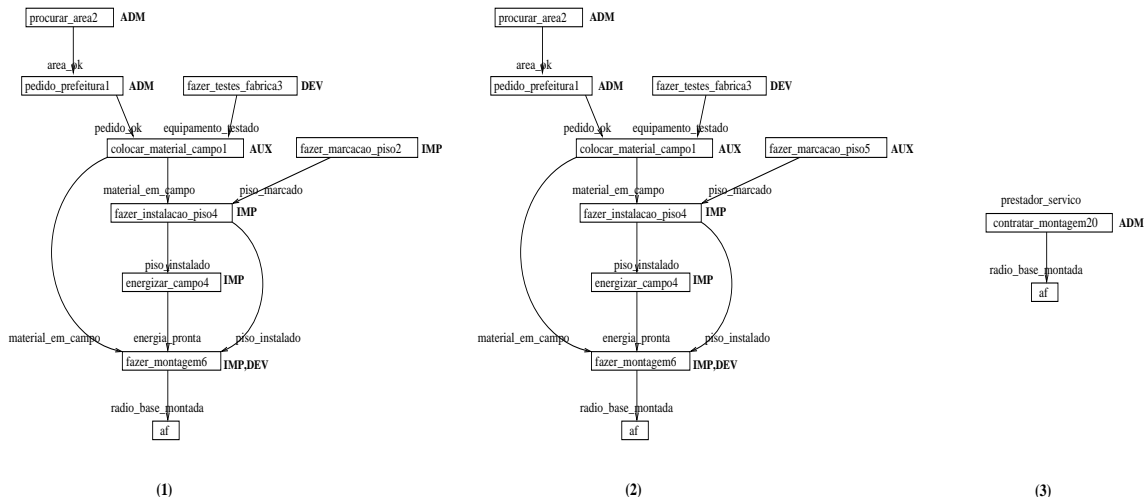


Figura 3: Planos parcialmente ordenados

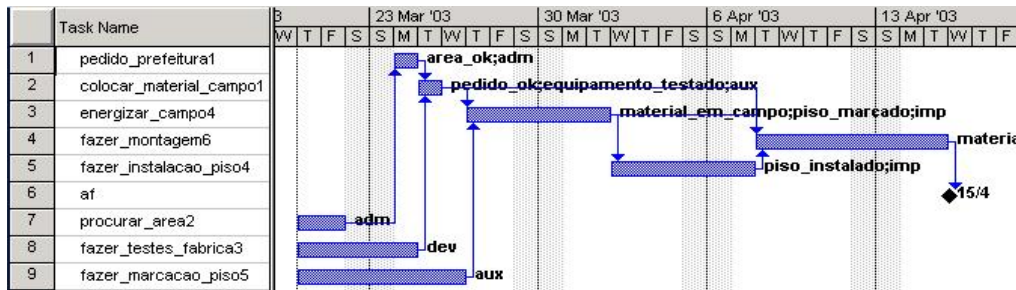


Figura 4: Rede de atividade do plano (2) da figura 3

pulares para gestão de projetos, o *Microsoft Project*, a fim de, fazer com que o gerente de projetos possa utilizar o plano proposto pela ferramenta para fazer suas modificações sobre o mesmo.

Os testes realizados foram alimentados com informações a partir de um conjunto de projetos já executados por uma empresa de telecomunicações. No total foram 60 projetos, média de 40 atividades por projeto, inseridos na base de conhecimento.

## 7. Considerações Finais

A comparação dos resultados obtidos pelo sistema desenvolvido com as redes de atividades propostas pelos gerentes de projetos, mostra que os resultados obtidos são coerentes com os projetos reais. Os resultados obtidos utilizando esta abordagem foram melhores do que os resultados utilizando a abordagem do cálculo de situações descrita em [Barth and Gomi, 2002a, Barth and Gomi, 2002b].

Os planos retornados apresentam todas as informações solicitadas pelos requisitos do sistema: o conjunto de atividades que deverá ser executada para desenvolver o produto, os recursos necessários para a execução das atividades, a sequência das atividades e a duração das atividades.

Também foi possível testar o uso de ontologias na reutilização do conhecimento para a construção de novos sistemas baseados em conhecimento. Neste trabalho, as

definições de atividade, recursos, pré-condições e efeitos da ontologia TOVE foram utilizadas para atribuir significado aos objetos que estão representados nos operadores da base de conhecimento.

Durante o desenvolvimento e a avaliação do sistema, constatou-se a importância do processo de aquisição de conhecimento, seja na forma manual ou automática. Se a aquisição de conhecimento for realizada de maneira correta, o sistema irá propor um ou vários planos corretos para um determinado problema, caso contrário, o sistema não apresentará soluções úteis. É por isso que, uma das sugestões para trabalhos futuros é o aperfeiçoamento do sistema adicionando módulos de aquisição automática sobre as informações de atividades de projetos. Um módulo de aquisição automática de conhecimento pode reduzir as chances de erro durante o processo de aquisição de conhecimento.

De qualquer forma, acredita-se que o sistema desenvolvido neste trabalho, possa servir como uma ferramenta para que uma empresa possa estruturar o conhecimento sobre projetos já executados, disponibilizando-o aos membros da organização. Isso permite minimizar alguns dos problemas relacionados a tarefa de planejamento na área de gestão de projetos: falta de conhecimento para realizar a tarefa de planejamento, complexidade inerente dos projetos e falta de soluções alternativas.

Além da sugestão para trabalho futuro já enumerada anteriormente, outras sugestões são: (i) desenvolver uma interface com o usuário mais amigável, por exemplo, utilizando uma interface gráfica com diagramas equivalentes ao da ontologia TOVE; (ii) avaliar a utilidade de algoritmos hierárquicos neste problema. e; (iii) utilizar uma abordagem que una técnicas de planejamento com técnicas para solucionar problemas de escalonamento, como descrito por [Laborie, 2003].

## Referências

- Barth, F. J. and Gomi, E. S. (2002a). An extension of situation calculus applied to project management. In *4th Argentine Symposium on Artificial Intelligence*, volume 4, pages 231–241, Santa Fé, Argentina.
- Barth, F. J. and Gomi, E. S. (2002b). Extensão do cálculo de situações aplicada à gestão de projetos. In Abe, J. M. and Filho, J. I. S., editors, *Advances in Logic, Artificial Intelligence and Robotics. Proceedings of the Congress of Logic Applied to Technology - LAPTEC'2002*, volume II, pages 139–146.
- Enterprise Integration Laboratory (2002). *TOVE Ontology Project*. Enterprise Integration Laboratory, <http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html>.
- Fikes, R. and Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2((3/4)):189–208.
- Fox, M. S. and Grüninger, M. (1998). Enterprise modelling. *AI Magazine*, pages 109–121.
- Grüninger, M. and Fox, M. S. (1994). An activity ontology for enterprise modelling. In *Workshop on Enabling Technologies - Infrastructures for Collaborative Enterprises*, West Virginia University.

- Laborie, P. (2003). Algorithms for propagating resource constraints in ai planning and scheduling: Existing approaches and new results. *Artificial Intelligence*, (143):151–188.
- Lifschitz, V. (1990). On the semantics of strips. In Allen, J., Hendler, J., and Tate, A., editors, *Readings in Planning*, pages 523–531. Morgan Kaufman.
- Nguyen, X. and Kambhampati, S. (2001). Reviving partial order planning. In *Proc. IJCAI-01*, pages 459–464, Seattle, WA.
- O’Leary, D. E. (1998). Enterprise knowledge management. *IEEE Computer*, pages 54–61.
- PMI (2000). *A Guide to the Project Management Body of Knowledge*. PMI, Maryland, USA.
- Pressman, R. S. (2002). *Engenharia de Software*. McGraw-Hill, 5 edition.
- Russel, S. J. and Norvig, P. (2003). *Artificial intelligence: a modern approach*. Prentice-Hall, 2 edition.
- Weld, D. S. (1994). An introduction to least commitment planning. *AI Magazine*, 15(4):27–61.

## **Agradecimentos**

Este trabalho foi financiado pela Fundação para o Desenvolvimento Tecnológico da Engenharia (FDTE), através de recursos provenientes do projeto *Desenvolvimento de um Sistema Integrado de Gestão de Projetos - SIGP*, que foi realizado em parceria com a Ericsson Telecomunicações S/A. dentro do âmbito da Lei de Informática/Ministério da Ciência e Tecnologia.