

ZENO STIVANIN

**TRAÇADO AUTOMÁTICO DE HIPERGRAFOS
DIRECIONADOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. André Luiz Pires Guedes

CURITIBA

2006

ZENO STIVANIN

**TRAÇADO AUTOMÁTICO DE HIPERGRAFOS
DIRECIONADOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. André Luiz Pires Guedes

CURITIBA

2006

ZENO STIVANIN

**TRAÇADO AUTOMÁTICO DE HIPERGRAFOS
DIRECIONADOS**

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Informática da Universidade Federal do Paraná, pela Comissão formada pelos professores:

Orientador: Prof. Dr. André Luiz Pires Guedes
Departamento de Informática, UFPR

Profa. Dra. Lilian Markenzon
Núcleo de Computação Eletrônica, UFRJ

Prof. Dr. Renato José da Silva Carmo
Departamento de Informática, UFPR

Curitiba, 10 de março de 2006

AGRADECIMENTOS

Desejo agradecer a todas as pessoas que, direta ou indiretamente, contribuíram para a realização desta Tese de Mestrado, especialmente ao meu orientador Professor Doutor André Luiz Pires Guedes.

Dedico este trabalho a minha família por tudo que ela representa e contribui infinitamente na minha vida e a Andreia Aparecida Barbiero por todo seu apoio, incentivo, inspiração, compreensão e verdadeiros puxões de orelha que me colocaram no caminho certo para finalizar este trabalho.

A todos vocês Obrigado.

SUMÁRIO

LISTA DE FIGURAS	v
RESUMO	vi
ABSTRACT	vii
1 INTRODUÇÃO	1
1.1 Objetivo	2
2 CONCEITOS FUNDAMENTAIS	4
2.1 Grafo e Grafo Direcionado	4
2.2 Hipergrafo e Hipergrafo Direcionado	5
2.3 Desenho de Grafos	8
2.3.1 Introdução	8
2.3.2 Padrão de Desenho	9
2.3.3 Características Estéticas	11
2.3.4 Algoritmos de Desenho de Grafos	12
3 HIPERGRAFOS DIRECIONADOS PARA GERÊNCIA DE PROJE-	
TOS	14
3.1 Representações Gráficas de Atividades em Gerência de Projetos	14
3.1.1 Método do diagrama de precedência (ADM)	15
3.1.2 Método do diagrama de flecha/arco (ADM)	16
3.1.3 Método do diagrama condicional (CDM)	16
3.2 Representações de Atividades em Gerência de Projetos Mapeadas em Hi-	
pergrafos Direcionados	17
3.3 Exemplo de Uso de Hipergrafos Direcionados para Representação das Ati-	
vidades de um Projeto	18

3.4	Abrangência da Representação	20
4	ESTUDO DA FORMA MAIS ADEQUADA DE TRAÇADO DE HIPERGRAFOS DIRECIONADOS PARA GERÊNCIA DE PROJETOS	24
4.1	Escolha do Algoritmo de Traçado e Padrão de Desenho	24
4.2	Transformação de Hipergrafos Direcionados em Grafos Direcionados	25
4.3	Testes de Traçado de Hipergrafos Direcionados	26
5	FERRAMENTA DE TRAÇADO AUTOMÁTICO DE HIPERGRAFOS DIRECIONADOS	32
5.1	Desenvolvimento da Ferramenta	32
5.1.1	Passos da Implementação da Ferramenta	33
5.2	Resultados Obtidos	36
5.3	Representação de Ciclos	37
6	CONCLUSÃO	39
6.1	Sugestões de Trabalhos Futuros	40
	BIBLIOGRAFIA	42

LISTA DE FIGURAS

2.1	Exemplos de representações de Grafos	5
2.2	Exemplo de Grafo Direcionado	5
2.3	Exemplo de Hipergrafo	6
2.4	Exemplo de Hipergrafo Direcionado	6
2.5	Desenho Poligonal	9
2.6	Desenho Linha Reta	10
2.7	Desenho Ortogonal	11
2.8	Desenho de Árvore utilizando padrão Linha Reta	11
2.9	Dois desenhos de um grafo de cubo	12
3.1	Exemplo de Diagrama de Dependências	16
3.2	Exemplo de Diagrama de Flecha	17
3.3	Exemplo de Diagrama Condicional	17
3.4	Exemplo de Representação de Projeto no Software Girassol	19
3.5	Exemplo de Projeto Visto pelo Software Girassol(Parte 1 de 2)	20
3.6	Exemplo de Projeto Visto pelo Software Girassol(Parte 2 de 2)	21
3.7	Tabela com os nomes dos Objetos	22
3.8	Exemplo de Arquivo Texto com Representação de um Hipergrafo Direcionado	22
3.9	Hipergrafo Direcionado traçado a partir de exemplo de projeto do software Girassol	23
4.1	Exemplo de Transformação de Hipergrafo Direcionado Para Grafo Direci- onado	26
4.2	Exemplos de Transformação de Hipergrafo Direcionado Para Grafo Direci- onado	27
4.3	Exemplo do Estudo Preliminar de Traçado	28
4.4	Traçado Automático de Hipergrafo do Girassol - Algoritmo Simetria	29

4.5	Traçado Automático de Hipergrafo do Girassol - Algoritmo Ortogonal . . .	30
4.6	Traçado Automático de Hipergrafo do Girassol - Algoritmo Hierarquico . . .	31
5.1	Exemplo da Tela Inicial do Software VGJ	33
5.2	Exemplo da Tela Inicial do Software VGJ com Alteração Para Leitura de Hipergrafos Direcionados	34
5.3	Padrão de Representação de Hiper-aresta Criado	35
5.4	Exemplos de Representação de Hiper-arestas	35
5.5	Resumo Gráfico das Modificações	36
5.6	Resultado Obtido com as Alterações do VGJ	38

RESUMO

Promover uma distribuição automática dos vértices e arestas de um grafo de maneira automática seguindo um conjunto de características estéticas pré-estabelecidas define o objetivo de um algoritmo de desenho de grafos. Esta área da computação tem crescido em importância nos últimos anos devido a sua aplicabilidade nas mais diversas áreas. Um hipergrafo direcionado é uma generalização de grafos que tem como característica principal a capacidade de representar paralelismo. O traçado automático para hipergrafos direcionados é uma área ainda pouco explorada. Desta forma o objetivo deste trabalho é, a partir de um estudo preliminar, propor um algoritmo de traçado automático para hipergrafos direcionados especialmente focados na representação de relacionamentos de precedência de atividades em gerência de projetos.

ABSTRACT

To promote an automatic distribution of vertices and edges of a graph following a set of established aesthetic characteristics defines the objective of a graph drawing algorithm. This subject has grown in importance in the last years due to its applicability in the most diverse areas. A directed hypergraph is a generalization of graphs that has as main characteristic the capacity to represent parallelism and flow. The automatic drawing of a directed hypergraph is a little explored field. The objective of this work is to consider an algorithm of automatic tracing for directed hypergraph representing relationships of activities in management of projects.

CAPÍTULO 1

INTRODUÇÃO

O estudo de técnicas de traçado automático de grafos é uma área de pesquisa que tem crescido de maneira significativa nos últimos anos devido a sua importância e abrangência. Há diversas aplicações para desenho automático de grafos tais como:

- Representação gráfica de navegação na internet.
- Representação de atividades em gerência de projetos.
- Desenhos de árvores nas mais variadas situações.
- Circuitos integrados.
- Microprocessadores.
- Atividades de processamento paralelo, entre outras.

Desenhar grafos de forma automática não é tarefa trivial. Os critérios que definem se um desenho de grafo é de boa qualidade são por demais subjetivos e caracterizá-lo entra em méritos até mesmo culturais. Para avaliar um desenho de grafo é necessário levar em conta as características desejadas para a representação formada, por exemplo, para desenhar placas de circuitos é necessário que o desenho seja formado somente por linhas retas e preferencialmente formando ângulos retos. Já a representação de uma árvore genealógica não ficaria bem representada utilizando-se esta configuração. Aplicações diferentes necessitam de características diferentes o que acarreta uma grande variedade de padrões de desenho como: padrão ortogonal, padrão árvore, circular, entre outros.

Outro ponto a ser observado é que, devido a grande variedade de tipos de grafos tais como, grafos direcionados, não direcionados, planares, não planares, sem citar os de características mais específicas, o desenvolvimento de um algoritmo de traçado que seja capaz de gerar desenhos com qualidade para qualquer tipo de grafo de entrada é

uma tarefa muito difícil e não muito utilizada. Por isso prioriza-se o desenvolvimento de algoritmos especialistas focados em aplicações específicas de modo a possibilitar uma melhor exploração das características agregadas aumentando a qualidade do desenho.

O hipergrafo direcionado, objeto de estudo deste trabalho, é uma generalização do conceito de grafo com características particulares utilizado principalmente na representação de paralelismo e fluxo [10] [1] [9]. Este tem importância significativa em áreas como:

- Arquitetura paralela onde é utilizado na representação de processamento paralelo.
- No gerenciamento de projetos, sendo utilizado para representar o relacionamento entre os serviços e ou atividades a serem realizadas no decorrer do mesmo.
- Na arquitetura de software, na representação gráfica de diagramas UML entre outras.

1.1 Objetivo

Serão comparados os traçados dos algoritmos por simetria, ortogonal e hierárquico aplicados a hipergrafos direcionados especialmente focados na aplicação de representação das dependências entre as atividades de hipergrafos direcionados retirados de gerência de projetos.

Desta forma será mostrado que o algoritmo de traçado hierárquico é o mais indicado para o traçado de hipergrafos gerenciais devido a características estéticas particulares e especialmente a capacidade de representação de fluxos.

Para atingir o objetivo será utilizada a transformação de hipergrafos direcionados em grafos direcionados [10] [11]. Com o resultado desta, torna-se possível a utilização de ferramentas de traçado de grafos aproveitando-se assim de algoritmos previamente implementados. Utilizando esta abordagem serão comparados os traçados dos três algoritmos escolhidos de forma a mostrar a superioridade do uso do algoritmo hierárquico.

Após este estudo será feita a adaptação de uma ferramenta de traçado de grafos de modo a prepará-la ao traçado automático de hipergrafos direcionados concluindo assim este trabalho.

A organização do trabalho é: no Capítulo 1 a apresentação do trabalho bem como o objetivo; no Capítulo 2 a apresentação dos conceitos fundamentais da teoria de grafos e desenho de grafos necessários para a compreensão do texto; no Capítulo 3 um resumo sobre os três algoritmos de traçado utilizados no trabalho; no Capítulo 4 a apresentação da teoria de gerência de projetos com as formas de representação das atividades e o exemplo prático do uso de hipergrafos direcionados para a representação de atividades em gerência de projetos; no Capítulo 5 a apresentação do estudo da forma mais adequada de traçado de hipergrafos direcionados para gerência de projetos; no Capítulo 6 o desenvolvimento da ferramenta de traçado automático de hipergrafos direcionados; no Capítulo 7 a conclusão do trabalho.

CAPÍTULO 2

CONCEITOS FUNDAMENTAIS

As seções abaixo trazem definições básicas sobre grafos, hipergrafos, desenho de grafos e algoritmos de desenho de grafos necessárias para o desenvolvimento do trabalho.

2.1 Grafo e Grafo Direcionado

Um grafo $G = (V, A)$ é um conjunto finito não vazio V (conjunto de vértices) e um conjunto A (conjunto de arestas) de pares não-ordenados de elementos distintos de V . A é um subconjunto de V com exatamente dois elementos.

Há diversas formas de representar um grafo, formas textuais e gráficas. Existem várias maneiras diferentes de se representar um grafo de maneira textual, não existe uma padronização adotada neste sentido, mas há maneiras consagradas, como por exemplo, a formatação de um simples arquivo texto onde cada linha é utilizada para representar uma aresta. Nomes são utilizados para representar os vértices e vírgulas para representar a ligação entre eles, a aresta. Outra maneira é representar a matriz de adjacência em arquivos texto. Uma matriz de adjacência apresenta uma linha e uma coluna para cada vértice do grafo, sendo os valores das arestas armazenados na matriz, conforme ilustrado na Figura 2.1. Nesta figura, os zeros representam as arestas inexistentes (qualquer outro caracter poderia ser usado para marcar as arestas inexistentes).

Uma maneira simples e uma das mais utilizadas para representar um grafo graficamente é fazer uso de círculos como padrão de representação dos vértices e utilizar linhas para a representação padrão das arestas. A Figura 2.1 ilustra as duas situações, a representação textual, tanto para o modo simples de arquivo texto quanto para a matriz de adjacência e a representação gráfica do mesmo grafo.

Um grafo direcionado $D = (V, A)$ é um par onde V é um conjunto finito de vértices e A é um conjunto finito de arcos, onde um arco " a " $\in A$ é um par ordenado de elementos

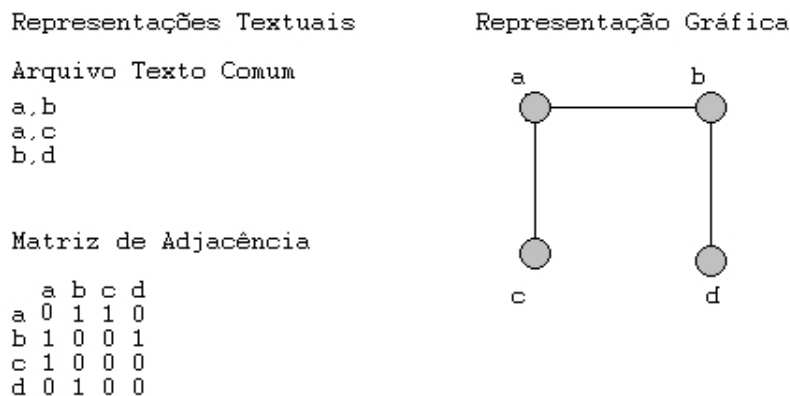


Figura 2.1: Exemplos de representações de Grafos

de V , ou seja, " a " \in $a \times V$. Observe a Figura 2.2 que mostra um exemplo simples de um grafo direcionado representado graficamente.

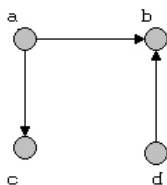


Figura 2.2: Exemplo de Grafo Direcionado

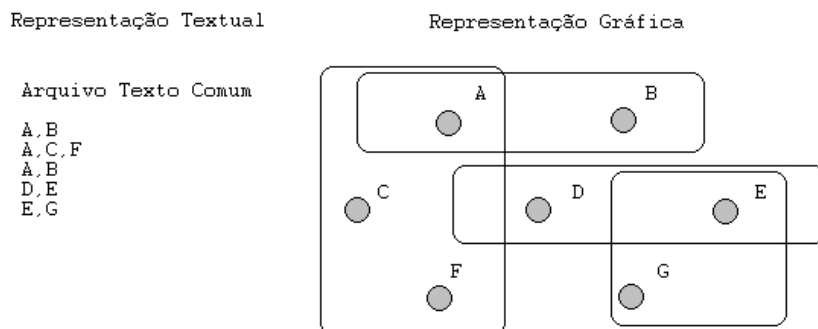
2.2 Hipergrafo e Hipergrafo Direcionado

Segundo [10], [1], [9] tem-se as seguintes definições para hipergrafo e hipergrafo direcionado.

O hipergrafo é uma generalização do conceito de grafos. Em um hipergrafo a cardinalidade das arestas pode ser diferente de dois. Um hipergrafo $H = (V, A)$ onde V é um conjunto finito de vértices e A é um conjunto finito de hiper-arestas, onde uma hiper-aresta $a \in A$ é um subconjunto não-vazio de V .

Na Figura 2.3 tem-se um exemplo de hipergrafo. Sua representação textual, um arquivo texto comum e sua representação gráfica segundo indicado por David Harel em [12].

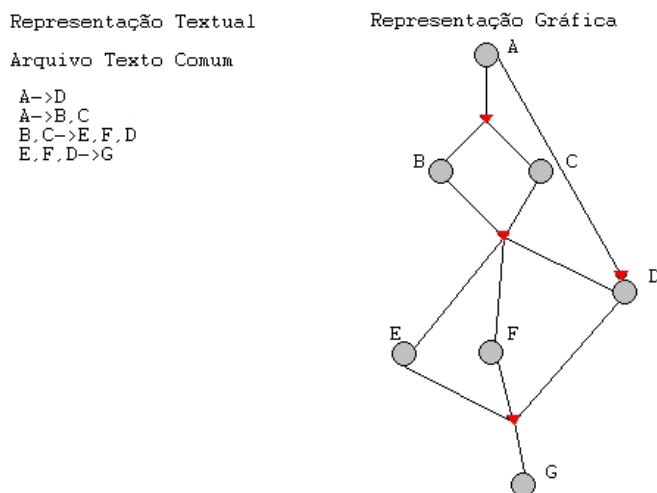
Tendo que um hipergrafo direcionado é uma generalização do conceito de grafos direci-



onados, e uma especialização de hipergrafos, pode-se visualizar um hipergrafo direcionado como um conjunto de vértices e um conjunto de hiper-arestas, onde as hiper-arestas são pares de conjuntos disjuntos de vértices (X, Y) .

Ou seja, um hipergrafo direcionado $H = (V, A)$ é um par onde V é um conjunto finito de vértices e A é um conjunto finito de hiper-arestas, onde uma hiper-aresta "a" $\in A$ é um par ordenado (X, Y) , onde X e Y são subconjuntos de elementos de V .

A Figura 2.4 traz um exemplo de hipergrafo direcionado, sua representação textual, arquivo texto comum e sua representação gráfica.



Segundo [10] o uso de hipergrafos direcionados se apresenta como uma alternativa para a modelagem de problemas em que relações binárias usuais não são adequadas. Segue um resumo de algumas aplicações conhecidas na literatura.

Em banco de dados relacionais tem-se o conceito de dependências funcionais, que são as dependências entre atributos de dados, dependências estas que são relações entre subconjuntos de um conjunto de atributos, ou seja são pares (X,Y) onde X e Y são conjuntos de atributos, indicando que Y depende de X (denotado por $X \rightarrow Y$). Neste caso os hipergrafos direcionados podem ser usados como um modelo de representação destas dependências funcionais.

O problema a ser resolvido neste caso é, dado um conjunto de dependências funcionais, encontrar um caminho mínimo que representa as mesmas dependências.

Na área da lógica tem-se o cálculo proposicional e as variações do problema de satisfação (SAT). Neste tipo de problema, temos um conjunto de cláusulas desta forma

$$C = P_1 \vee P_2 \vee \dots \vee P_r \wedge \neg P_{r+1} \wedge \dots \wedge P_q,$$

onde, para $i=1,\dots,q$, P_i é um literal, que pode ser verdadeiro ou falso. O significado de C é que pelo menos um literal P_1,\dots,P_r deve ser verdadeiro quando todos os literais P_{r+1},\dots,P_q o forem. Se isso for verdade, então a cláusula C é verdadeira, caso contrário é falsa.

Nesta aplicação tem-se que cada cláusula C pode ser representada por uma hiper-aresta $(\{P_{r+1},\dots,P_q\},\{P_1,\dots,P_r\})$ em um hipergrafo direcionado onde os vértices são os literais mais os símbolos t e f (respectivamente verdadeiro e falso).

Um outro exemplo desta aplicação é a modelagem de sistemas especialistas baseados em regras, onde as regras que representam o conhecimento do sistema podem ser modeladas como hiper-arestas de um hipergrafo direcionado (como cláusulas). Desta forma, verificações de consistência e completude podem ser feitas mais facilmente.

Ainda em lógica, tem-se os grafos E/OU que podem, entre outras coisas, são usados na resolução de problemas através da redução por subproblemas. Desta vez aparecem as relações entre um grupo de problemas (entrada) que podem ser resolvidos por um conjunto de subproblemas (saída). E mais uma vez pode-se modelar a situação com hiper-arestas, desta vez na forma (i,X) .

Na área de programação paralela, tem-se algumas aplicações, entre elas o estudo do

comportamento de programas paralelos, onde as interdependências entre as tarefas podem ser modeladas por hiper-arestas, nos mesmos moldes de uma rede de Petri.

Todas estas aplicações aqui apresentadas estão relacionadas com conceitos de caminhos (hiper-caminhos). Algumas buscam caminhos mínimos em relação a alguma grandeza, outras buscam apenas encontrar um caminho, ou todos os caminhos, e algumas usam fechos transitivos (de caminhos). Desta forma são utilizações de hipergrafos direcionados que dispensam representações gráficas e requerem processamento para a solução dos problemas a que se destinam a resolver.

Na área de gerência de projetos analisando um projeto com seus eventos e relacionamentos entre eles, tomando os eventos como vértices e seus relacionamentos como hiper-arestas tem-se a representação de um projeto na forma de um hipergrafo direcionado. Esta aplicação, sim, necessita de uma representação gráfica de modo a transmitir para o administrador, gerente, profissional que estará fazendo uso do diagrama, as informações sobre o fluxo armazenado nas atividades gerenciadas.

As informações de fluxo contidas nas hiper-arestas são de grande importância para a análise e percepção de possíveis gargalos contidos no fluxo do projeto. Encontrar tais gargalos pode ser uma tarefa automatizada por meio de um algoritmo. Tendo esta classificação de gargalo seria possível informar ao analista sobre os mesmos, mas esta informação não estaria completa sem o uso da representação gráfica. O profissional de posse desta representação gráfica torna-se mais consciente dos problemas que poderão ocorrer devido aos gargalos do fluxo do projeto podendo tomar ações gerenciais preventivas.

2.3 Desenho de Grafos

2.3.1 Introdução

O estudo do desenho de grafos endereça o problema de construir representações geométricas de grafos abstratos [8]. Esta é uma área de pesquisa que combina teoria de topologia de grafos com geometria computacional. A geração automática de grafos possui importantes aplicações em áreas chave da ciência da computação tais como engenharia de software,

desenho de bancos de dados, interfaces visuais, entre outras.

Pesquisas sobre desenho de grafos têm sido especialmente estudadas na última década. A bibliografia anotada sobre algoritmos de desenho de grafos de G. Di Battista, P. Eades, R. Tamassia e I.G. Tollis [2] lista mais de 300 artigos. O 13 Simpósio Internacional de Desenho de Grafos de 2005 [5], reuniu 45 trabalhos publicados com autores de diversas partes do mundo.

2.3.2 Padrão de Desenho

Define-se Padrão de Desenho como sendo um conjunto de características que determinam como os vértices e as arestas são dispostos no espaço.

Usualmente os vértices são representados por pontos ou círculos e cada aresta (u,v) é representada por uma simples linha ligando os pontos associados aos vértices u e v [2].

Vários padrões de desenho de grafos têm sido propostos para a representação de um grafo no plano. Segue um resumo dos principais padrões conhecidos na literatura.

Um desenho em que cada aresta é representada por um conjunto de segmentos de retas dispostas de maneira poligonal é chamado de poligonal (Figura 2.5) [2]. Note que o padrão poligonal pode ser modificado de modo a prover desenhos de arestas com curvas.

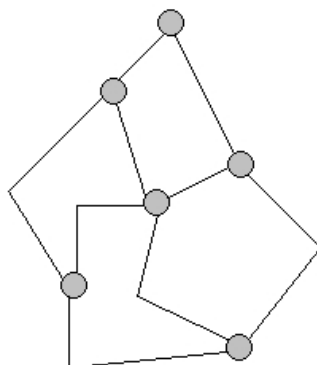


Figura 2.5: Desenho Poligonal

Existem dois casos especiais para o padrão poligonal, o linha reta e o ortogonal.

O padrão linha reta (Figura 2.6), dita que cada aresta deve ser representada como um simples segmento de reta e os vértices como círculos ou outro objeto qualquer. Em textos

sobre teoria de grafos esta é a representação mais comumente utilizada.

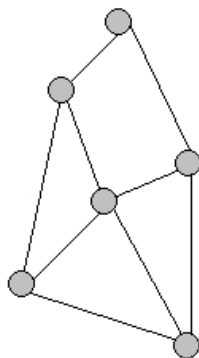


Figura 2.6: Desenho Linha Reta

Já o padrão ortogonal mapeia cada aresta em uma cadeia de segmentos verticais e horizontais (Figura 2.7). Desenhos do tipo ortogonal são comuns em várias áreas, mas são especialmente utilizadas em aplicações nas áreas técnicas e de engenharia onde sua característica de clareza é desejada. Diagramas entidade relacionamento no projeto de bases de dados são geralmente desenhados de acordo com este padrão de desenho.

Há o padrão grade (*grid*) que é uma variação do padrão ortogonal onde os vértices e os cotovelos (*bends*) das arestas possuem coordenadas inteiras. Este padrão é encontrado no desenho de hardware de computadores e microprocessadores que por serem muito complexos é necessário para desenhá-los o uso de ferramentas CAD de maneira que a responsabilidade de criar um layout das portas lógicas e das conexões entre os microprocessadores e as placas de circuitos ficam a cargo da ferramenta.

A representação de grafos em formato de árvores tem importância tão acentuada que possui um padrão próprio para este fim. Este é utilizado frequentemente para representar estruturas hierárquicas como árvores genealógicas, gráficos organizacionais de empresas e árvores de busca. Padrões Linha Reta e Ortogonais tem sido comumente utilizados para representar árvores (Figura 2.8).

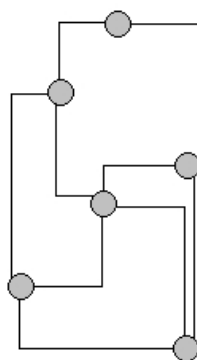


Figura 2.7: Desenho Ortogonal

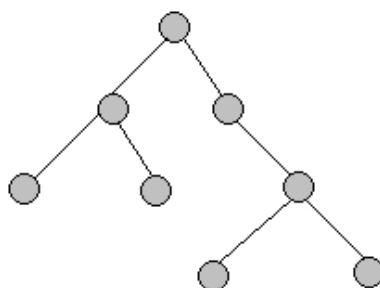


Figura 2.8: Desenho de Árvore utilizando padrão Linha Reta

2.3.3 Características Estéticas

Dentro de um padrão de desenho de grafos, é possível desenhar um mesmo grafo de diversas maneiras, entretanto, em quase todas as aplicações de apresentação de dados o grau de utilidade de um desenho de grafo depende da capacidade de lê-lo, interpretá-lo, ou seja, a capacidade de extrair seu significado de maneira rápida e clara. O grau de facilidade de leitura e interpretação de um grafo é expresso em termos estéticos que podem ser descritos como objetivos de otimização para os algoritmos de desenho. Em termos gerais a estética depende do padrão de desenho de grafo adotado e da classe de algoritmo a ser utilizado.

Um objetivo estético clássico e fundamental é a minimização dos cruzamentos entre

as arestas. No modelo poligonal é desejável evitar cotovelos nas arestas. Em todos os modelos de desenho uma visualização com características de simetria é desejável. Outros padrões estéticos desejáveis são: complexidade visual, regularidade, consistência, forma, tamanho e proporção.

Deve-se notar que estas preocupações estéticas são subjetivas e devem ser toleradas de modo a admitir preferências pessoais, tradicionais e culturais. Por exemplo, embora a representação de um cubo seja planar, tradicionalmente ele é desenhado com cruzamentos de suas arestas, como se observa na Figura 2.9.

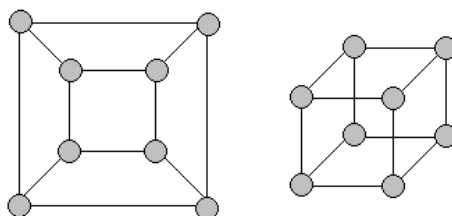


Figura 2.9: Dois desenhos de um grafo de cubo

2.3.4 Algoritmos de Desenho de Grafos

Neste trabalho são utilizados três algoritmos de desenho para o padrão poligonal contemplando os dois casos especiais citados, o linha reta e ortogonal.

Os algoritmos utilizados são: simetria, ortogonal e hierárquico.

O algoritmo de simetria é descrito por [7]. Neste método de desenho de grafos o escopo de entrada são os grafos não direcionados e o desenho resultante tem duas dimensões. As características estéticas a serem alcançadas com o uso deste método são: uniformidade, minimização de cruzamentos, uniformização dos tamanhos das arestas e simetria. A característica estética mais marcante neste método é a simetria.

A idéia básica deste algoritmo é que cada vértice é visto como um anel e cada aresta como uma mola caracterizando um sistema mecânico. Os vértices são colocados em um estado inicial aleatório e deixa-se que as forças das molas movam o sistema para um estado de mínima energia.

O segundo algoritmo desenha um grafo automaticamente de forma ortogonal. Tamassia e Tollis [16] descrevem um algoritmo composto de quatro passos que são:

- Construir uma representação de visibilidade.
- Transformar/criar uma representação ortogonal.
- Redução de cotovelos.
- Computar um *grid* ortogonal para a representação ortogonal.

As características estéticas a serem alcançadas com o uso deste método são muito parecidas com a do primeiro algoritmo apresentado: uniformidade, minimização de cruzamentos, uniformização dos tamanhos das arestas e simetria. Diferente do primeiro algoritmo, a característica estética mais marcante neste método não é a simetria e sim a uniformização.

O terceiro algoritmo produz uma representação hierárquica do grafo de entrada [13]. As características estéticas a serem alcançadas com o uso deste método são muito parecidas com a do primeiro e segundo algoritmo apresentado: uniformidade, minimização de cruzamentos, uniformização dos tamanhos das arestas, simetria e fluidez da representação hierárquica. A característica estética mais marcante neste método é a fluidez.

CAPÍTULO 3

HIPERGRAFOS DIRECIONADOS PARA GERÊNCIA DE PROJETOS

Como descrito no Capítulo 2, ter uma representação gráfica de um projeto modelado como um hipergrafo direcionado é ferramenta útil para gerência de projetos.

A seguir será apresentado um resumo da teoria de representações gráficas de atividades em gerência de projetos e um exemplo do uso de hipergrafos direcionados para construção de uma representação gráfica do fluxo das atividades relacionadas a um projeto.

3.1 Representações Gráficas de Atividades em Gerência de Projetos

Antes de citar as formas de se representar atividades em um projeto, vale definir o que é um projeto.

Existem várias definições de projeto disponíveis na literatura. Dentre estas, as mais utilizadas são [15]:

- Um projeto é uma organização de pessoas dedicadas que visam atingir um propósito e objetivo específico. Projetos geralmente envolvem gastos, ações ou empreendimentos únicos de alto risco e devem ser completados uma certa data por montante de dinheiro, dentro de alguma expectativa de desempenho. No mínimo, todos os projetos necessitam ter seus objetivos bem definidos e recursos suficientes para poderem desenvolver as tarefas requeridas.
- Um projeto é um processo único, que consiste em um grupo de atividades coordenadas e controladas com datas para início e término, empreendido para alcance de um objetivo conforme requisitos específicos, incluindo limitações de tempo, custo e recursos.

- Um empreendimento temporário feito para criar um produto, serviço ou resultado único.

Existem várias formas de representação do projeto, dentre elas destacam-se: o diagrama de precedência (*PDM - Preceding Diagramming Method*); o diagrama de flecha ou arcos (*ADM - Arrow Diagramming Method*); o diagrama de *Gantt*; o diagrama de marcos (*milestones*) e o diagrama de barras ou histograma de recursos.

A seguir são apresentadas as representações que mais podem se beneficiar do uso do hipergrafo direcionado no traçado.

3.1.1 Método do diagrama de precedência (ADM)

O método mais utilizado na maioria dos softwares disponíveis para gestão de projetos é o diagrama de precedência (*PDM - Preceding Diagramming Method*), que representa as atividades nos nós (*AOM - activity-on-node*), por retângulos, e as relações de precedência são estabelecidas por setas. Esta forma de representação também é conhecida como método francês. Este diagrama é utilizado no *MsProject* [3].

As precedências, que neste modelo são representadas nas setas, podem ser de vários tipos:

- Término/Início: o início do trabalho da sucessora depende do término da predecessora.
- Término/Término: o término do trabalho da sucessora depende do término da predecessora.
- Início/Início: o início do trabalho da sucessora depende do início da predecessora.
- Início/Término: o término do trabalho da sucessor depende da predecessora.

A precedência término/início: é o tipo mais usado, os demais são raramente utilizados e ainda não foram implementados de forma consistente. Observe que estes podem ser feitos, com pequenas adaptações, com a utilização do primeiro tipo.

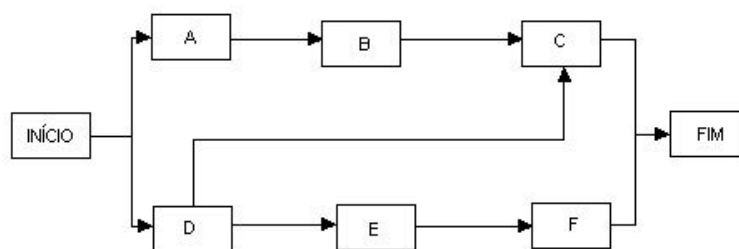


Figura 3.1: Exemplo de Diagrama de Dependências

A Figura 3.1 traz um exemplo do diagrama de precedências do software *MSPProject*.

Devido a característica de representação de precedência de atividades temporais este diagrama não permite a utilização de ciclos.

3.1.2 Método do diagrama de flecha/arco (ADM)

Embora a maioria dos softwares não utilize este tipo de representação, esta é de grande importância quando soluções de customização forem desenvolvidas, pois auxilia na programação, uma vez que utiliza as regras de programação em grafos. O método do diagrama de flecha ou arco (*ADM - Arrow Diagramming Method*) que representa, como o próprio nome sugere, as atividades nas flechas/arcos (*AOA - activity-on-arrow*) e as relações de precedência são definidas nos eventos, sendo apenas do tipo término/início.

Para a elaboração deste tipo de representação não devem existir loops, desvios condicionais ou ciclos fechados.

A Figura 3.2 traz um exemplo do diagrama de flecha.

3.1.3 Método do diagrama condicional (CDM)

Este método de representação assemelha-se ao diagrama de precedência (PDM). No entanto, nesta forma de representação são permitidos ciclos e desvios condicionados, que tanto no PDM como no ADM não são permitidos.

A Figura 3.3 traz um exemplo do diagrama condicional.

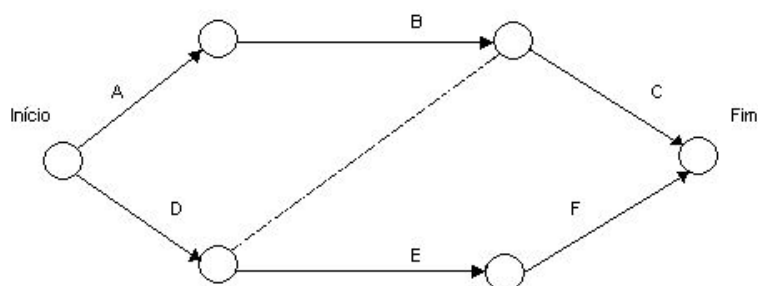


Figura 3.2: Exemplo de Diagrama de Flecha

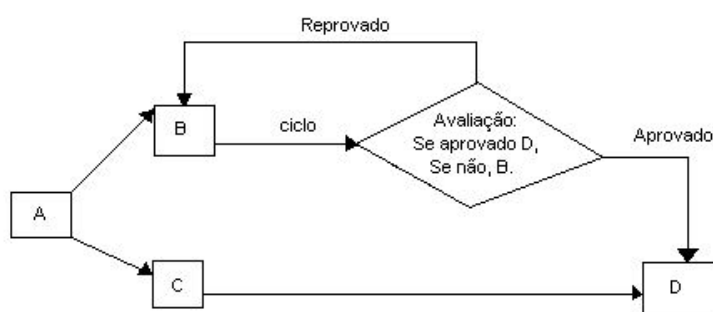


Figura 3.3: Exemplo de Diagrama Condicional

3.2 Representações de Atividades em Gerência de Projetos Mapeadas em Hipergrafos Direcionados

Tendo visto os métodos de representação gráfica de atividades em gerência de projeto, mais precisamente o Método do diagrama de precedência onde as atividades são representadas em vértices e as relações entre as mesmas representadas entre as arestas, nesta sessão será apresentada a modelagem de atividades gerenciais para hipergrafos direcionados.

O Método do diagrama de precedência foi escolhido por ser o mais utilizado para a representação gráfica e o método que facilita visualmente a interpretação do resultado gráfico em relação a detecção de gargalos e de pontos chave nos projetos.

3.3 Exemplo de Uso de Hipergrafos Direcionados para Representação das Atividades de um Projeto

Contextualizando, a COPEL [4], Companhia Paranaense de Energia, possui como ferramenta de gerência de projetos de tecnologia de informação um software chamado Girassol. Esta ferramenta foi desenvolvida para funcionar em ambiente de *intranet* via um navegador *html*.

O Girassol foi desenvolvido com o objetivo de possibilitar o cadastro e manutenção de demandas, sua posterior classificação entre projetos e serviços e seu gerenciamento. Neste gerenciamento de demandas é possível ao responsável, caso necessário, designar atividades para o auxílio do desenvolvimento e melhor gerenciamento da demanda. O responsável por esta atividade pode por sua vez criar outras atividades relacionadas e designar responsáveis para sua execução e assim sucessivamente sem limite pré-determinado criando desta forma uma árvore de atividades abaixo da demanda gerenciada. Com isso todas as atividades necessárias para o desenvolvimento da demanda ficam registradas e são gerenciadas através da ferramenta.

Observe a Figura 3.4 que exemplifica o processo. A representação textual dos relacionamentos indica a dependência entre as atividades e suas respectivas sub-atividades. Note o grafo correspondente indicando a arborização das atividades.

Estas relações identificadas entre as atividades e suas sub-atividades representadas internamente no software Girassol servem apenas para indicar uma dependência direta entre a atividade pai e suas respectivas filhas, mas não possuem relação de dependência entre as atividades e o fluxo de ocorrência das mesmas.

Tomando um projeto, no caso de desenvolvimento de software, modelado no software Girassol tem-se a seguinte visão dos produtos de projetos e suas respectivas atividades conforme as Figuras 3.5, 3.6.

Note que não há nenhuma representação do fluxo de ocorrência das atividades o que torna muito difícil para o administrador do projeto identificar os gargalos existentes.

Durante o cadastro das atividades a única relação existente é a de parentesco, ou seja

Representação Interna

Demanda X, Nivel 0 (DX)
Atividade 1 da Demanda X, Nivel 1 (A1) - Sub-Atividade 1 da Atividade 1, Nivel 2 (A1.1) - Sub-Atividade 1 da Sub-Atividade 1, Nivel 3 (A1.1.1) - Sub-Atividade 2 da Sub-Atividade 1, Nivel 3 (A1.1.2) - Sub-Atividade 2 da Atividade 1, Nivel 2 (A1.2) - Sub-Atividade 1 da Sub-Atividade 2, Nivel 3 (A1.2.1)
Atividade 2 da Demanda X, Nivel 1 (A2) - Sub-Atividade 1 da Atividade 2, Nivel 2 (A2.1) - Sub-Atividade 1 da Sub-Atividade 1, Nivel 3 (A2.1.1) - Sub-Atividade 1 da Sub-Atividade 1, Nivel 4 (A2.1.1.1)
Atividade 3 da Demanda X, Nivel 1 (A3) - Sub-Atividade 1 da Atividade 3, Nivel 2 (A3.1) - Sub-Atividade 2 da Atividade 2, Nivel 2 (A3.2)

Exemplo de Representação Gráfica

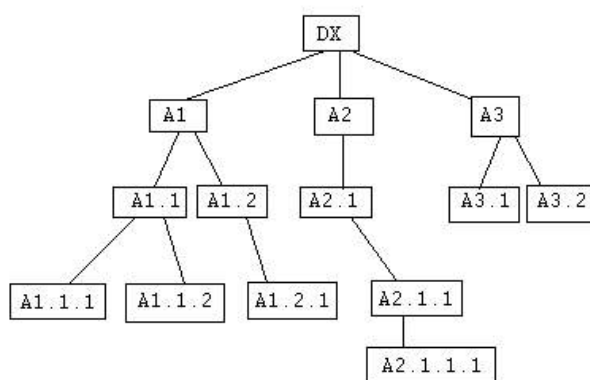


Figura 3.4: Exemplo de Representação de Projeto no Software Girassol

de uma atividade pai, são cadastradas as atividades filhas. Informações de precedência como início/início, início/término, término/início e término/término não são cadastradas inviabilizando a visualização de fluxo entre as atividades e visualização de criticidade e gargalos.

Para tornar mais fácil a visualização do hipergrafo extraído do projeto serão dados nomes para cada objeto conforme a tabela representada na Figura 3.7.

Dados os nomes para cada objeto do projeto, em conjunto com o líder do projeto, representante da COPEL, modelou-se a relação entre os objetos determinando o fluxo. Modelado o relacionamento chegou-se as relações que determinam o hipergrafo direcionado com suas hiper-arestas correspondentes (Figura 3.8).

A Figura 3.9 representa graficamente o hipergrafo direcionado que modela o fluxo das atividade do projeto utilizado como exemplo, foi utilizado o método do diagrama de precedência como forma de representação. Nota-se claramente observando o gráfico

.. Produtos do Projeto:					
Título	Previsão	Realizado	Responsável	Situação	
<u>Proposta de Estrutura Básica de Armazenamento</u>	12/07/2005 a 01/08/2005	29/07/2005 a 02/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
- <u>Elaborar proposta de estrutura básica de armazenamento</u>	12/07/2005 a 01/08/2005	29/07/2005 a 02/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
<u>Análise de Requisitos</u>	27/07/2005 a 04/08/2005	12/07/2005 a 31/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
- <u>Apresentação do Domino.Doc</u>	27/07/2005 a 03/08/2005	09/08/2005 a 09/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
- <u>Elaborar documento da análise de requisitos</u>	28/07/2005 a 03/08/2005	12/08/2005 a 31/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
-- <u>Levantar os requisitos dos grupos de trabalho</u>	11/08/2005 a 26/08/2005	12/08/2005 a 24/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
-- <u>Confeccionar o documento de análise de requisitos</u>	15/08/2005 a 26/08/2005	29/08/2005 a 31/08/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
<u>Ata da reunião para fechamento dos requisitos</u>	01/08/2005 a 05/08/2005	12/07/2005	Alessandro Rogerio Bergamo Silvestre	Execução	
- <u>Efetuar reunião de fechamento dos requisitos</u>	01/08/2005 a 05/08/2005		Alessandro Rogerio Bergamo Silvestre	Execução	
- <u>Elaborar ata da reunião</u>	01/08/2005 a 05/08/2005		Alessandro Rogerio Bergamo Silvestre	Execução	
<u>Cronograma</u>	18/07/2005 a 10/08/2005	25/07/2005	Alessandro Rogerio Bergamo Silvestre	Execução	
- <u>Elaborar o cronograma de atividades</u>	18/07/2005 a 27/07/2005	25/07/2005 a 27/07/2005	Alessandro Rogerio Bergamo Silvestre	Concluído	
- <u>Revisar o cronograma</u>	02/08/2005 a 10/08/2005		Alessandro Rogerio Bergamo Silvestre	Execução	
<u>Estrutura de armazenamento (pastas) no Domino.Doc</u>	03/08/2005 a 26/08/2005	12/07/2005	Alessandro Rogerio Bergamo Silvestre	Execução	
- <u>Preparar proposta da estrutura de armazenamento no Domino.Doc</u>	03/08/2005 a 12/08/2005		Alessandro Rogerio Bergamo Silvestre	Execução	
- <u>Aprovar estrutura de armazenamento no Domino.Doc</u>	05/08/2005 a 23/08/2005		Alessandro Rogerio Bergamo Silvestre	Execução	

Figura 3.5: Exemplo de Projeto Visto pelo Software Girassol(Parte 1 de 2)

a utilidade da modelagem e armazenamento das características de precedência entre atividades. Construindo a representação visual forma-se uma ferramenta que auxilia no gerenciamento das atividades de maneira a contornar os entraves que possam ocorrer.

3.4 Abrangência da Representação

O foco deste trabalho é voltado a determinar o padrão de desenho de hipergrafos direcionados mais adequado à representação tendo como premissa atender as características predominantes na especificidade dos fluxos inerentes a objetos gerenciados em projetos como no caso dos projetos gerenciados através do Girassol.

Os hipergrafos direcionados produzidos devem contemplar as representações para o

- Definir a matriz de responsabilidades	03/08/2005 a 12/08/2005		Rogério Bergamo Silvestre	Execução
- Implementar a matriz de responsabilidades no Domino.Doc	10/08/2005 a 31/08/2005		Anderson J. Azambuja Guiera	Execução
Customizações no Domino.Doc	01/08/2005 a 31/08/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Implementar as customizações no Domino.Doc	01/08/2005 a 26/08/2005		Anderson J. Azambuja Guiera	Execução
Plano de teste	02/08/2005 a 10/08/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Elaborar o plano de teste	02/08/2005 a 10/08/2005		Anderson J. Azambuja Guiera	Execução
Casos de teste	02/08/2005 a 31/08/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Elaborar casos de teste	02/08/2005 a 12/08/2005		Anderson J. Azambuja Guiera	Execução
- Executar casos de teste	04/08/2005 a 31/08/2005		Anderson J. Azambuja Guiera	Execução
Termo de Aceitação dos testes	24/08/2005 a 14/09/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Disponibilizar aplicação para usuário testador	24/08/2005 a 02/09/2005		Anderson J. Azambuja Guiera	Execução
- Acompanhar testes do usuário testador	24/08/2005 a 14/09/2005		Anderson J. Azambuja Guiera	Execução
Manual do Usuário	08/08/2005 a 02/09/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Elaborar o manual para o usuário da aplicação	08/08/2005 a 02/09/2005		Anderson J. Azambuja Guiera	Execução
Plano de Entrega	08/08/2005 a 16/09/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Elaborar plano de entrega	08/08/2005 a 24/08/2005		Anderson J. Azambuja Guiera	Execução
- Executar plano de entrega	25/08/2005 a 16/09/2005		Anderson J. Azambuja Guiera	Execução
Relatório de Encerramento	30/08/2005 a 20/09/2005	12/07/2005	Alessandro Rogério Bergamo Silvestre	Execução
- Elaborar relatório de encerramento do projeto	30/08/2005 a 16/09/2005		Alessandro Rogério Bergamo Silvestre	Execução
- Reunião com os usuários para conclusão do projeto	31/08/2005 a 20/09/2005		Alessandro Rogério Bergamo Silvestre	Execução
Gerência de Configuração	08/08/2005 a 20/09/2005	12/07/2005	Anderson J. Azambuja Guiera	Execução
- Disponibilizar a estrutura para gerência de configuração	08/08/2005 a 24/08/2005		Anderson J. Azambuja Guiera	Execução
- Acompanhar e garantir a gerência de configuração	08/08/2005 a 20/09/2005		Anderson J. Azambuja Guiera	Execução

Figura 3.6: Exemplo de Projeto Visto pelo Software Girassol(Parte 2 de 2)

método do diagrama de precedência e para o método do diagrama de flecha/arco. O método do diagrama condicional não será contemplado neste trabalho já que o mesmo não é utilizado na representação do fluxo de atividades e sim na representação de processos. Deste modo não haverá a preocupação na representação de ciclos.

Os exemplos de projetos modelados e estudados em forma de hipergrafos direcionados foram fornecidos pela COPEL.

Nome da Atividade Gerenciada	Apelido da Atividade Gerenciada
Proposta de Estrutura Básica de Armazenamento	P1
Elaborar Proposta de Estrutura Básica de Armazenamento	AP1
Análise de Requisitos	P2
Apresentação do Domino.Doc	AP21
Elaborar documento de análise de requisitos	AP22
Levantar requisitos dos grupos de trabalho	AP221
Confeccionar o documento de análise de requisitos	AP222
Ata de Reunião para fechamento de requisitos	P3
efetuar reunião de fechamento de requisitos	AP31
elaborar ata de reunião	AP32
Cronograma	P4
Elaborar o cronograma de atividades	AP41
Revisar o cronograma	AP42
Estrutura de armazenamento (pastas) no Domino.doc	P5
Preparar proposta da estrutura de armazenamento no Domino.doc	AP51
aprovar estrutura de armazenamento no Domino.doc	AP52
Implementar estrutura de armazenamento no Domino.doc	AP53
Matriz de Responsabilidades (acessos)	P6
definir a matriz de responsabilidades	AP61
Implementar a matriz de responsabilidades no Domino.doc	AP62
Customizações no Domino.doc	P7
Implementar as customizações no Domino.doc	AP71
Plano de Testes	P8
Elaborar o plano de testes	AP81
Casos de Teste	P9
Elaborar os casos de teste	AP91
executar os casos de teste	AP92
Termo de aceitação dos testes	P10
disponibilizar aplicação para usuário testador	AP101
acompanhar testes do usuário testador	AP102
Manual do usuário	P11
Elaborar o manual para o usuário da aplicação	AP111
Plano de Entrega	P12
elaborar plano de entrega	AP121
executar plano de entrega	AP122
Relatório de encerramento	P13
elaborar relatório de encerramento do projeto	AP131
reunião com usuário para a conclusão do projeto	AP132
Gerência de configuração	P14
Acompanhar e garantir a gerência de configuração	AP141

Figura 3.7: Tabela com os nomes dos Objetos

```

M0->P4,P14,P1
P1->AP11
P4->AP41,AP42
P14->AP141
AP11->P2
P2->AP21,AP22
AP22->AP221,AP222
AP21,AP221,AP222->P3
P3->AP31,AP32
AP32,AP31->P5,P6,P7,P8,P9
P5->AP51,AP52,AP53
P6->AP61,AP62
P7->AP71
P8->AP81
P9->AP91,AP92
AP51,AP52,AP53,AP61,AP62,AP71,AP81,AP91,AP92->P10,P12
P10->AP101,AP102
P12->AP121,AP122
AP101,AP102->P11
P11->AP111
AP111,AP121,AP122->P13
P13->AP131,AP132
AP141,AP131,AP132->FIM

```

Figura 3.8: Exemplo de Arquivo Texto com Representação de um Hipergrafo Direcionado

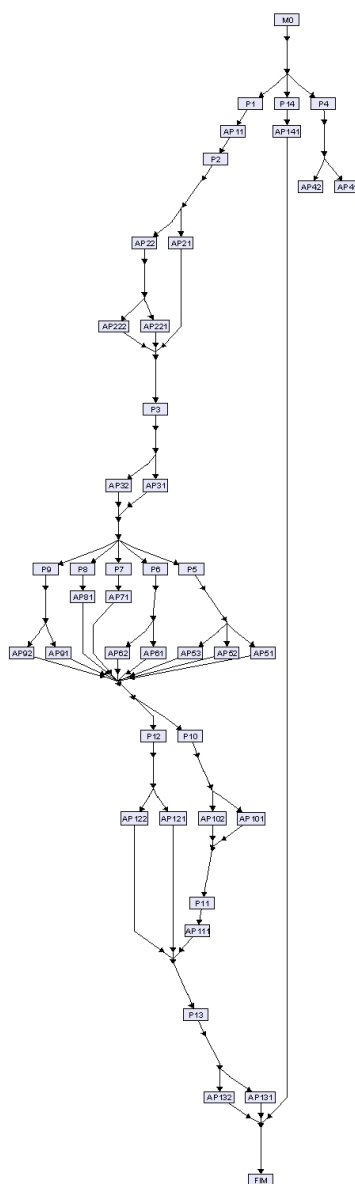


Figura 3.9: Hipergrafo Direcionado traçado a partir de exemplo de projeto do software Girassol

CAPÍTULO 4

ESTUDO DA FORMA MAIS ADEQUADA DE TRAÇADO DE HIPERGRAFOS DIRECIONADOS PARA GERÊNCIA DE PROJETOS

Neste capítulo serão abordados os passos efetuados para a escolha do algoritmo e padrão de traçado que mais se adaptam ao hipergrafo direcionado foco de estudo, e a forma com que se comprovou na prática o formato a ser adotado.

4.1 Escolha do Algoritmo de Traçado e Padrão de Desenho

Observando os hipergrafos direcionados voltados para modelagem de fluxo de atividades gerenciais notam-se as características de uniformidade, minimização de cruzamentos, uniformização dos tamanhos das arestas, simetria e fluidez da representação hierárquica.

Desta forma o padrão de desenho ortogonal não é o mais indicado por ter falta de cuidado em relação a representabilidade de fluxos, fator de forte importância no tipo de grafo trabalhado.

Da mesma forma o uso do algoritmo Force Directed Placement que gera um traçado baseado no padrão linha reta também não se aplica a solução do problema já que o mesmo preza pela simetria deixando de lado, como no caso do traçado ortogonal, em segundo plano de importância, a representabilidade do fluxo.

A forma mais adequada de desenhar o hipergrafo em questão é utilizar o algoritmo de traçado hierárquico gerando o padrão de desenho poligonal. As características de algoritmo casam com as características almejadas no desenho dos hipergrafos em questão. A representação do fluxo é bem abordada, juntamente com as outras características.

A abordagem de representação adotada neste trabalho será o uso do algoritmo de traçado de grafos de forma hierárquica aplicado ao traçado de hipergrafos direcionados.

De modo a comprovar na prática a eficiência desta representação desenvolveu-se um software de traçado de hipergrafos direcionados. Nas seções a seguir serão descritas as etapas realizadas de modo a atingir este objetivo.

4.2 Transformação de Hipergrafos Direcionados em Grafos Direcionados

Para testar o traçado de hipergrafos direcionados com alguns padrões de desenho com o objetivo de escolher a abordagem mais coerente para o tipo de hipergrafo escolhido torna-se extremamente útil a transformação de hipergrafos direcionados em grafos direcionados.

O objetivo é pegar o produto da transformação e utilizar softwares especialistas em traçar grafos de modo a estudar as características de cada padrão.

A técnica de transformação utilizada neste trabalho é descrita em [10] e [11]. Técnica esta que dita uma forma de transformação de hipergrafos direcionados para grafos direcionados guardando as características de planaridade.

Toma-se um hipergrafo direcionado qualquer e para cada hiper-aresta promove-se a transformação da seguinte forma:

Para cada hiper-aresta, cria-se no lugar do fluxo, dois vértices interligados por uma aresta. Cada um dos vértices criados será ligado via arestas com os vértices que compõem o grupo de origem e de destino da hiper-aresta.

Ou seja, para cada hiper-aresta $h = (X, Y)$ crie os arcos $(X, C1)$, $(C1, C2)$ e $(C2, Y)$. A Figura 4.1 exemplifica a transformação.

Tomando como base a técnica de transformação descrita, foi desenvolvida uma ferramenta de transformação. Esta foi construída em linguagem Java e tem como característica receber dois parâmetros de entrada que são respectivamente: nome do arquivo de entrada, nome do arquivo de saída e o nome do arquivo de saída em formato *CSV*, formato utilizado pelo software *GDE* [14].

Observando os parâmetros de entrada, a chamada do programa de transformação criado ficou da seguinte forma;

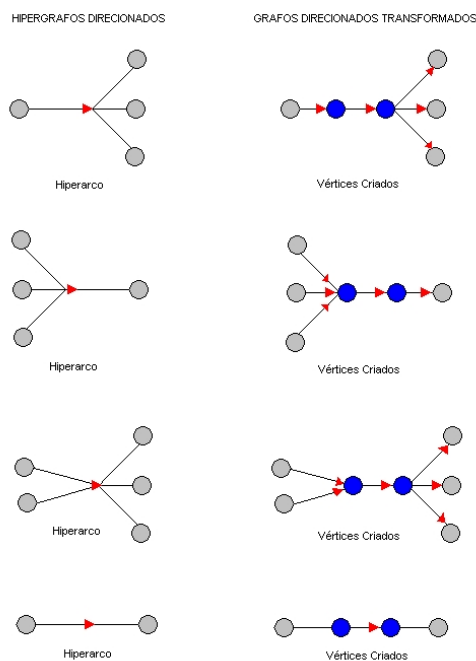


Figura 4.1: Exemplo de Transformação de Hipergrafo Direcionado Para Grafo Direcionado

Hiper2Graph entrada saida.cvs

Quanto ao formato do arquivo de entrada, o mesmo deve ser composto de uma ou mais linhas. Cada linha deverá conter um par $O \rightarrow D$ onde O é um conjunto não vazio de vértices representando origens separados por vírgula quando necessário e D é outro conjunto não vazio de vértices também separados por vírgula quando necessário. O conjunto de caracteres \rightarrow representa a orientação do hipergrafo. Desta maneira tornou-se possível a representação de um hipergrafo direcionado de maneira simples e clara.

Desta forma observe a Figura 4.2 que exemplifica alguns casos de transformação de hipergrafos direcionados em grafos direcionados.

Note nos exemplos que os vértices criados pelo algoritmo são nomeados iniciando com a letra C de modo a tornar fácil a distinção entre os demais vértices já existentes.

4.3 Testes de Traçado de Hipergrafos Direcionados

Tendo implementado o transformador de hipergrafos direcionados em grafos direcionados tornaram-se possíveis os testes de traçado nos padrões disponíveis na ferramenta *GDE*.

Esta ferramenta foi escolhida devido a qualidade do seu traçado e variedade de padrões

<p>entrada1.txt</p> <p>1->2,3,4</p>	<p>entrada2.txt</p> <p>1,2,3->4</p>	<p>entrada3.txt</p> <p>1,2,3->4,5</p>	<p>entrada4.txt</p> <p>1->4 2->3 1->3 2->4 1,2->3,4</p>
<p>saida1.csv</p> <p>1:C1 C1:C2 C2:2 C2:3 C2:4</p>	<p>saida2.csv</p> <p>1:C1 2:C1 3:C1 C1:C2 C2:C4</p>	<p>saida3.csv</p> <p>1:C1 2:C1 3:C1 C1:C2 C2:4 C2:5</p>	<p>saida4.csv</p> <p>1:C1 C1:C2 C2:4 2:C3 C3:C4 C4:3 1:C5 C5:C6 C6:3 2:C7 C8:C8 C8:4 1:C9 C1:C2 C2:4 1:C9 2:C9 C9:C10 C10:3 C10:4</p>

Figura 4.2: Exemplos de Transformação de Hipergrafo Direcionado Para Grafo Direcionado

de traçado programados.

Primeiramente foram efetuados testes preliminares com hiper-aresta simples, sem levar em conta os exemplos reais de modo a verificar as características estéticas mais gerais.

Observe que a Figura 4.3 traz os traçados automáticos nos padrões Hierárquico, Ortogonal e por Simetria produzidos automaticamente através da ferramenta *GDE* após a importação dos arquivos chamados *saida1.csv*, *saida2.csv*, *saida3.csv*, *saida4.csv* que são o resultado da transformação feita através da ferramenta de transformação desenvolvida aplicados aos arquivos *entrada1.txt*, *entrada2.txt*, *entrada3.txt* e *entrada4.txt* descritos anteriormente.

Observando a Figura 4.3 nota-se que mesmo para exemplos pequenos como os utilizados nos primeiros experimentos já é possível notar a capacidade de representar fluxo utilizando-se do algoritmo hierárquico. Com o aumento da complexidade dos desenhos que ocorre da esquerda para a direita o poder de representação de fluxos torna-se mais explícito.

Já analisando o resultado do uso do algoritmo de traçado ortogonal observa-se que para

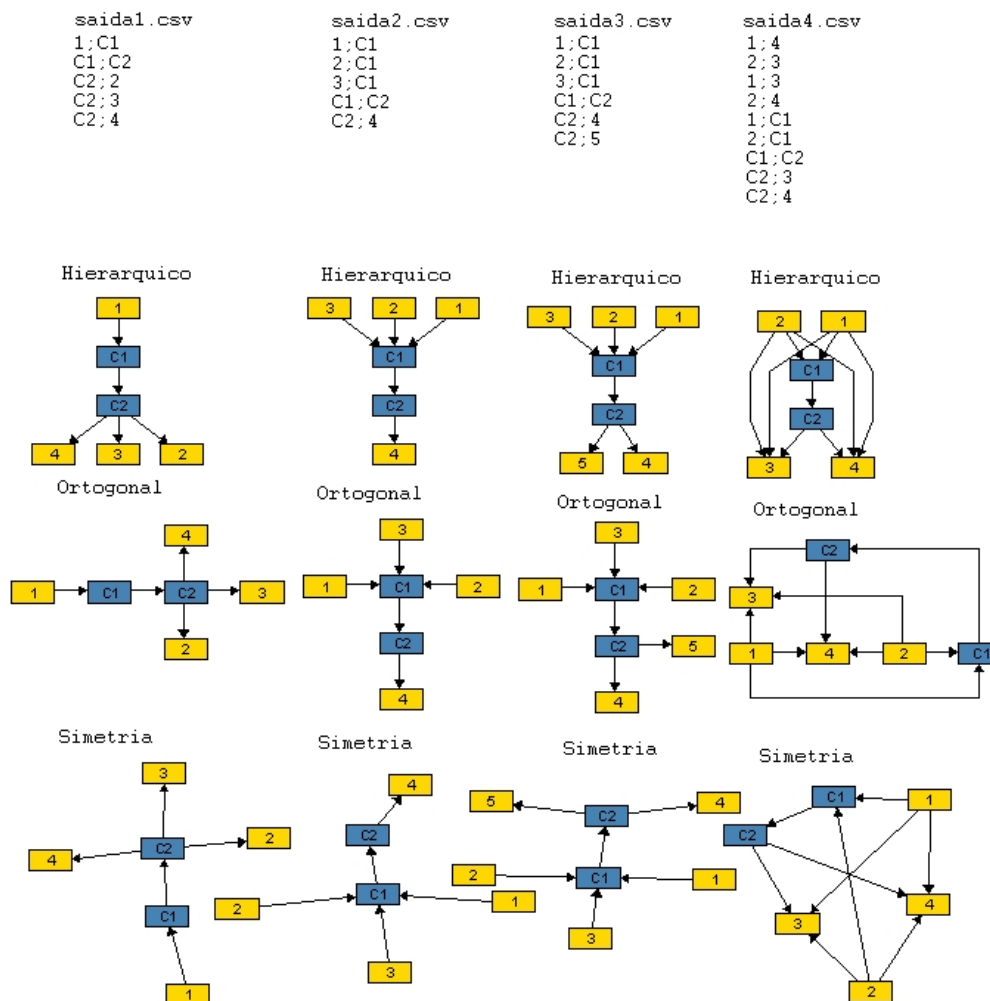


Figura 4.3: Exemplo do Estudo Preliminar de Traçado

exemplos simples, como o encontrado no arquivo `saida1.csv`, a diferença na qualidade de representação dos fluxos não fica tão evidente. Com o aumento da complexidade do fluxo, mais precisamente nos dois últimos exemplos, a diferença na qualidade desta característica fica evidente. O algoritmo ortogonal acaba perdendo esta característica.

O algoritmo de Simetria (*Force Directed Placement*), nos primeiros exemplos, do primeiro ao quarto hipergrafo direcionado, apresentou uma boa aderência a característica aqui comparada. Houve um pouco de perda de qualidade devido a característica de prezar pela distribuição uniforme dos vértices, buscando a padronização das distâncias entre eles, o que levou a um mascaramento no fluxo, fato que não ocorre no algoritmo hierárquico. No último exemplo, um fluxo um pouco mais complexo, já trouxe problemas no uso do uso

do algoritmo em relação ao fluxo também devido as características inerentes ao mesmo.

Após estes testes preliminares foram efetuados testes com hipergrafos direcionados voltados ao foco de estudo, fluxo de atividades em projetos. A seguir segue a análise efetuada no hipergrafo modelado no Capítulo 4 do exemplo real retirado do software Girassol.

Observando o resultado do uso do algoritmo de traçado automático por simetria sobre o hipergrafo objeto de estudo, Figura 4.4, tem-se com o caso real a verificação de que com este algoritmo não é possível uma boa representação visual do fluxo inerente à modelagem tornando-se não adequado o uso deste algoritmo para o traçado dos hipergrafos objeto deste estudo.

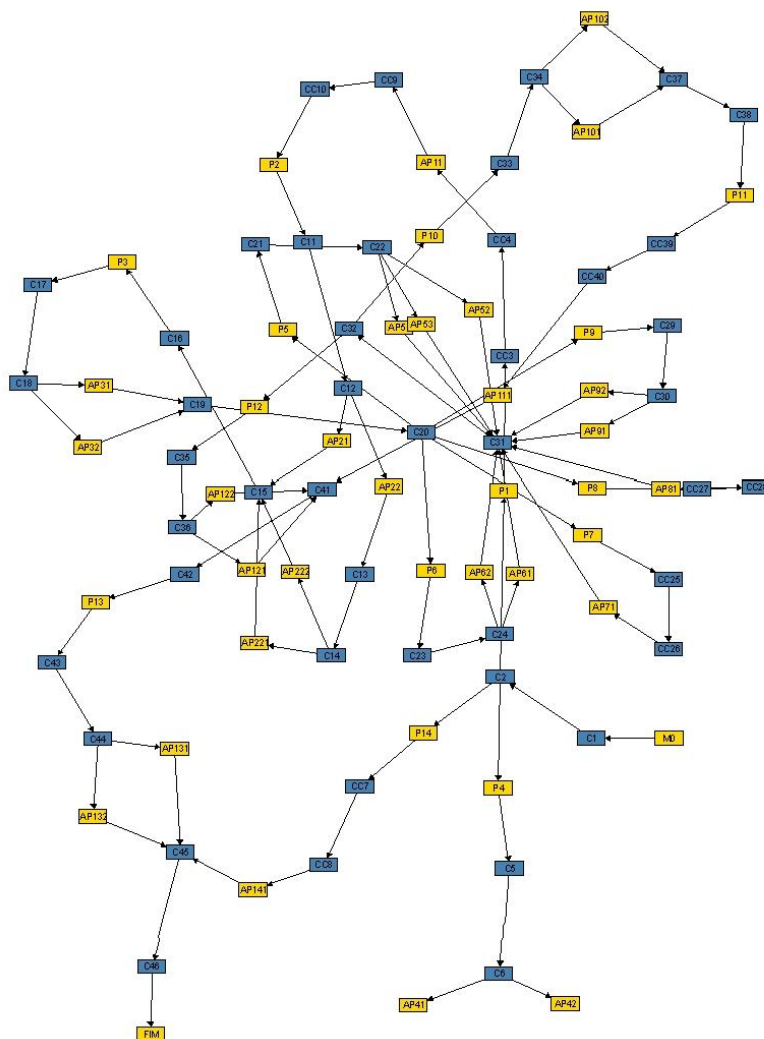


Figura 4.4: Traçado Automático de Hipergrafo do Girassol - Algoritmo Simetria

O mesmo acontece no traçado do hipergrafo quando utilizamos o algoritmo de traçado ortogonal, Figura 4.5. Há uma capacidade da visualização do fluxo um pouco maior do que observado no caso anterior, mas ainda insuficiente para que a pessoa que for fazer uso desta imagem tenha uma clara interpretação do fluxo podendo analisar o mesmo.

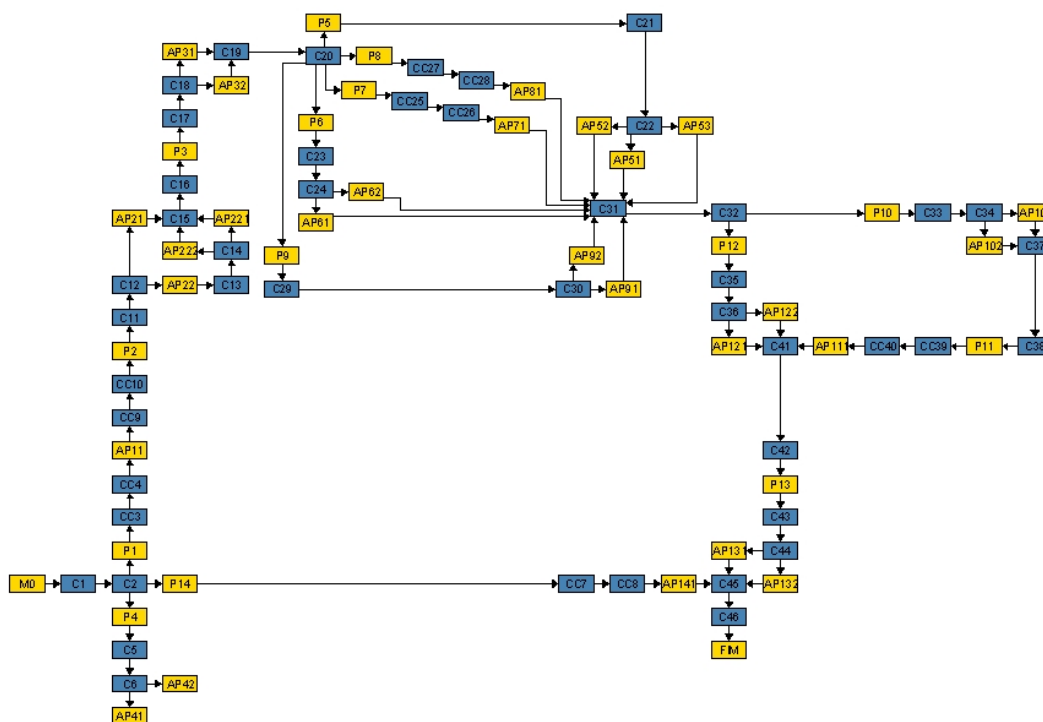


Figura 4.5: Traçado Automático de Hipergrafo do Girassol - Algoritmo Ortogonal

Observando o resultado do traçado automático do hipergrafo utilizando-se do algoritmo hierárquico, Figura 4.6, é visualmente clara a qualidade da representação do fluxo atestando que esta é a escolha mais adequada tratando-se do tipo de hipergrafo especial em que o estudo é focado.

Tendo esta análise dos resultados do estudo do traçado com base nos resultados dos três algoritmos comprovou-se de forma prática que a maneira mais adequada de traçar hipergrafos direcionados para gerenciamento de projetos é utilizando o algoritmo hierárquico. Sendo assim este será o algoritmo a ser adaptado.

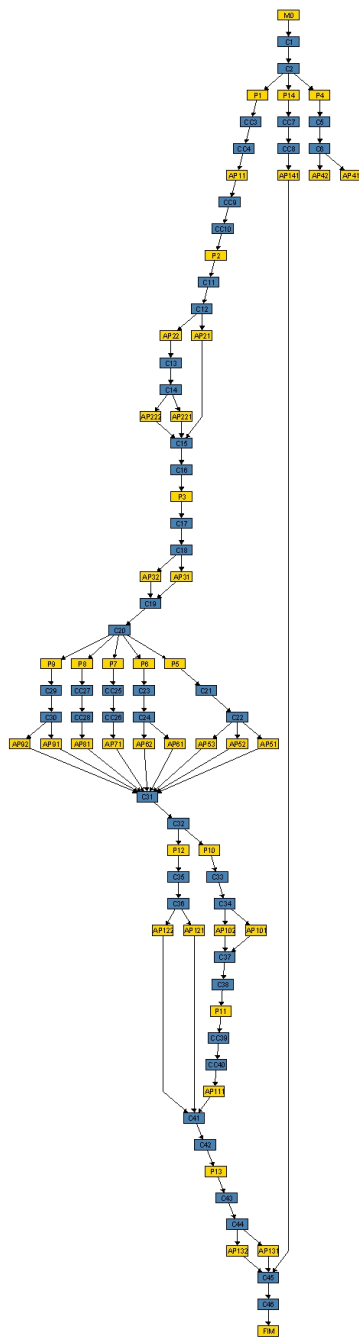


Figura 4.6: Traçado Automático de Hipergrafo do Girassol - Algoritmo Hierarquico

CAPÍTULO 5

FERRAMENTA DE TRAÇADO AUTOMÁTICO DE HIPERGRAFOS DIRECIONADOS

Este capítulo destina-se a descrever as etapas empregadas na adaptação da ferramenta VGJ[18] de modo a prepará-la para traçar automaticamente hipergrafos direcionados.

5.1 Desenvolvimento da Ferramenta

Analisado o problema de decidir a mais adequada forma de representação do hipergrafo direcionado relacionado a representação das atividades em um projeto, ou seja, visto que a representação hierárquica é a que, neste caso em especial, mais agrega valor a representação, pode-se investir na construção de uma ferramenta apropriada para produzir este traçado automático.

Atualmente não há ferramentas preparada para produzir o resultado esperado. Há várias ferramentas de traçado automático de grafos mas nenhuma preparada pra trabalhar com hipergrafos direcionados. Sendo assim escolheu-se uma ferramenta de desenho de grafos para ser adaptada permitindo o traçado desejado abrindo um leque de possibilidades de aplicações.

Foi escolhido a ferramenta VGJ[18] para adaptação e uso neste trabalho por ser implementada em Java com política de código aberto [6] e preparada para funcionar através da *Web* sobre tecnologia *emphapplet*. Com isso a distribuição da ferramenta após as modificações fica facilitada. Outra característica que guiou a escolha da ferramenta foi o fato da mesma possuir implementado um algoritmo de traçado de grafos direcionados hierarquicamente. Com isso, a adaptação para o traçado do hipergrafo direcionado fica restrito a adaptações e não a construção total de um algoritmo novo.

A Figura 5.1 mostra a tela inicial do aplicativo VGJ.

A seguir descreve-se as implementações efetuadas na ferramenta de modo a atingir o

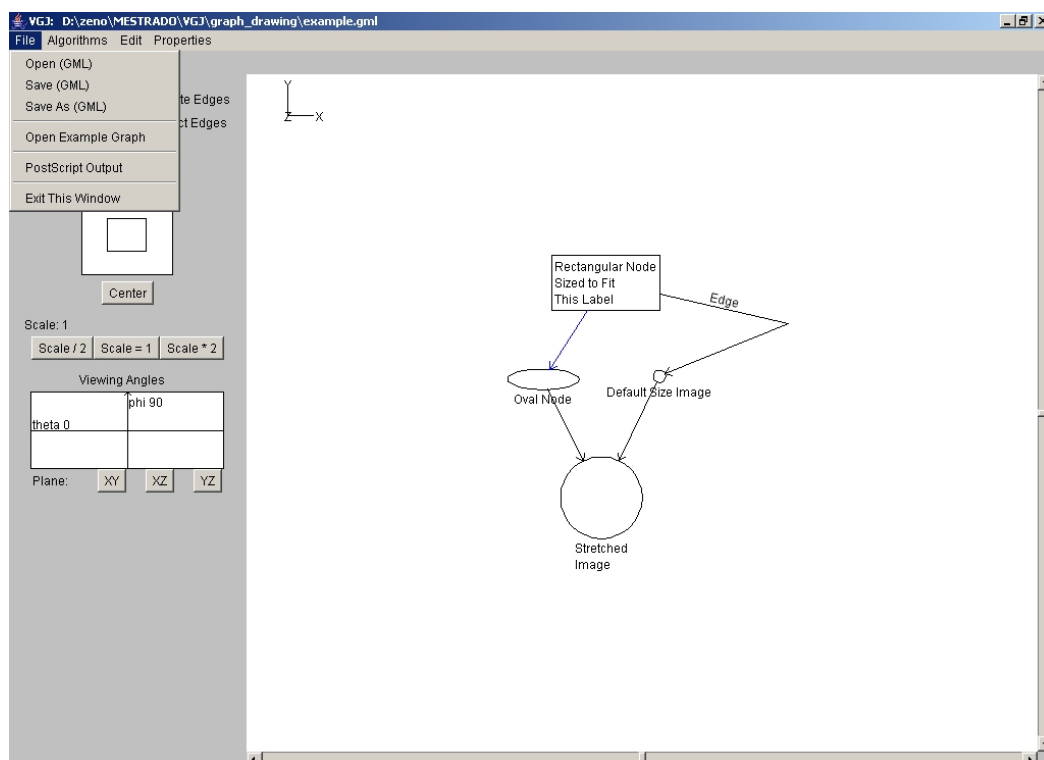


Figura 5.1: Exemplo da Tela Inicial do Software VGJ

propósito determinado.

5.1.1 Passos da Implementação da Ferramenta

Escolhida a ferramenta VGJ[18] a ação inicial foi de analisar seu funcionamento de modo a se familiarizar com a mesma e desvendar todas as suas características e funcionalidades já implementadas. Fora a análise de funcionamento foi necessário um estudo sobre o código fonte, entender as classes principais e a distribuição dos objetos implementados o que possibilitaria as implementações necessárias.

Feito este estudo prévio partiu-se para a adaptação. A primeira implementação efetuada no programa original foi referente a permitir a leitura de um arquivo contendo uma representação de hipergrafo direcionado de modo que o programa entenda o mesmo como um grafo direcionado.

Inicialmente padronizou-se o tipo de arquivo de entrada da mesma forma como feito no experimento apresentado anteriormente sobre o transformador de hipergrafo direcionados em grafos direcionados. Ou seja, o arquivo de entrada deve ser composto de uma ou

mais linhas. Cada linha deverá conter um par $O \rightarrow D$ onde O é um conjunto não vazio de vértices representando origens separados por vírgula quando necessário e D é outro conjunto não vazio de vértices também separados por vírgula quando necessário. O conjunto de caracteres \rightarrow representa a orientação do hipergrafo. Desta maneira tornou-se possível a representação de um hipergrafo direcionado de maneira simples e clara.

Definido o padrão foi desenvolvida no programa uma nova forma de leitura de arquivos específica para hipergrafos direcionados conforme indicado na Figura 5.2.

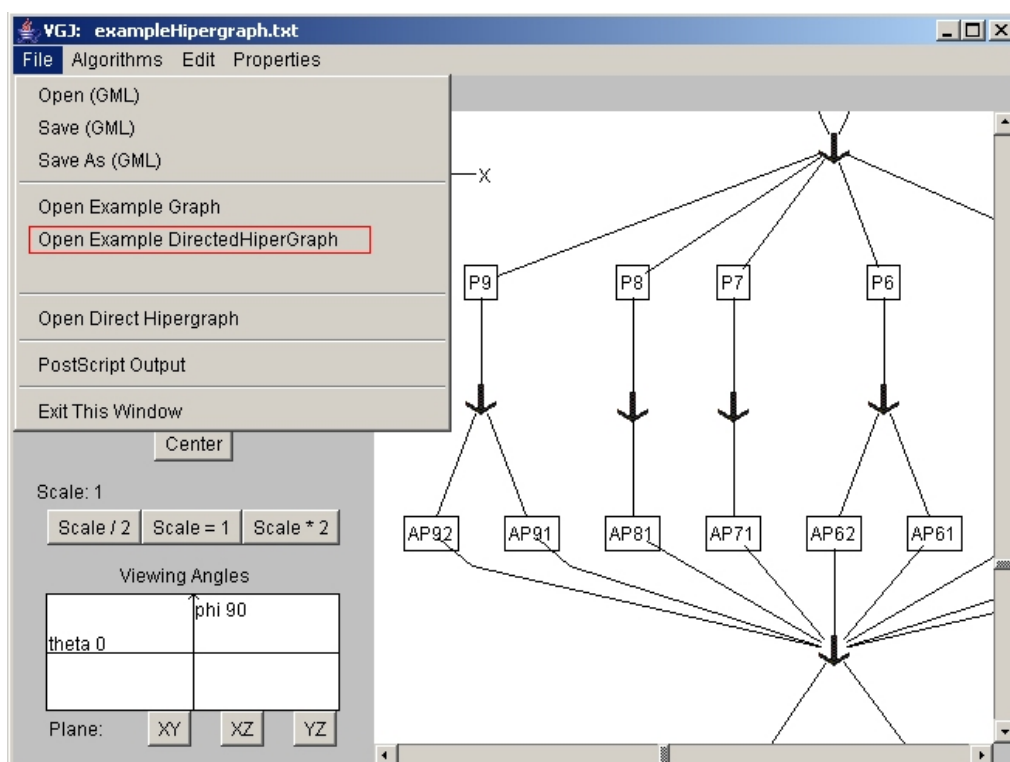


Figura 5.2: Exemplo da Tela Inicial do Software VGJ com Alteração Para Leitura de Hipergrafos Direcionados

Neste ponto o programa estava preparado para ler arquivos contendo a representação padronizada de hipergrafos direcionados. Com isso desenvolveu-se então a parte de transformação de hipergrafos direcionados em grafos direcionados. O algoritmo utilizado foi baseado na teoria de transformação apresentada anteriormente [10].

O padrão de arquivos entendido pelo programa é o formato GML [17]. Este formato de representação de grafos não está preparado para trabalhar com o tipo de grafo em questão. Desta forma foi necessário promover a transformação dos objetos em memória contendo a representação do grafo direcionado obtido de modo a criar uma representação

GML em memória para que o programa possa trabalhar com a mesma.

Ainda nesta etapa inicial alguns cuidados com a representação gráfica foram tomados de modo a tornar o desenho resultado intuitivo e representativo em relação ao fluxo. Os cuidados tomados foram em relação à forma de representação dos vértices criados durante a transformação. De modo a não deixá-los de maneira crua, ou seja, com a mesma representação utilizada para os vértices normais, como retângulos, círculos ou ovais, padronizou-se que um vértice criado de origem teria tamanho zero, ou seja, ficaria invisível e um vértice criado de destino teria uma representação especial de modo a demonstrar a união ou partida do fluxo representado na hiper-aresta. Assim foi desenvolvida uma representação gráfica com tal objetivo (Figura 5.3).

A Figura 5.4 mostra na letra "a" como ficaria a representação crua de uma hiper-aresta simples, ou seja os dois vértices criados durante o algoritmo de transformação e na letra "b" uma hiper-aresta com o primeiro tratamento, origem invisível e destino com a representação visual criada.



Figura 5.3: Padrão de Representação de Hiper-aresta Criado

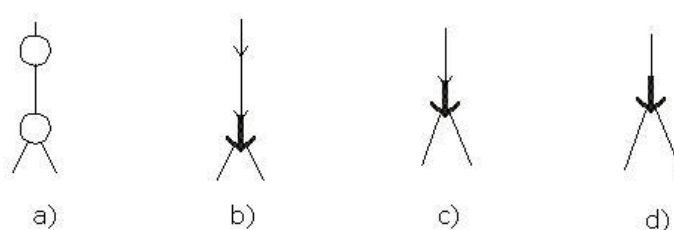


Figura 5.4: Exemplos de Representação de Hiper-arestas

Com estas modificações iniciais concretizou-se a primeira etapa, de leitura e transformação, permitindo o traçado automático utilizando o algoritmo hierárquico implementado na ferramenta. Somente com estes tratamento já era possível traçar hiprgrafos direcionados.

Observando os resultados obtidos com estas modificações iniciais observou-se que as

modificações efetuadas nos vértices criados não foram suficientes para se chegar ao estado visual considerado o mais adequado para a representação dos fluxos. De modo a resolver este problema a abordagem utilizada foi a de unir estes dois vértices artificiais criado durante a transformação em um grupo e alterar a representação deste grupo para que este represente a hiper-aresta resolvendo desta forma o problema de representação do hiper-aresta que foi transformada anteriormente.

Para conseguir este resultado, após a execução do algoritmo de traçado hierárquico a representação do grafo é passado por um segundo algoritmo de modo que este varra a representação e crie os grupos onde for necessário. A Figura 5.4 letra "c" mostra o resultado da criação do grupo.

A Figura 5.5 traz o resumo gráfico das implementações e tratamentos efetuados na ferramenta de modo a possibilitar o traçado de hipergrafos direcionados.

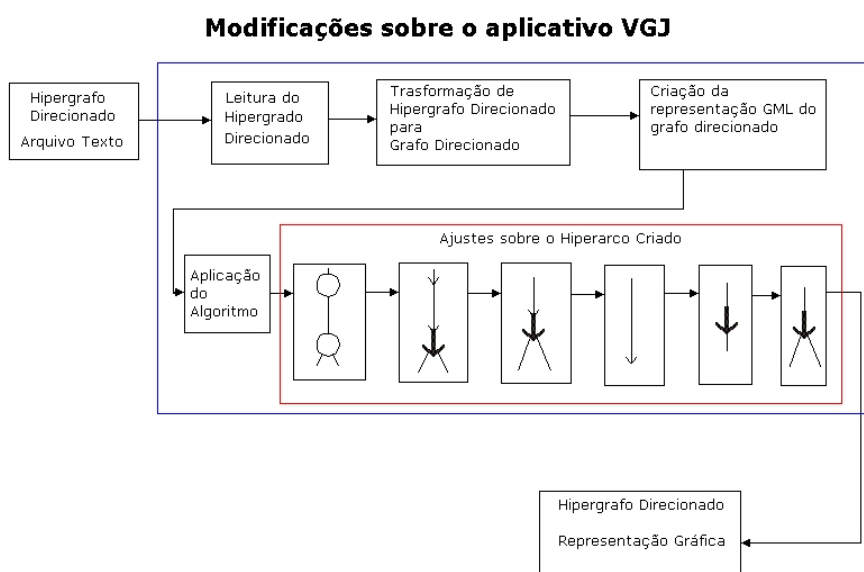


Figura 5.5: Resumo Gráfico das Modificações

5.2 Resultados Obtidos

A Figura 5.6 traz três representações gráficas do mesmo hipergrafo direcionado. Hipergrafo este que foi modelado no Capítulo 4 a partir de exemplo prático extraído do software Girassol da Copel. A letra "a" mostra uma representação esperada, representação esta que

traz embutida as características desejáveis para a boa interpretação do problema de fluxos em atividades de projetos. A letra "b" é o resultado do traçado do mesmo hipergrafo direcionado utilizando a ferramenta GDE para o estudo da abordagem mais adequada. Note que este traçado foi feito com a representação de um grafo direcionado resultado da transformação do hipergrafo objeto de origem. Finalmente a letra "c" é o resultado gráfico obtido com o uso da ferramenta VGJ após a aplicação das implementações aqui descritas.

O resultado obtido possui uma ótima qualidade de representação de fluxo cumprindo assim os objetivos pré estabelecidos anteriormente. Nota-se que a distribuição dos vértices contém a característica de que a direção da aresta do ancestral sempre é anterior aos seus descendentes, o balanceamento dos vértices ocorre da direita para a esquerda nos níveis, há uma minimização de cruzamentos entre as arestas e poucos cotovelos.

5.3 Representação de Ciclos

Conforme mencionado anteriormente, para a representação das atividades e suas inter-relações na modelagem dos eventos de um projeto não há a necessidade de ter-se a preocupação com a representação de ciclos. Entretanto sabe-se da importância da representação desta característica para outras aplicações.

O algoritmo implementado na ferramenta escolhida para aproveitamento, VGJ, não é capaz de trabalhar com hipergrafos direcionados que possuam fluxo.

Para resolver este problema pode-se fazer uso da implementação de um algoritmo de busca em profundidade de modo que como resultado da busca identifiquem-se as hiperarestas componentes do fluxo acendente do ciclo. Desta maneira, identificados as hiperarestas, marcam-se os mesmos.

A partir da marcação, assume-se que estas hiperarestas estão em sentido contrário, executa-se o algoritmo de traçado e depois inverte-se as representações de setas das hiperarestas marcadas.

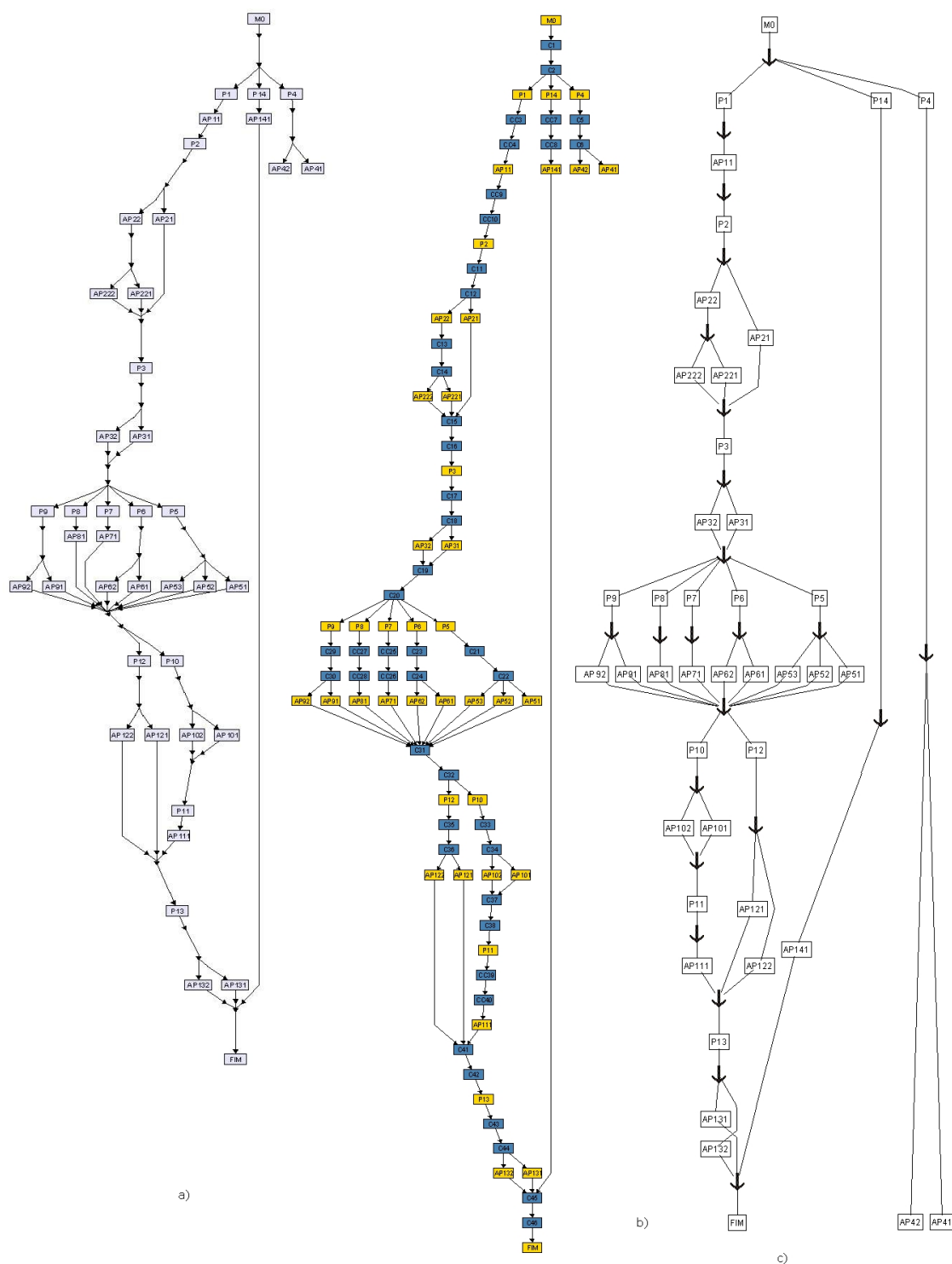


Figura 5.6: Resultado Obtido com as Alterações do VGJ

CAPÍTULO 6

CONCLUSÃO

Neste trabalho prezou-se pela utilização de algoritmos consagrados de traçado automático de grafos para o estudo sobre o traçado de hipergrafos direcionados. O que tornou esta abordagem possível foi a utilização da transformação do hipergrafo direcionado em grafo direcionado segundo Guedes [10] [11]. Com o resultado desta transformação fez-se possível a utilização de programas prontos de desenho de grafos, a promoção de testes com diversos algoritmos e padrões diferentes de traçado, a análise dos resultados práticos e a escolha da forma de traçado mais adequada para o problema em questão. Desta forma as comparações entre as formas de representação ocorreram com uma mínima implementação.

Outros pesquisadores que desejarem estudar o desenho de hipergrafos direcionados podem utilizar-se da mesma abordagem de aproveitamento de representação sobre um grafo direcionado e assim aproveitar toda a gama de algoritmos desenvolvidos para tal fim, promover os testes, comparações e análises com grande facilidade.

O software desenvolvido sobre o aplicativo VGJ[18] permitiu o aproveitamento do algoritmo de traçado hierárquico pré-implementado para traçar automaticamente hipergrafos direcionados. Por ter sido desenvolvido sobre tecnologia Java e com a possibilidade de ser aplicado sobre uma página de internet há um grande poder de adaptação a ser explorado. Por exemplo, qualquer software de gerência de projetos que deseje fazer uso da ferramenta desenvolvida para agregar visualização do fluxo de atividades dos projetos, pode facilmente acoplar o mesmo simplesmente construindo uma pequena interface com tal finalidade.

Após feita a integração pode-se ainda com muita facilidade construir uma interface para a visualização, configuração e cadastro de características inerentes as atividades, aumentando assim o poder da ferramenta. Tem-se, desta forma, o poder da representação gráfica agregado a fácil visualização e manutenção das características das atividades re-

lacionadas.

Em alguns casos notou-se que o algoritmo de traçado utilizado não conseguiu eliminar totalmente os cruzamentos entre as arestas para os grafos planares. Esta característica deve ser evitada para melhorar a qualidade do desenho gerado.

6.1 Sugestões de Trabalhos Futuros

Devido a importância da representação de ciclos em outras aplicações de grafos, tem-se como um trabalho futuro promover a alteração na ferramenta desenvolvida com o objetivo de permitir o traçado de hipergrafos direcionados que contenham ciclos. Desta forma a ferramenta desenvolvida torna-se mais genérica e poderá ser utilizada com outros objetivos diferentes do abordado neste trabalho.

Outro ponto que pode contribuir para o aumento da qualidade do traçado gerado é o estudo e melhoria no algoritmo de traçado de forma que a minimização dos cruzamentos entre as arestas seja melhorada principalmente para os casos dos grafos planares.

BIBLIOGRAFIA

- [1] Giorgio Ausiello, Alessandro D'Atri, e Domenico Saccà. Graph algorithms for functional dependency manipulations. *Journal of the ACM*, 30(4):752–766, outubro de 1983.
- [2] G. Di Battista, P. Eades, R. Tamassia, e I. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry: Theory and Applications*, (4):235–282, 1994.
- [3] Microsoft Corporation. Microsoft Project 2003. <http://www.microsoft.com/brasil/office/project/default.asp>, acessado em 14/04/2005.
- [4] Companhia Paranaense de Energia. Companhia Paranaense de Energia. <http://www.copel.com/>, acessado em 20/10/2005.
- [5] Graph Drawing. GD. <http://www.graphdrawing.org/>, acessado em 30/09/2005.
- [6] Free Software Foundation. GNU GENERAL PUBLIC LICENSE . http://www.eng.auburn.edu/department/cse/research/graph_drawing/COPYING, acessado em 04/04/2005.
- [7] Thomas M. J. Fruchterman e Edward M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.
- [8] Ashim Garg e Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2002.
- [9] Giorgio Gallo, Giustino Longo, Sang Nguyen, e Stefano Pallotino. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(177-201):3–60, 1993.
- [10] André Luiz Pires Guedes. *Hipergrafos Direcionados*. Thesis (D.Sc.), Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2001.

- [11] André Luiz Pires Guedes e Lilian Markenzon. Directed hypergraph planarity. *Pesquisa Operacional*, 25(3):383–390, setembro/dezembro de 2005.
- [12] David Harel. On visual formalisms. *Communications of the ACM*, 31(5):514–530, 1988.
- [13] S. Tagawa K. Sugiyama e M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, 11(2):109–125, 1981.
- [14] Oreas Automatic Diagram Layouts. Oreas - Automatic Diagram Layouts. <http://www.oreas.com/>, acessado em 04/04/2005.
- [15] Marly Monteiro de Carvalho e Roque Rabechini Jr. *Construindo competências para gerenciar projetos: teoria e casos*. Atlas, 1th edition, 2005.
- [16] Michael Kaufmann and Dorothea Wagner. *Drawing Graphs: Methods and Models*. Springer-Verlag, 1th edition, 2001.
- [17] Marcus Raitner. The GML File Format. <http://infosun.fmi.uni-passau.de/Graphlet/GML/>, acessado em 20/10/2004.
- [18] Computer Science e Software Engineering. Drawing Graphs with VGJ. http://www.eng.auburn.edu/departament/cse/research/graph_drawing/vgj.html, acessado em 30/03/2005.

ZENO STIVANIN

**TRAÇADO AUTOMÁTICO DE HIPERGRAFOS
DIRECIONADOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. André Luiz Pires Guedes

CURITIBA

2006