

Linguística Computacional Interativa

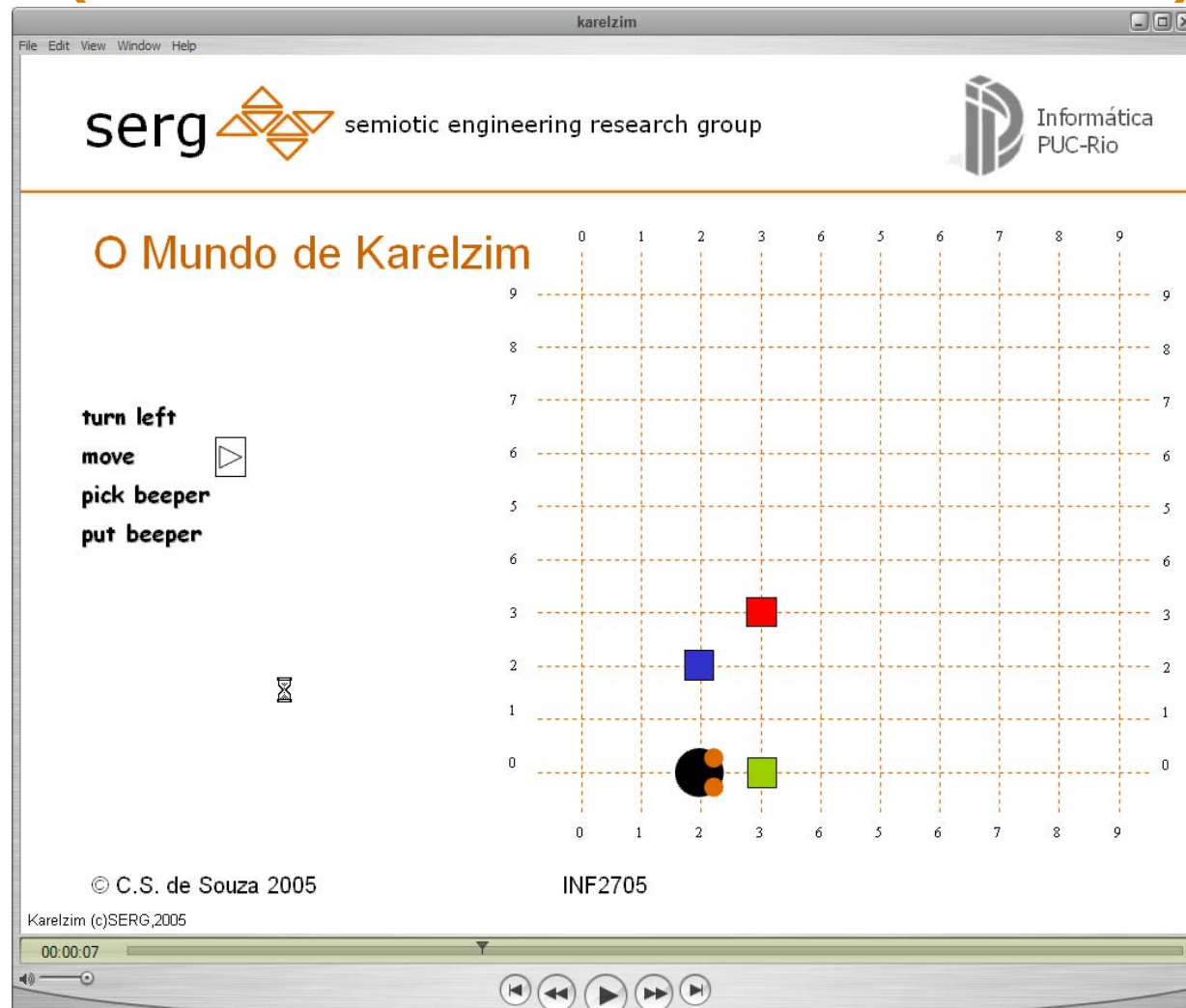
Diálogos Explicativos e Descrições Estáticas (Estruturais, Funcionais, etc.)

Aula de 30 de outubro de 2012

Perguntas e Respostas Explicativas

- Perguntas “Qu_?”
 - Quem?
 - Qual?
 - Quando?
 - Onde?
 - Como?
 - Por quê?
 - Quanto?
- Pergunta “E se?”
- Perguntas “Sim ou Não?”

Karelzim (Baseado nos livros de Richard Pattis)



```

1 ?- start.
true.

2 ?- where(karel).
Karel is at corner (loc(0,0)), facing ori(a).
true.

3 ?- where(beepers).
The beepers in this world are located at the following places:
[[1.9],[2.1],[2.3],[2.9],[3.9],[4.9],[5.9],[7.3],[8.2]]
true.

4 ?- where(walls).
The walls in this world are located at the following corners:
[[0.2],[0.3],[0.4],[1.2],[1.4],[2.2],[2.8],[3.2],[3.4],[3.8],[4.2],[4.3],[4.4],[4.8],[5.8],[6.0],[6.1],[6.2],[6.8],[7.0],[7.8],[8.0],[9.0],[9.1],[9.2]]
true.

5 ?- howmany(beepers,Place).
Karel has 1 beeper(s) in his bag.
Place = bag ;
At this moment, 9 beeper(s).
Place = world.

6 ?- move,move,move,move.
true.

7 ?- where(karel).
Karel is at corner (loc(4,0)), facing ori(a).
true.

8 ?- what_if.
true.

9 ?- move,move,move.
false.

10 ?- why_not.
Because Karel is facing a wall.
true.

11 ?- where(karel).
Karel is at corner (loc(5,0)), facing ori(a).
true.

12 ?- ok.
true.

13 ?- where(karel).
Karel is at corner (loc(4,0)), facing ori(a).
true.

```

Programação via Interface

- Tese de Doutorado de S.D.J. Barbosa (1999)
 - Interpretador de Metáforas e Metonímias para estender uma aplicação (*end user programming*)
 - Referência listada no website do SERG:



Knowledge-Based Systems 14 (2001) 15–27

Knowledge-Based
SYSTEMS

www.elsevier.com/locate/knosys

Extending software through metaphors and metonymies

S.D.J. Barbosa*, C.S. de Souza¹

Computer Science Department, Pontifícia Universidade Católica, R. Marquês de São Vicente, 225 — Gávea, Rio de Janeiro, RJ 22453-900, Brazil

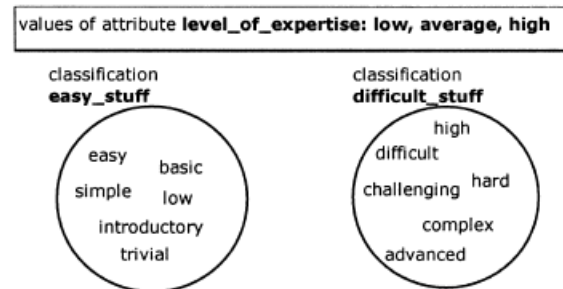


Fig. 2. Classifications to be used for generating metaphors.

order to obtain more sophisticated metaphorical utterances. Fig. 2 presents two sample classifications in this domain.

Combining the static and dynamic models, on the one side, with the classifications provided by designers, on the other, we now turn to the examples that illustrate metaphorical and metonymic expressions that can be interpreted.

Example 1 (Creating an entity). Our first example creates a new entity based on an existing one. Let us consider the following utterance: “A binder is a group of documents”. It could be rewritten as the following predicate:

```
analyze_sentence(binder, is, group, of,
document)
```

In this case, since *groups* is an entity and *binder* does not

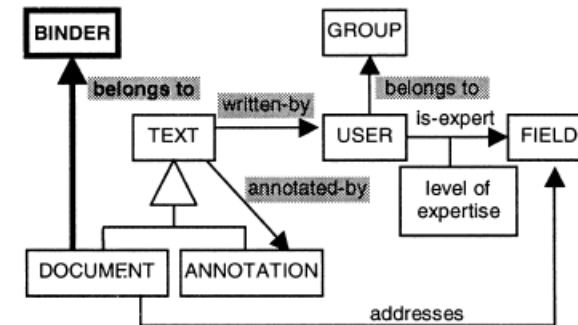


Fig. 4. Model extended through a metaphor.

documents, and execute the corresponding operation for each document.

Example 2 (Creating an entity with an additional attribute). Our next example considers the utterance: “An evaluation is an annotation with a grade”. This may be considered an analogy, and rewritten as:

```
analyze_sentence(evaluation, is, annotation, with, grade)
```

We cannot a priori decide whether *grade* should be considered an attribute of the new entity *evaluation* or another entity which is part of it. Therefore, the user should be presented with the alternative options and be asked to choose one of them:

GEMA: Geração Automática de Textos Descritivos em Português do Brasil

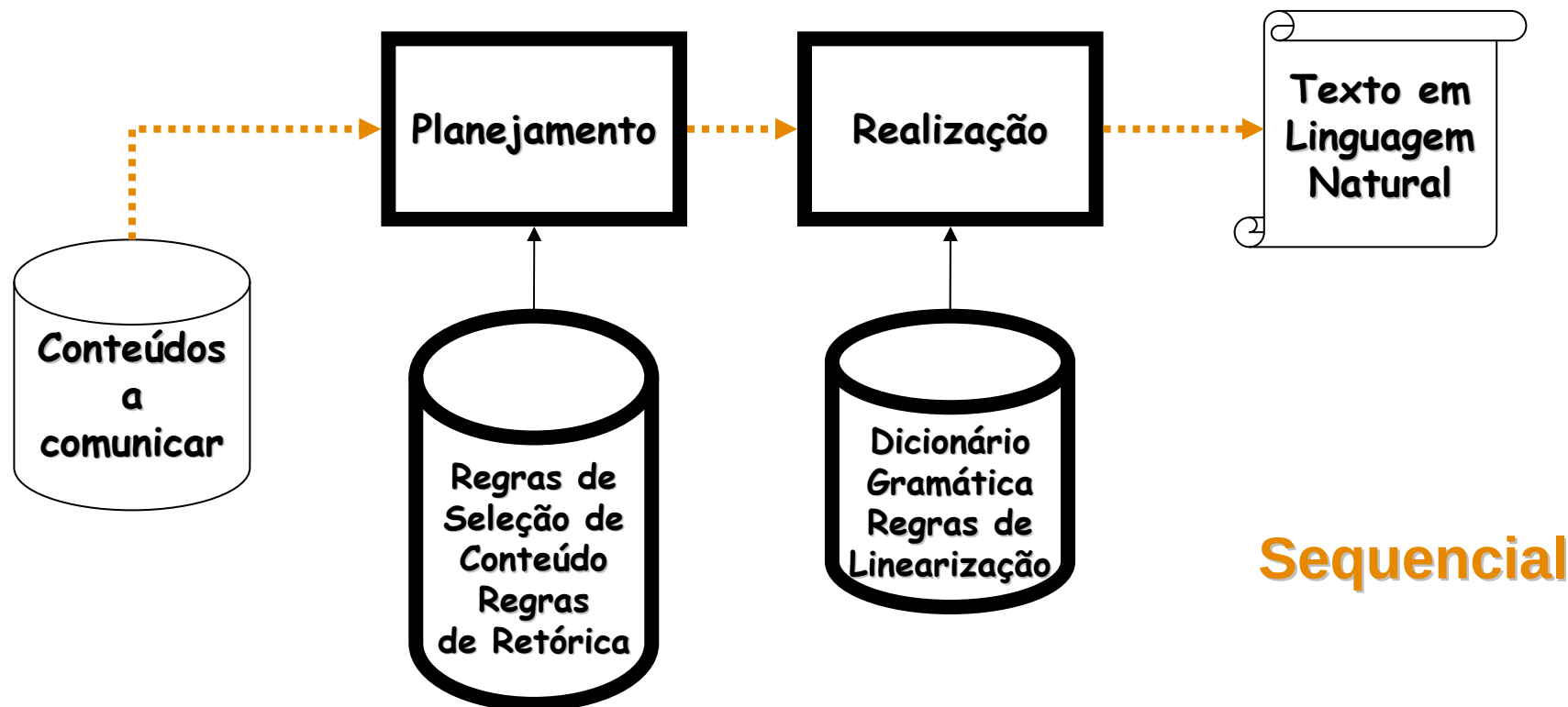
Referências:

1. Oliveira, Lucia L.G. (1991) *Um realizador para o gerador automático de texto descritivo do Gema*. Dissertação de Mestrado. Departamento de Informática da PUC-Rio. Setembro de 1991. Orientadora: Clarisse Sieckenius de Souza
http://www2.dbd.puc-rio.br/pergamum/biblioteca/php/mostrateses.php?arqtese=1991-OLIVEIRA_L_L_G_D.pdf
2. Scott, D.R.; de Souza, C.S. (1988) Conciliatory planning for extended descriptive texts. *Monografias em Ciência da Computação* MCC13/88. Departamento de Informática, PUC-Rio.
ftp://ftp.inf.puc-rio.br/pub/docs/techreports/88_13_scott.pdf

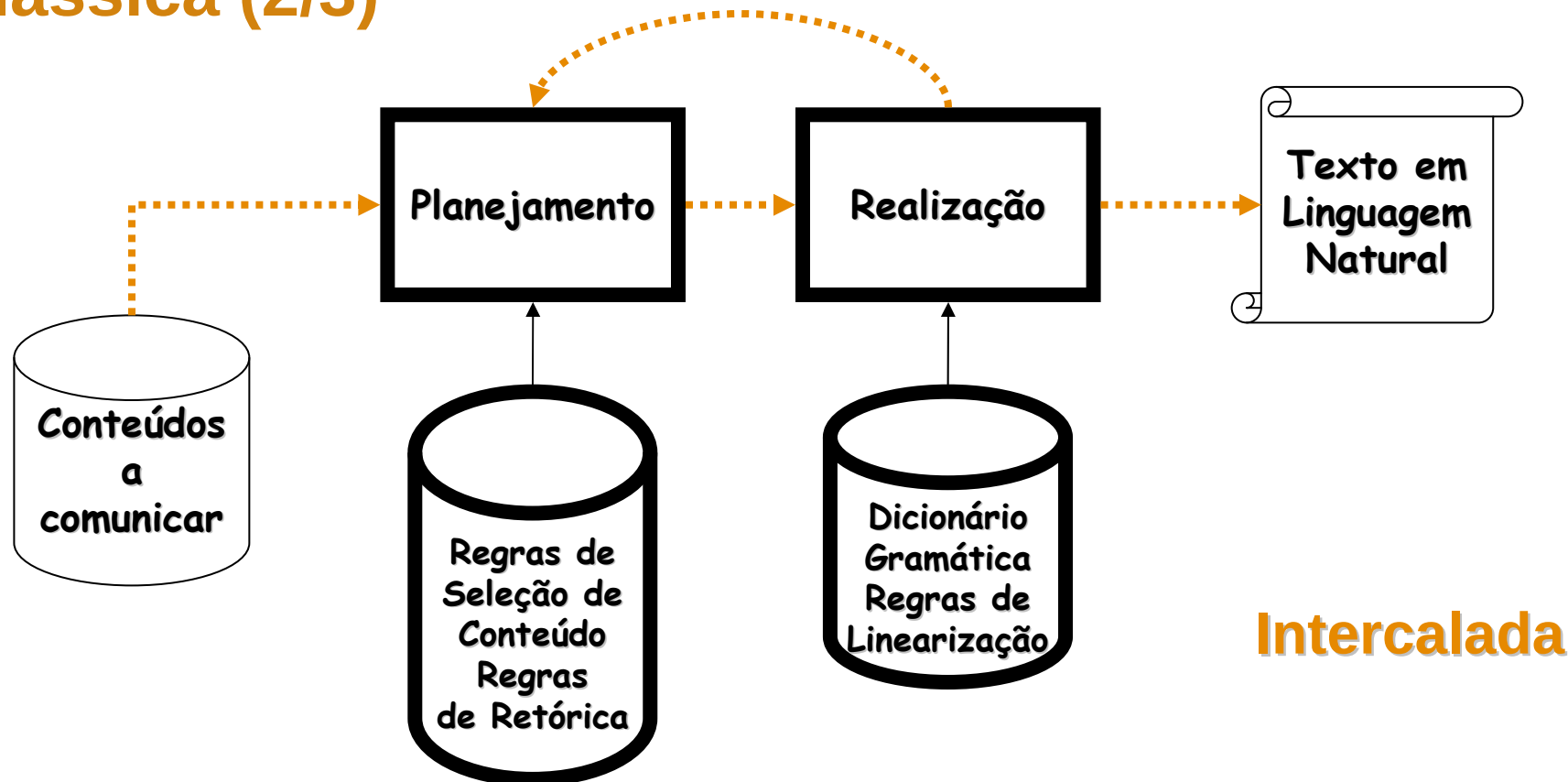
Geração de Textos: Planejamento e Realização

- Planejamento:
 - O que dizer no texto (seleção de informação)
 - Como dizê-lo (organização da informação)
- Realização:
 - Dizer (conversão de representações internas em “texto em linguagem natural”, também chamada de *linearização*)

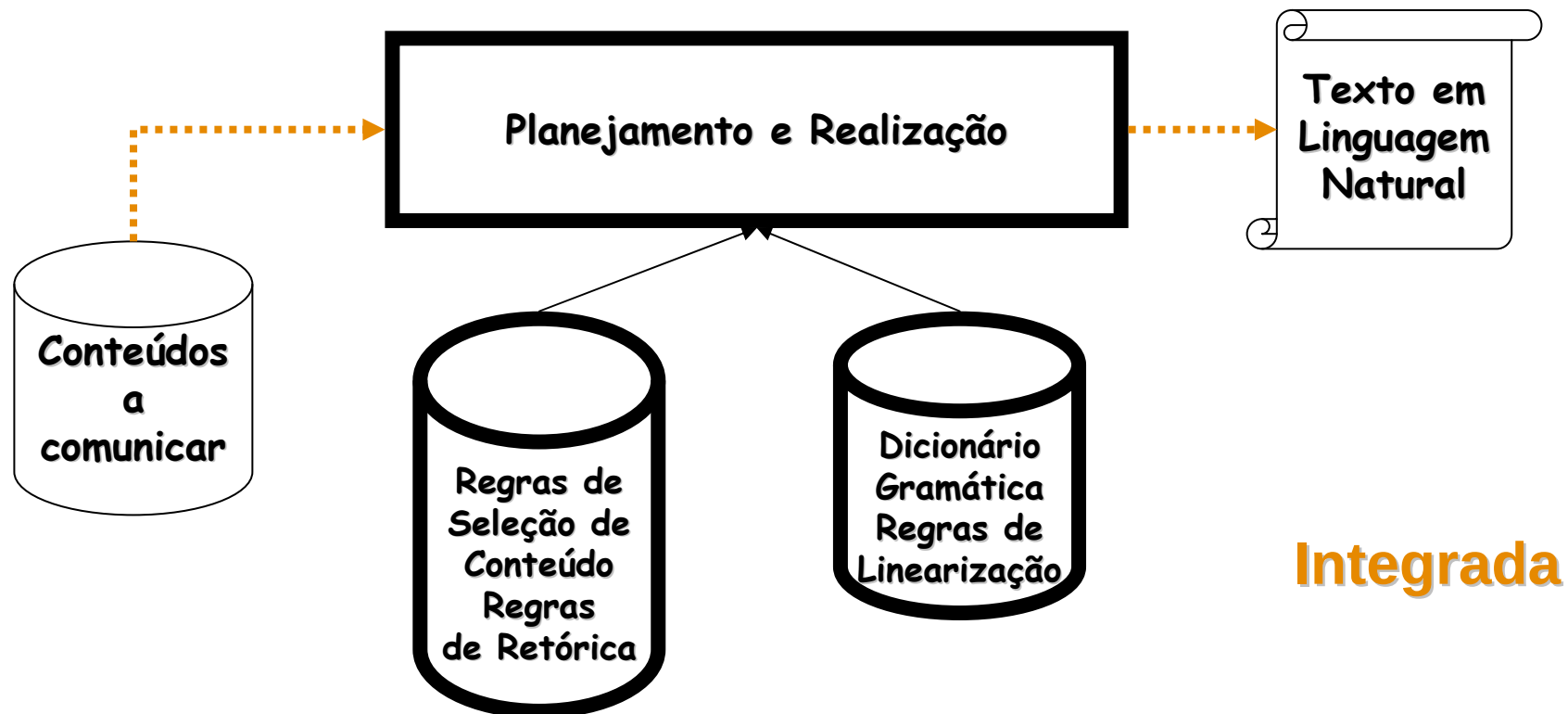
Arquiteturas Alternativas em Geração de Texto Clássica (1/3)



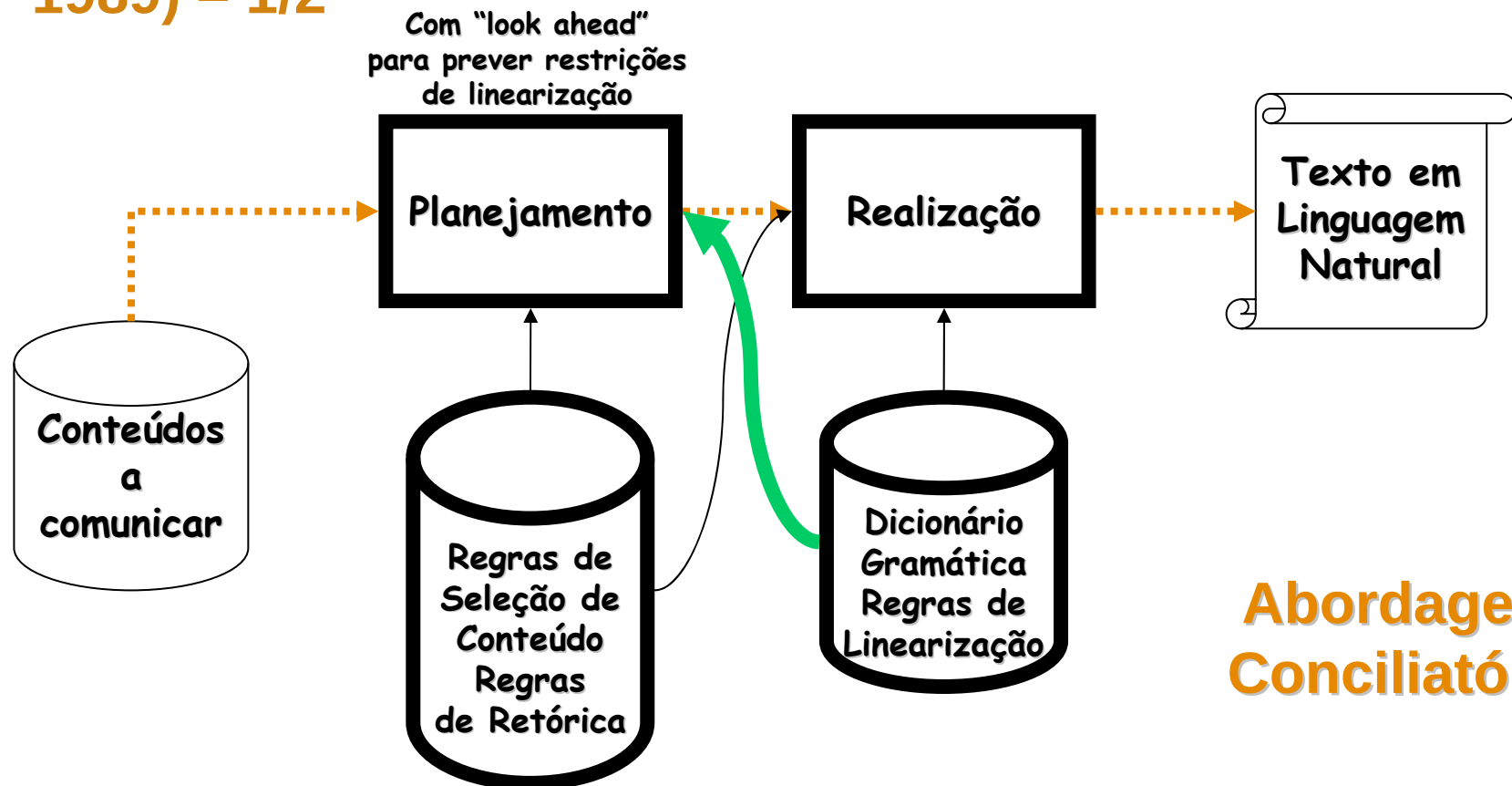
Arquiteturas Alternativas em Geração de Texto Clássica (2/3)



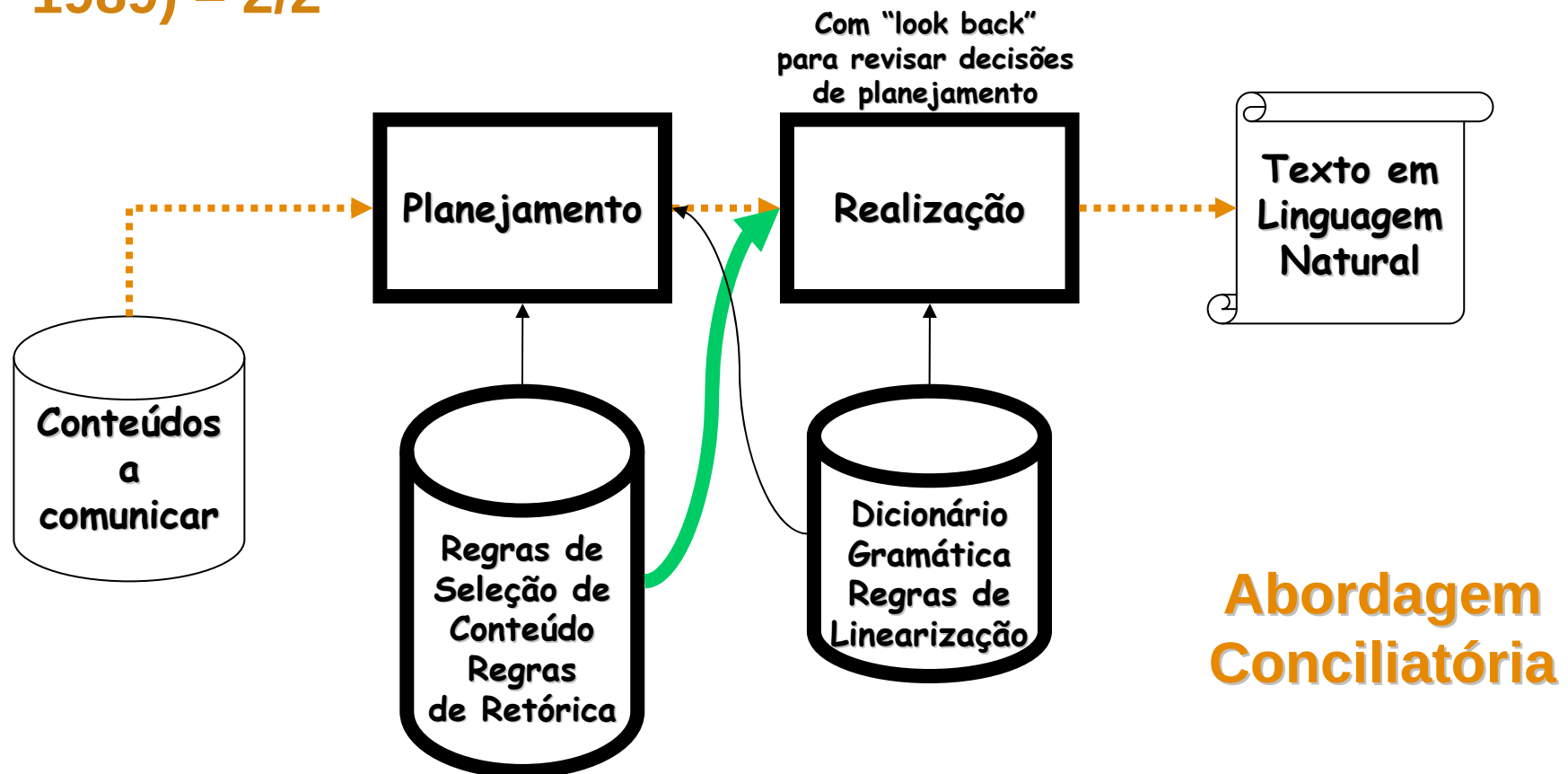
Arquiteturas Alternativas em Geração de Texto Clássica (3/3)



O “planejamento conciliatório” (Scott & de Souza, 1988; 1989) – 1/2



O “planejamento conciliatório” (Scott & de Souza, 1988; 1989) – 2/2



**Abordagem
Conciliatória**

O GEMA

- Gerador Automático de **Textos Descritivos Extensos**
 - Texto Descritivo: não trata de temporalidade (como seria o caso de textos *narrativos*), embora possa tratar de causalidade (o que tem implicações para a marcação de *precedência* de informação no texto gerado)
 - Texto Extenso: contém múltiplos parágrafos, cada um deles com múltiplas sentenças
- Domínio de Aplicação Selecionado
 - Descrição de Modelos de Fluxos de Dados representados por DFD's (*data flow diagrams*)

Representação Interna de DFD's – Primitivas Semânticas

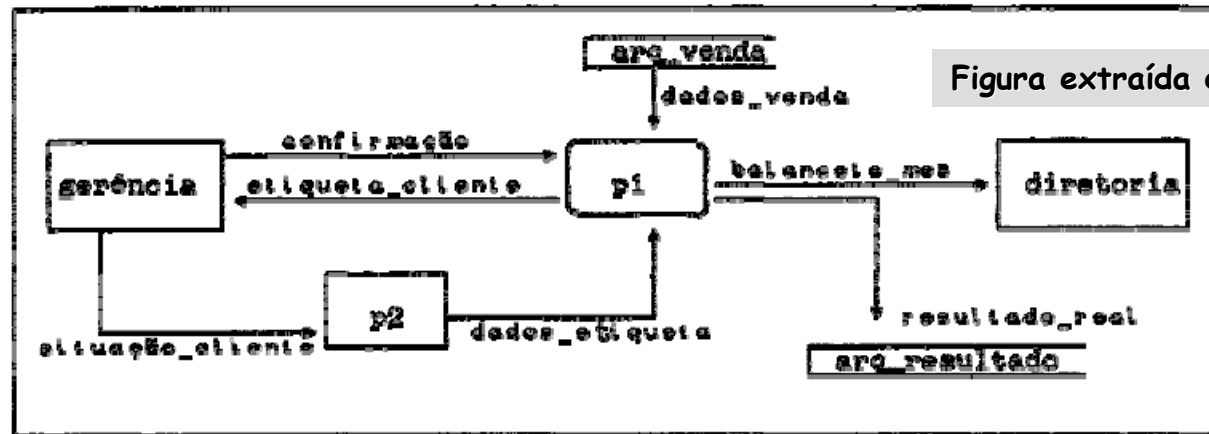


Figura 8: Fragmento de um DFD.

Representações Simbólicas de DFD's (gráficos)

- do-acts: Ação geral do Processo
- r-acts: ações de recebimento de dados
- p-acts: processamento específico de fluxos de dados
- s-acts: ações de envio de dados

Exemplos referentes à figura acima

- do-act (p1, 'emitir', 'relatório gerencial')
- r-act (LER, p1, 'dados_venda', 'arq_venda')
- r-act (RECEBER, p1, 'confirmação', 'gerência')
- p-act (COMPOR, p1, 'confirmação', 'balancete_mês')
- p-act (COMPOR, p1, 'dados_venda', 'balancete_mês')
- s-act (ENVIAR, p1, 'balancete_mês', 'diretoria')

Alinhamento da estrutura de representação

predicado (argumento₁, argumento₂, ..., argumento_n)

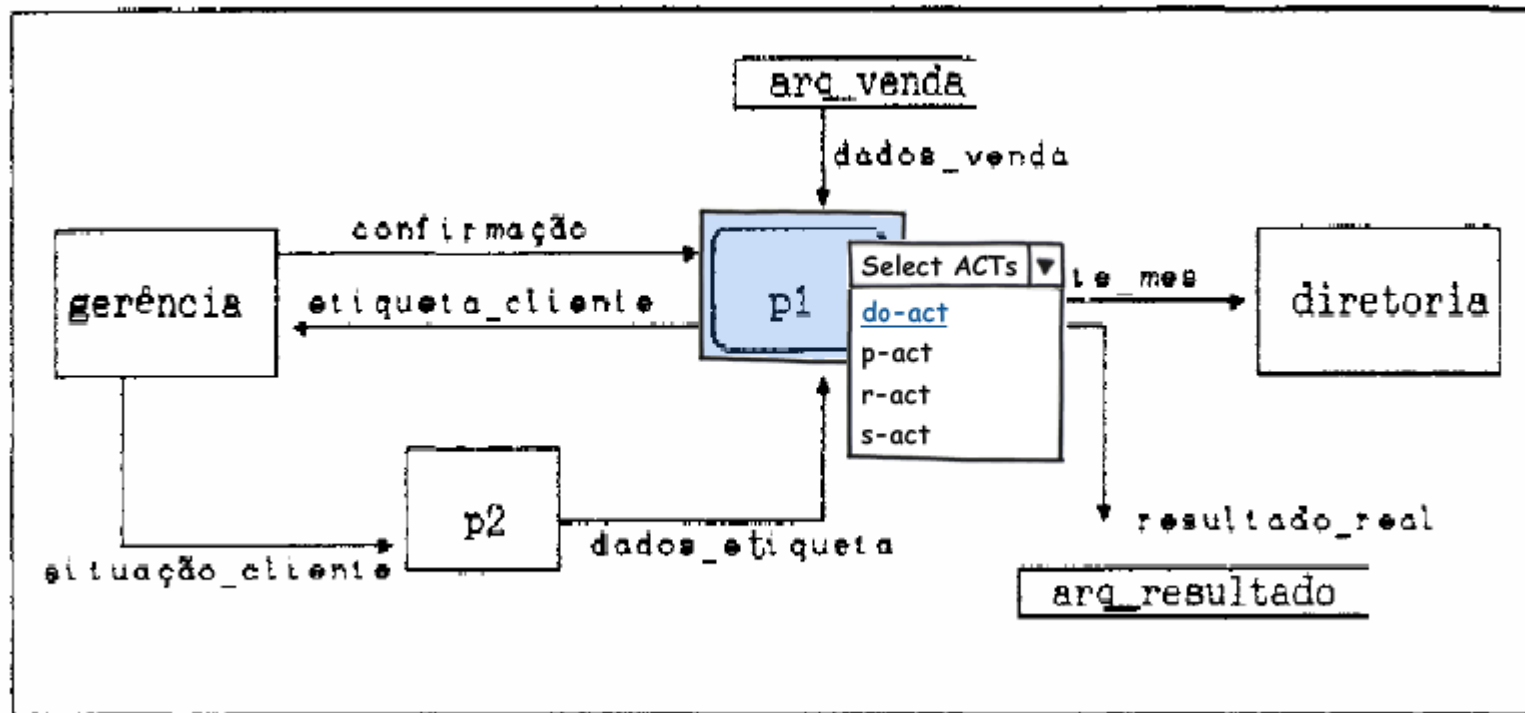
- do-acts:
 - predicado 'do-act'
 - argumentos (NOME-DO-PROCESSO(agente), VERBO(ação), COMPLEMENTO(objetos e demais complementos)
 - Fontes da Informação para o Gerador:
 - Nome do Processo: Marcação no DFD
 - Verbo: Fornecido interativamente pelo usuário na forma padrão
 - Complemento: Fornecido interativamente pelo usuário na forma padrão
- r-acts e s-acts:
 - predicados: 'r-act', 's-act'
 - argumentos (VERBO-PADRÃO(ação), NOME-DO-PROCESSO(agente), DADO(fluxo IN/OUT), COMPLEMENTO (ORIGEM/DESTINO))
 - Fontes da Informação para o Gerador:
 - Verbo-Padrão: Selecionado interativamente pelo usuário dentre elementos de conjunto definido
 - Nome do Processo: Marcação no DFD
 - Dado: Marcação no DFD
 - Complemento: Marcação no DFD

Alinhamento da estrutura de representação (cont.)

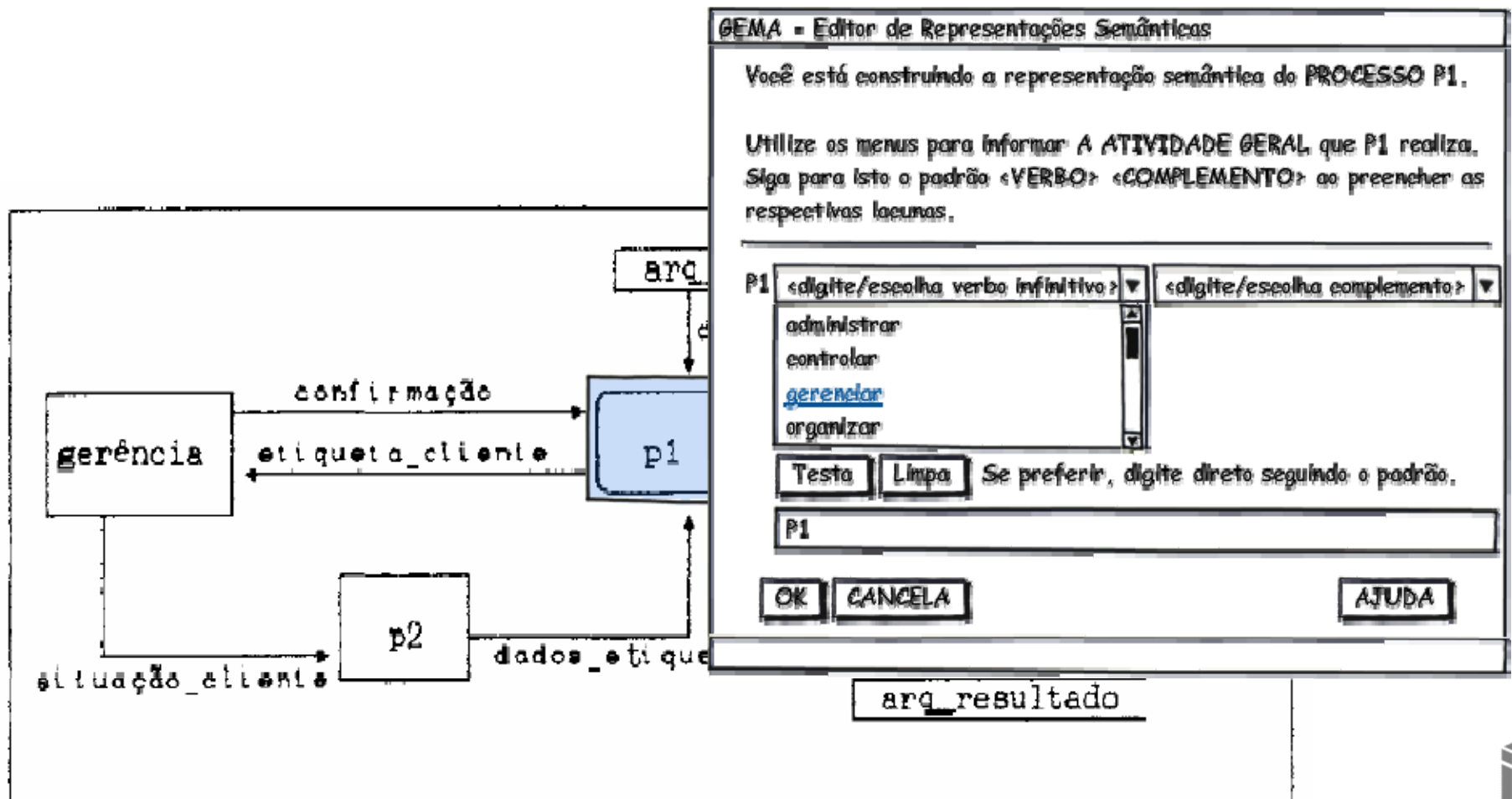
predicado (argumento₁, argumento₂, ..., argumento_n)...

- r-acts e s-acts:...
- p-acts:
 - predicado: 'p-act'
 - argumentos (VERBO-PADRÃO(ação), NOME-DO-PROCESSO(agente), DADO-IN, DADO-OUT)
 - Fontes da Informação para o Gerador:
 - Verbo-Padrão: Selecionado interativamente pelo usuário dentre elementos de conjunto definido
 - Nome do Processo: Marcação no DFD
 - Dado-IN: Indicado pelo usuário sobre fluxo marcado no DFD
 - Dado-OUT: Indicado pelo usuário sobre fluxo marcado no DFD

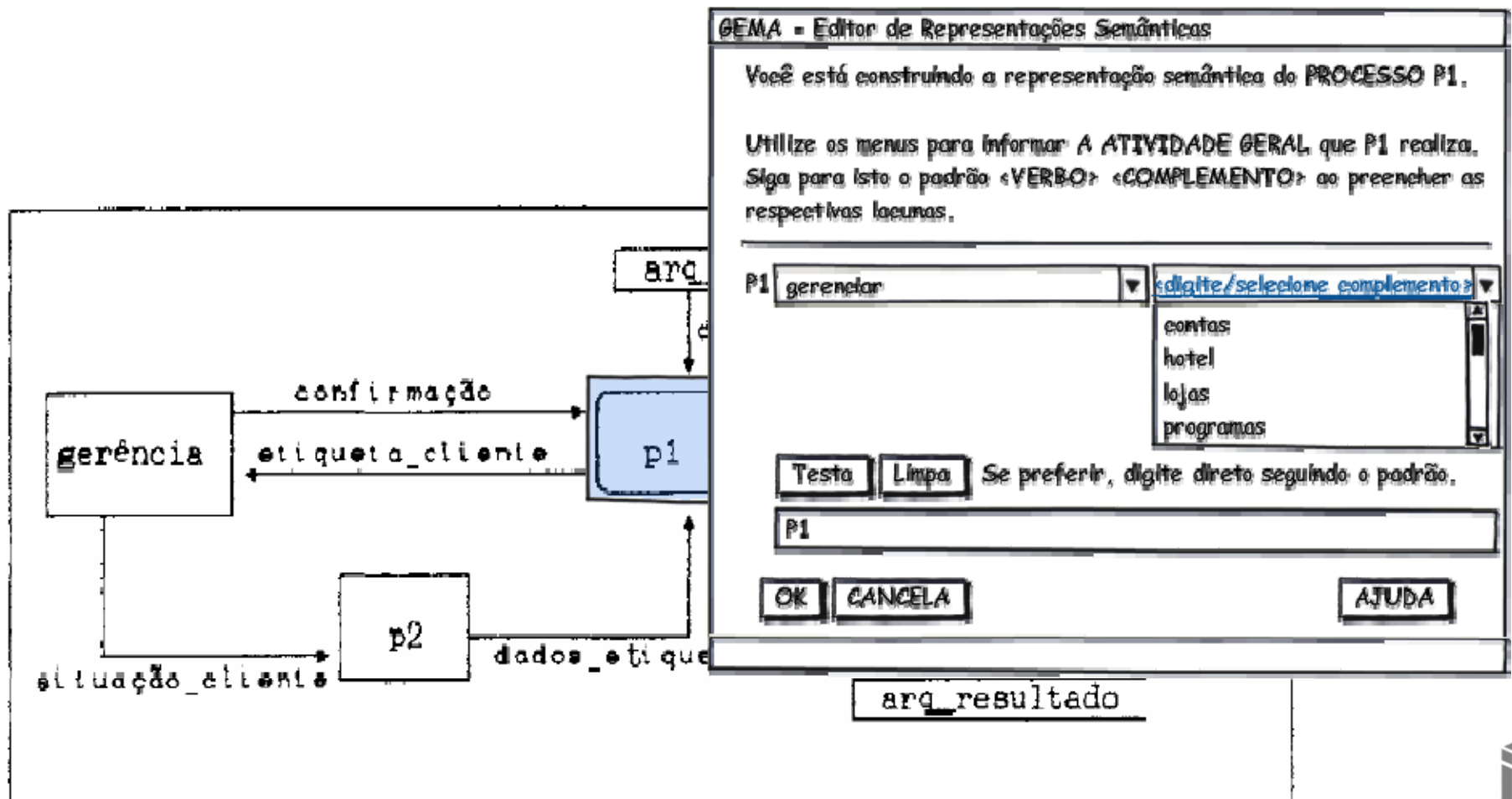
Sketch de interface para captura de representação da dados



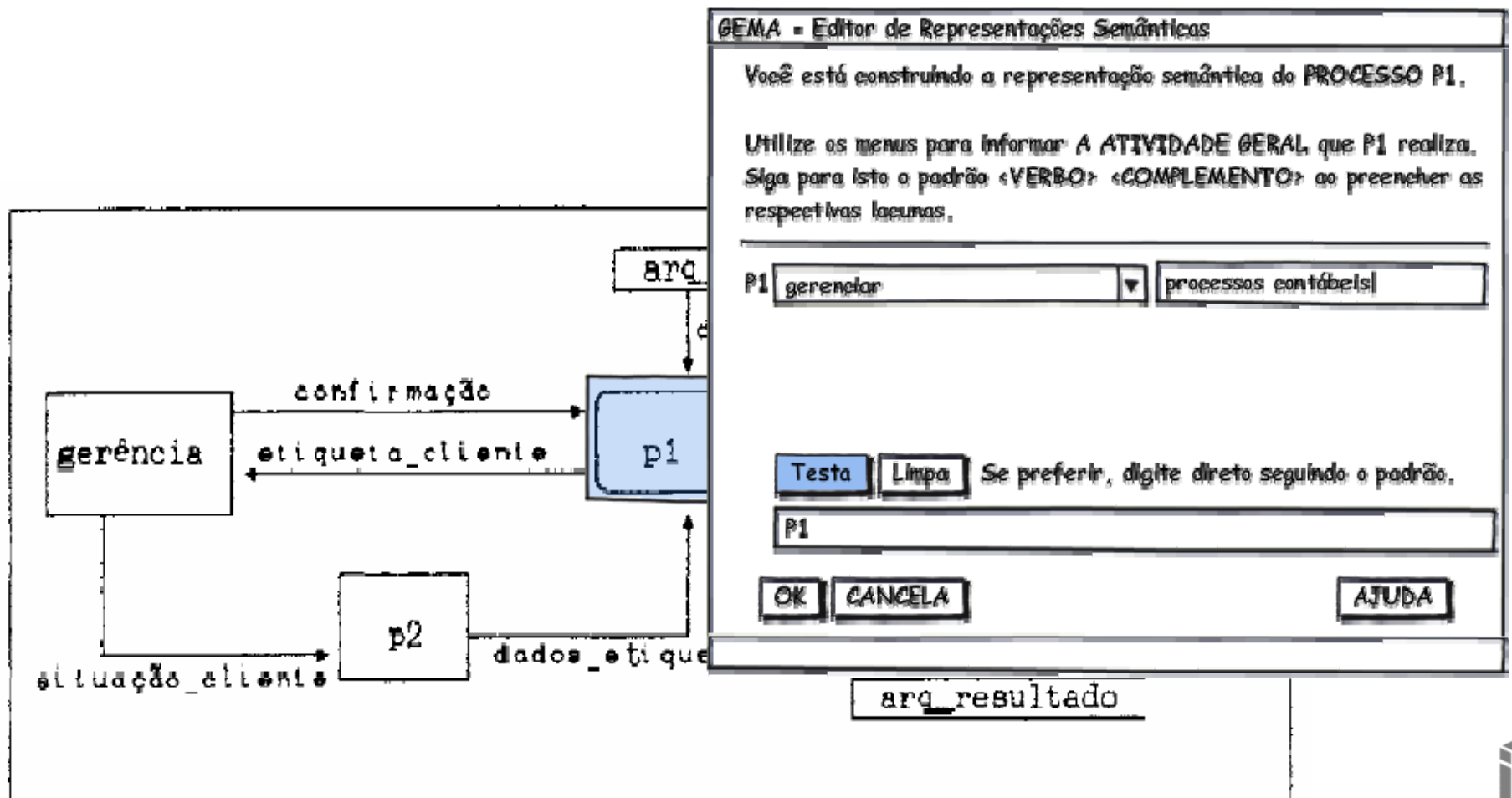
Sketch de interface para captura de representação da dados



Sketch de interface para captura de representação da dados



Sketch de interface para captura de representação da dados



Sketch de interface para captura de representação da dados

