

Pontifícia Universidade Católica do Paraná
Programa de Pós-Graduação em Informática Aplicada

A Realidade Construída – I

Dissertação apresentada à Pontifícia Universidade
Católica do Paraná para obtenção do título de
Mestre em Ciências

Curitiba – Paraná

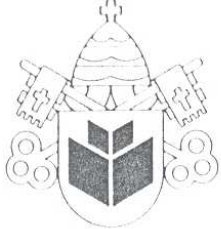
Julho de 2001

Dedicatória

À minha querida Gilda,
Com todo o meu amor,
E com todo o meu agradecimento,
Pois em ti encontrei apoio
Quando tudo parecia cair ao meu redor.

Aos meus filhos

Lilian,
Daniel,
Juliana,
Também com meu amor,
Que, espero, só possa aumentar.



ATA DA SESSÃO PÚBLICA DE DEFESA DE DISSERTAÇÃO DE MESTRADO
DO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA
DA PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

DEFESA DE DISSERTAÇÃO Nº 034

Aos 31 dias do mês de julho de 2001 realizou-se a sessão pública de defesa da dissertação "**Realidade Construída I**", apresentada por **Orlando Alcântara Soares** como requisito parcial para a obtenção do título de **Mestre em Ciências**, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Alcides Calsavara
PUCPR (Presidente)

Alcides Calsavara
assinatura

Aprovado sem restrições
parecer

Prof. Dr. Edson Scalabrin
PUCPR

Edson Scalabrin

Aprovado sem restrições

Profa. Dra. Itana Gimenes
UEM

Itana Gimenes

Aprovado sem restrições

Prof. Dr. Ricardo Gudwin
UNICAMP

Ricardo Gudwin

Aprovado sem restrições

Prof. Dr. Marcos Shmeil
PUCPR (suplente)

Marcos Shmeil

Aprovado sem restrições

Conforme as normas regimentais do PPGIA e da PUCPR, o trabalho apresentado foi considerado APROVADO SEM RESTRIÇÕES (aprovado sem restrições, aprovado com exigências ou reprovado), segundo a avaliação da maioria dos membros da Banca Examinadora acima indicada.

Observações, exigências e/ou restrições da Banca Examinadora, quando houver:

Devem ser complementados os capítulos 17 e 18 a fim de tornar explícito

como os conceitos apresentados são usados nos exemplos de aplicação.
continuar no verso, se necessário

Carlos Alberto Maziero
Prof. Dr. Carlos Alberto Maziero, Diretor
Programa de Pós-Graduação em Informática Aplicada
Pontifícia Universidade Católica do Paraná

Agradecimentos

Aos meus amigos (em ordem alfabética):

Prof. Ailton Renato Dörl

– a quem minhas próprias palavras não podem agradecer e que, agindo de acordo com sua crença, faz o bem;

Prof. Dr. Alcides Calsavara

– quem, com companheirismo e sensibilidade, enxerga muito longe e, assim, proporcionou a orientação segura que levou este trabalho a bom termo;

Prof. Dr. Braulio Ávila

– cuja disposição ímpar em ajudar é um exemplo a ser seguido;

Prof. Dr. Edson Emílio Scalabrin

– cujo finíssimo senso e extraordinária generosidade constituem um privilégio para todos os que lhe são próximos;

Prof. Dr. Marcos Hochuli Shmeil

– primeiramente meu aluno, hoje meu irmão, cuja capacidade de compreender é uma inspiração para todos.

Conteúdo

Resumo.....	11
Abstract.....	13
Tema.....	15
1 Introdução	16
Parte I – O Problema	19
2 Dificuldades dos Formalismos de Representação	20
2.1 Inviabilidade da Modelagem Monoplana	20
2.2 A Pesquisa.....	21
2.3 Delimitação do Tema.....	23
2.4 A Proposta.....	25
Parte II – Epistemologia	27
3 Significado	28
4 Teoria do Signo de Peirce.....	31
4.1 Signo, Objeto e Interpretante	31
4.2 Conceito de Semiose.....	34
4.3 Aplicação do Conceito de Semiose	36
4.4 Implicações da Teoria do Signo.....	38
Parte III – Ontologia.....	40

5	A Estrutura.....	41
5.1	Conceito de Objeto	46
5.2	Representação de Objeto	50
6	O Modelo	58
6.1	O que é um Modelo	58
6.2	Modelo e Realidade	62
6.3	Crítica: Modelo, Ontologia e Base de Conhecimento.	64
	Parte IV – Fenomenologia.....	68
7	Predicado	69
7.1	Conceito Tradicional de Predicado.....	69
7.2	Conceito Lógico de Predicado.....	71
7.3	Conceito Semântico de Predicado	74
8	Interpretação de um Predicado.....	78
9	Exemplo de Aplicação de Predicado e Papéis Semânticos.....	84
9.1	Venda	86
	Parte V – Ética	88
	Estado.....	89
10	Noção Tradicional de Estado.....	90
11	Conceito Formal de Estado.....	94
11.1	Base Intuitiva	94

11.2	Observações sobre o Conceito	96
11.3	Representação Gráfica	96
11.4	Aspectos Semânticos do Conceito de Estado	98
12	Definição Formal de Estado.....	102
12.1	Estado: Formalização em Eiffel.....	105
12.2	Formalização em Object Pascal	108
12.3	Exemplo de Aplicação do Conceito de Estado	112
	Evento.....	113
13	Noção Tradicional de Evento.....	114
14	Definição Formal de Evento.....	116
14.1	Especificação Formal de Evento com Implementações	117
15	Consequências das Noções Tradicionais.....	119
16	Implicações dos Conceitos de Estados e Eventos	122
	Parte VI – Pragmática	125
17	Aplicação de Evento – Análise de Estrutura Organizacional.....	126
17.1	Função_SEBRAE	128
18	Aplicação de Estados e Eventos – Rede Neural	132
18.1	Topologia de Redes Neurais – Diagrama	133
18.2	Topologias de Redes Neurais – Classes	134
18.3	Arquiteturas de Redes Neurais	135

18.4	Perceptron	137
18.5	Neurônio	140
19	Desenvolvimento Futuro	144
20	Conclusão.....	148
21	Bibliografia	153

Lista de Figuras

Figura 1: Tríade de Peirce.....	33
Figura 2: Classe CIDADAO	54
Figura 3: Modelo e Semiose	59
Figura 4: Sentença tradicional	70
Figura 5: Predicado lógico.....	72
Figura 6: Predicado e seus componentes	74
Figura 7: Predicado e papéis semânticos	75
Figura 8: Classe, atributos e operações.....	80
Figura 9: Interpretação de um predicado	82
Figura 10: Aplicação do conceito de predicado.....	85
Figura 11: Estado - Representação gráfica	95
Figura 12: Estado - Representação gráfica completa.....	97
Figura 13: Aplicação do conceito de estado	112
Figura 14: Domínio - Níveis Lógicos de Representação.....	123
Figura 15: Estados e hierarquia de funções	131
Figura 16: Topologia de redes neurais.....	133

Resumo

Todo sistema de informação é uma representação de determinadas idéias; um sistema em computador corresponde à representação que se torna operacional sob a forma de software. A modelagem conceitual dos sistemas de informação situa-se na base do processo de criação (e do posterior funcionamento) de tais sistemas: seu objetivo é a formalização dos conceitos que determinados grupos de pessoas utilizam para realizar suas metas; ou seja, idealmente é a **formalização do conhecimento** que as pessoas possuem sobre ambientes e assuntos. Daí resulta a importância das técnicas de modelagem conceitual e coloca-se a questão fundamental: as técnicas de modelagem, para atingir seu potencial de utilização, precisam ser encaradas, em primeiro lugar, como um problema de representação de conhecimento. Isto coloca questões comumente não consideradas pelas técnicas usuais de especificação de sistemas. Muito da dificuldade de representação de conhecimento resulta de um problema bastante profundo; as metodologias de modelagem usualmente não separam dois aspectos fundamentais: (i) representação dos objetos básicos de um domínio conceitual, e (ii) representação de políticas – ou regras de operação – sobre esses objetos básicos – como resultado, níveis lógicos diferentes são representados no *mesmo* plano de modelagem, levando a soluções de grande complexidade, além de teoricamente inviáveis, como pretendemos mostrar neste trabalho.

Não apenas na informática, mas em qualquer sistema semiótico (aí também a linguagem natural), a compreensão dos níveis de representação baseia-se no adequado entendimento dos conceitos de estados e eventos. O tratamento dado a estado, na literatura técnica, tem sido muito influenciado pela abordagem behaviorista, centrada na análise do *comportamento observável* de um objeto que recebe estímulos. Colocar o foco da análise no comportamento individual de objetos faz com que os estados sejam considerados também do ponto de vista de objetos isolados, quando, na realidade, a caracterização de um determinado estado envolve a participação conjugada de vários objetos simultaneamente. A falta de formalização (como, também, a incompreensão) do conceito de estado levou a que, para modelar estados e eventos, sejam utilizados três tipos de modelos, produzindo, como consequência, representações de grande complexidade e dificuldade de adaptação. Se adotarmos a visão tradicional de evento (instantâneo), os eventos só estarão no modelo “por suas manifestações”, sem serem modelados diretamente. A visão de um evento como uma ocorrência instantânea detectável apenas por suas consequências é um modo de excluir o evento da modelagem. Isto se baseia em uma visão completamente irrealista do que efetivamente é um evento.

De acordo com o Teorema de Gödel (fato antecipado, informalmente, por B. Russell), as regras aplicáveis a uma estrutura não podem ser definidas, por *procedimentos gerais*, dentro da própria estrutura. A consequência prática deste princípio é que *colocar toda a modelagem em um único nível hierárquico lógico* (modelagem monoplane) constitui-se em um empreendimento fadado ao insucesso.

Os recursos tradicionais de modelagem de sistemas, tais como máquinas de estados e diagramas de seqüência, apenas contribuem para a análise dos sistemas, fornecendo elementos para serem acrescentados a um modelo da situação – que é, geralmente, tratado como se possuísse apenas um plano lógico (modelo monopiano). Isto demonstra a fraqueza teórica da modelagem de sistemas de informações como é comumente praticada hoje. A raiz do tratamento inadequado está em não se entenderem os fundamentos lógicos dos processos de representação, e à informalidade a que são relegados os conceitos de estados e eventos. Um modelo de sistema de informação precisa ser pensado como uma hierarquia de estruturas, cada nível estrutural codificando regras que se aplicam aos níveis inferiores; para desenvolver essa hierarquia de metaestruturas, é fundamental a definição explícita do que são estados e eventos. A dificuldade de se lidar com estados e eventos deriva da impossibilidade *intrínseca* de se poderem definir, dentro de uma estrutura, as leis de transformação da própria estrutura. Desenvolvemos, neste trabalho, uma formalização dos conceitos de estados e eventos de um modo orgânico à teoria da modelagem do conhecimento; este resultado contribui para explicitar as leis que regem os processos formais de representação. O texto investiga os fundamentos epistemológicos, ontológicos e fenomenológicos dos processos de representação, a partir de um ponto de vista não conteudista.

Abstract

Every information system is a representation of some sort of ideas; a computer system corresponds to representation that becomes operational by means of software. Conceptual modeling of information systems relies on the roots of the process: its aim is to formalize the concepts that people use for accomplishing their aims; in other words, ideally it is the **formalization** of the **knowledge** that people have about environments and subjects. Thus there comes the importance presented by the techniques of conceptual modeling, resulting in a fundamental question: to accomplish its potential of use, modeling techniques must be considered mainly as a problem of knowledge representation. This fact places questions commonly not regarded by the usual techniques of systems specification. Many troubles on the representation of knowledge about a subject result from a very deep problem; usually, the methodologies of modeling do not distinguish two fundamental matters: (i) the representation of policies and knowledge *about* the modeled situation belongs to (ii) a conceptual and logical level that is separated from the modeling level of the *very* situation – as a consequence, different logical levels are represented on the *same* model, bringing solutions of hard complexity and theoretically unattainable, as we hope to show in this work.

Not only in informatics, but in any semiotic system (including natural languages), understanding the representational level is built on the proper understanding of states and events. The behaviorist approach has marked the way that states are addressed by technical literature, based on the analysis of the *observed behavior* shown by objects that receives stimuli. To place the focus of analysis on the individual behavior of objects leads the states to be considered, too, from the viewpoint of isolated objects, but as a matter of fact the characterization of a given state is associated to the joint participation of several objects. The lack of formalization of the concept of state leads to the use of three kinds of models, leading to great complexities. If we adopt the traditional view of an (instantaneous) event, the event will be no the model but by “its manifestations”, and not by properly being modeled – and so it is excluded from modeling. Thus it is reached a completely unrealistic way of understanding what an event actually is.

Following the Gödel's Theorem (informally anticipated by B. Russell), the rules that are applicable to a structure cannot be defined, by *general procedures*, inside the same structure. The practical consequence of this principle is that *to place the entire modeling on the same model* (the monoplanar modeling) is enterprise faded to be unsuccessful. State machines and sequence diagrams are only proper only to analysis, adding elements to the situation model – the monoplanar model. This shows the theoretical weakness of the information modeling as it is put in practice nowadays. The roots of the unreliable way of doing rests on the bad understanding of the logical foundations of the representational processes., and to the lack of formality applied to states and events. An information system

model must be thought as a hierarchy of structures, each structural level being responsible for defining rules that applies to lower levels – to develop this hierarchy of meta-structures its is fundamental the explicit definition of states and events. This work develops a formalization of the concepts of states and events, from an organic standpoint on the theory of knowledge modeling; this result is a contribution to explain the laws that governs the formal process of representation. The work investigates the epistemological, ontological and phenomenological foundations of the process of representation, from a content-free point of view.

Tema

... pois ele é o Grande, e ultrapassa todas as suas obras.

Eclo 43, 28

embora em qualquer dada linguagem existam coisas que tal linguagem não pode expressar, é sempre possível construir uma linguagem de ordem superior em que tais coisas podem ser ditas. Haverá ainda, na nova linguagem, coisas que ela não pode dizer, mas que poderão ser ditas na linguagem seguinte e assim ad-infinitum.

B. Russell, Meu Desenvolvimento Filosófico

1 Introdução

É fora de questão que o desenvolvimento de qualquer software sempre deva tomar por base a representação conceitual dos assuntos que lhe dão motivo. A medida é ainda mais crítica quando se trata do desenvolvimento de sistemas corporativos, em que a construção de um modelo conceitual da organização é absolutamente essencial. Do ponto de vista econômico, é importante que uma tal representação da organização seja estável. Isto é, que o modelo não apenas possa ser compartilhado pelos sistemas já existentes, mas também que não necessite alterar-se para satisfazer as aplicações em desenvolvimento, exceto se a alteração for para incorporar acréscimos; além disso – ideal máximo – o modelo deve estar disponível para projetos que sequer tenham sido ainda cogitados. Cumprir semelhante propósito pode ser considerado utópico, mas o princípio da estabilidade mostra o tipo de problemas que habita o cotidiano de todo aquele cuja tarefa é lidar com modelos corporativos. A busca do modelo estável tem mobilizado mais de uma geração de modeladores e a impossibilidade de atingir mesmo metas mais modestas tem provocado profundas reflexões e desenvolvimento de muita tecnologia. Entretanto, um fato chama a atenção: até o momento – por mais que se apliquem metodologias, procedimentos de administração ou software de apoio – os modelos corporativos produzidos nunca são estáveis. Independente dos recursos

utilizados e da competência técnica dos modeladores, os modelos sempre são confrontados com situações imprevistas, exigindo alterações de tal ordem que sempre acabam por prejudicar as aplicações em funcionamento, ou, fato mais comum, levando à duplicação de sistemas. A instabilidade dos modelos corporativos mostra-se tão amplamente difundida que suas origens não podem ser atribuídas ao mau uso da tecnologia nem à imperícia dos modeladores.

Para fazer face ao problema, o presente trabalho realiza uma investigação sobre alguns princípios que têm permanecido inexplorados pela área de sistemas de informações, mas que devem, ou deveriam, regular qualquer processo de representação formal, aí incluída a especificação de software corporativo. Advogaremos a idéia de que a grande complexidade e rigidez dos modelos de sistemas (efetiva causa das dificuldades) resulta do modo inadequado como os conceitos de *estados* e *eventos* são compreendidos no campo da informática. Ademais, esperamos poder mostrar que a inadequação não pode ser resolvida por meio de eventuais reorganizações das atividades práticas de desenvolvimento. A complexidade dos atuais modelos é consequência de um enfoque teórico que necessita revisão a partir de seus fundamentos conceituais e que condiciona o próprio processo de desenvolvimento de aplicações.

Como principais resultados, o trabalho:

- propõe uma definição formal dos conceitos de estado e de evento, a partir de uma base filosófica firme,
- e implementa essas definições em linguagem de programação, para emprego no processo genérico de criação de sistemas de informação.

Uma consequência imediata da proposta é que estados e eventos tornam-se, de modo natural, diretamente representáveis nos modelos (e não representáveis “por suas manifestações”¹), e, assim, podem ser diretamente processados, fato que constitui

¹ É o caso, por exemplo, de se usar um atributo (ou uma função) denominado *tipo* para representar um dos estados *estoque-baixo*, *estoque-médio*, *estoque-normal* ou *estoque-alto* de um *produto*, solução em que o *estado* do *produto* se manifesta por meio de um *atributo* ou função (*tipo*), mas em

fator importante para a qualidade do software. Sob um prisma mais geral, uma concepção adequada sobre estados e eventos é fundamental para o entendimento de qualquer processo genérico de representação.

Nas análises que desenvolveremos, tomaremos por base a orientação a objetos, dado o largo uso dessa tecnologia pela engenharia de software. É preciso ressaltar, entretanto, que os temas investigados pertencem ao domínio dos métodos formais de representação, e, assim, ultrapassam o próprio âmbito da modelagem de sistemas de informações. Em outras palavras: as questões abordadas não se constituem em particularidades da orientação a objetos – como a análise que será feita de algumas técnicas (máquinas de estado) poderia fazer parecer –, os problemas considerados se manifestam em qualquer processo de representação. Deste modo, as propostas do trabalho aplicam-se, de modo geral, aos vários paradigmas de especificação de sistemas (baseados ou não em recursos computacionais), não apenas à orientação a objetos.

que o próprio estado não é representado. Sobre eventos detectáveis apenas por suas manifestações em um modelo, uma das principais fontes é [Cook94], um dos clássicos da orientação a objetos, pag. 79: *"We can know that an event has occurred only by detecting its effect on our model."*, *"Specifically, we are only interested in those events which cause our model to change its state."*

Parte I – O Problema

Os discípulos vieram e lhe perguntaram, por que lhes falais em parábolas? ... Porque a vós foi dado conhecer os mistérios do reino dos céus, a eles não.

Mt, 13: 10-11

É extremamente difícil falar do sentido e dizer a seu respeito algo que tenha sentido. Só através de um único meio poderíamos fazê-lo convenientemente: construir-se uma linguagem que nada signifique e estabelecer assim uma distância objetivante que permitisse emitir discursos desprovidos de sentido acerca de discursos com sentido.

A. J. Greimas, Du Sens

2 Dificuldades dos Formalismos de Representação

2.1 Inviabilidade da Modelagem Monoplana

De acordo com o Teorema de Gödel (fato antecipado, informalmente, por B. Russell), as regras aplicáveis a uma estrutura não podem ser definidas, por *procedimentos gerais*, dentro da própria estrutura. A consequência prática deste princípio é que *colocar toda a modelagem em um único nível lógico* (modelagem monoplana), como se ensina atualmente mesmo nos melhores manuais, constitui-se em um empreendimento fadado ao insucesso². Isto demonstra a fraqueza teórica da modelagem de sistemas de informações como é praticada hoje. A raiz do tratamento inadequado está em não se considerarem os fundamentos lógicos dos processos de representação, e à informalidade a que são relegados os conceitos de estados e eventos. Um modelo de sistema

² Os manuais de modelagem de sistemas de informação não distinguem níveis lógicos de modelagem (isto é, separação entre objetos e regras sobre objetos – regras que também podem ser objetos); a consequência é que os modelos produzidos são tratados como se possuissem apenas um nível lógico. Este processo monoplano, no caso geral, é inviável, pelo Teorema de Gödel.

de informação precisa ser pensado como uma hierarquia de estruturas, cada nível estrutural codificando regras que se aplicam aos níveis inferiores, e, para desenvolver essa hierarquia de metaestruturas, é fundamental a definição explícita do que são estados e eventos: “... *H. Kelsen caracterizou ... a estrutura do direito como uma pirâmide de normas, cimentada por uma relação geral de implicação entre normas ...: no seu ápice se encontraria a “norma fundamental”, que funda a legitimidade do todo e em particular da constituição ...*”, [Piag79].

2.2 A Pesquisa

Ao longo de nossa vida profissional, sempre intrigou-nos a difícil convivência que estados e eventos apresentam com os objetos básicos dos domínios conceituais. O problema entretanto, parecia ser consequência de razões muito profundas e intrínsecas, tal a constância do fenômeno (que, por sinal, aparece em qualquer técnica de modelagem, não apenas na orientação a objetos). De todo modo, tínhamos uma certeza: não poderia estar bem resolvido o modo como estados e eventos são compreendidos (e portanto representados) como objetos, embora não fizéssemos idéia de como resolver a questão de modo natural.

Esta intensa insatisfação levou-nos a desenvolver uma pesquisa para a qual colocamos o objetivo de não sair senão com uma de três respostas: (a) encontrar a solução em alguma fonte até então desconhecida; (b) demonstrar que o problema é insolúvel, aceitando conscientemente as soluções ordinárias; (c) elaborar uma nova solução.

Como primeira atividade, empreendemos uma análise do material relevante disponível até aquele momento. Não surpreendeu encontrar apenas confirmações dos

métodos comuns da literatura. O fato apenas reforçou a impressão de que alguma idéia fundamental estava escapando à teoria-padrão³.

Entretanto, no decorrer do processo de análise, ocorreu-nos que a dificuldade de se lidar com estados e eventos derivava da impossibilidade *intrínseca* de se poder definir, dentro de uma estrutura, as leis de transformação da própria estrutura. Podem-se definir as transformações, mas não as leis que regem os conjuntos de transformações. Caso a lei seja codificada dentro da estrutura, o será por expedientes *ad-hoc*, não por procedimento genérico. Esta idéia foi-nos sugerida por uma passagem de [Russ59]: “*Em minha introdução ao Tractatus, sugeri que, embora em qualquer dada linguagem existam coisas que tal linguagem não pode expressar, é sempre possível construir uma linguagem de ordem superior em que tais coisas podem ser ditas. Haverá ainda, na nova linguagem, coisas que ela não pode dizer, mas que poderão ser ditas na linguagem seguinte e assim ad-infinitum.*”. Idéias análogas já nos eram conhecidas da teoria dos conjuntos ($2^n > n$,⁴), da lógica (Teorema de Gödel, teoria dos tipos⁵), da lingüística estrutural (nível⁶). Ficamos chocados ao perceber que esta idéia, que, sob variadas formas, aparece em tantas áreas da ciência, era exatamente o conceito que permitiria o tratamento de estados e eventos, de um modo orgânico, na teoria da modelagem do conhecimento. Estranhamente, a idéia parece ignorada por toda

³ Denominaremos *teoria-padrão* à teoria da orientação a objetos como é apresentada em [UP99] e [Cook94], obras que exibem, seguramente, as mais elaboradas e refinadas propostas deste paradigma.

⁴ Dado um conjunto N , com n elementos, para qualquer número cardinal n , seu conjunto potência $P(N)$ – o conjunto de todos os subconjuntos de N – terá 2^n elementos; uma vez que $2^n > n$ (mostrado por Cantor), tem-se que, em qualquer correspondência um-a-um entre N e $P(N)$, haverá “sobra” de elementos em $P(N)$. Ou seja, “*Somos então confrontados com o fato que a condição de ‘todas’ as coisas de um certo tipo serem postas em uma coleção pode nem sempre ser preenchida; quando pensamos ter todas essas coisas (na coleção), não estarão todas na coleção apesar de tudo.*” [Hausd37], pag. 39. Para os fins que nos interessam, a consequência é que, no caso geral, o conjunto de *todas* as regras sobre um domínio (na realidade, o conjunto de todos os predicados n -ários) não pode ser definido, em sua totalidade, no mesmo nível de representação dos objetos básicos do domínio: sempre haverá uma “sobra” que “saltará” para um plano lógico superior. Esta é a idéia que Russell informalmente antecipou (citação anterior). Segundo Russell, “tal fato agora é amplamente conhecido, mas, na época em que foi explicitamente afirmado, constituiu-se em uma grande descoberta.” – citado de modo livre.

⁵ A mesma idéia da nota acima, porém formalizada por Russel.

⁶ “Na lingüística estrutural, ... o nível ... é constituído de unidades de mesmo grau, definidas pelas relações que mantêm entre si ... e com as unidades de nível superior.” [DicS79], pag. 305.

a teoria dos sistemas de informação, ao menos sob a forma como a praticam os especialistas em informática. Os teóricos informáticos, talvez por atuarem em uma ciência muito jovem e inexplorada, costumam ignorar as contribuições das ciências afins. Possivelmente por isso o princípio tenha se mantido inexplorado por tanto tempo.

Esta foi a idéia básica, que, nesta dissertação, levou à formalização dos conceitos de estado e evento, demonstrando que estados e eventos - e, após eles, toda uma hierarquia de metastruturas - são objetos como quaisquer outros, embora com uma estrutura peculiar. A fundamentação filosófica e técnica da idéia, suas implicações, a elaboração de detalhes, as comparações com procedimentos correntes, constam da pesquisa que resultou no presente trabalho. Cremos ter dado um tratamento bastante pormenorizado e orgânico a uma família de problemas que são sempre abordados por meio de recursos arbitrários e destituídos de um fio teórico condutor.

2.3 Delimitação do Tema

Neste trabalho, apresentamos uma formalização dos conceitos de estado e de evento, a partir de fundamentos filosóficos precisos.

Nosso objetivo é mostrar a viabilidade de um modelo teórico, explicitando suas bases conceituais, exibindo exemplos de aplicação prática, investigando suas implicações, e usando os resultados obtidos para uma crítica a práticas correntes de modelagem de sistemas de informações. Portanto, o enfoque não está dirigido para o desempenho operacional das aplicações, o que antes é um problema de engenharia e de aplicação da tecnologia atual. Nosso objetivo é mostrar a funcionalidade de um modelo de modelagem conceitual, válido como modo de representação, diretamente implementável, mas, no momento, não considerando como problemática a performance operacional dos resultados obtidos. Ainda que tenhamos idéias bem definidas a respeito do tipo de linguagem operacionalmente mais adequada em que a proposta deva ser implementada, as questões de implementação tecnológica não fazem parte deste trabalho, ficando para um desenvolvimento posterior. Mas faz parte deste traba-

lho mostrar a utilidade prática dos resultados como “linguagem e metodologia de representação”.

Nosso objetivo é dar o primeiro passo para formular uma linguagem genérica de formalização de conhecimento, de caráter não conteudista; ou seja, em nenhum momento se formulam hipóteses ou pressupõe-se algo sobre o mundo (natureza, finalidade ou origem dos objetos), ou sobre a natureza dos processos mentais ou de percepção; o desenvolvimento do tema é puramente *formal*, destituído de considerações a respeito da natureza de qualquer conteúdo. Deste modo, a linguagem tem por objetivo representar *conteúdos cognitivos*, sem nenhum compromisso com consistência lógica (aristotélica), correspondência com fatos, coerência entre teorias, critérios de existência etc. – todos indicadores de hipóteses sobre o mundo, o que ultrapassa de muito aquilo que pode fazer uma linguagem; qualquer afirmação sobre conteúdo, por mais genérica e filosófica que seja, faz parte de uma *crença*, que, mesmo bem fundamentada, pertence a alguma área do conhecimento, sendo alheia à linguagem, e alheia mesmo ao fenômeno cognitivo. Isto é o que diferencia uma linguagem semiótica, em sentido amplo – assunto deste trabalho – do caso mais restrito de uma linguagem lógica⁷. Assim, manter-nos-emos à margem de qualquer consideração a respeito da organização fisiológica do conhecimento e, também, alheios a qualquer definição de verdade ou de correspondência empírica; apenas nos ateremos às leis formais que regem os formalismos e os processos de representação; consideramos que tudo o mais é um problema de existência que extrapola o que pode fazer uma linguagem: uma

⁷ Em um sentido geral, como observa o Prof. Ricardo Gudwin, linguagens lógicas e semióticas são a “mesma coisa”, segundo uma abordagem rigorosamente peirceana. Nos termos deste trabalho, linguagem lógica é um caso particular de linguagem (artificial) semiótica, correspondendo àquele tipo de linguagem semiótica que contém o conceito (formal) de estado. Acreditamos não haver desacordo entre a conceituação proposta pelo Prof. Gudwin e a que estamos utilizando: (i) se aceitarmos que uma expressão lingüística expressa “estados e eventos do mundo”, ou que estados e eventos sejam considerados, eles próprios, “termos de linguagem”, teremos de aceitar que há identidade entre linguagens lógicas e semióticas; (ii) entretanto, estamos empregando “estado” – e, por consequência, “evento” – em sentido puramente formal, sem aventar qualquer hipótese a respeito da natureza dos termos de uma expressão lingüística artificial; deste modo, *linguagem lógica* – segundo nossa proposta – é uma linguagem artificial que formaliza o conceito de estado, sendo, assim, um caso particular de *linguagem semiótica*, a qual pode não formalizar o conceito de estado. Ou seja, uma linguagem lógica está necessariamente comprometida com valores-verdade formais (codificados na expressão lingüística), não necessariamente booleanos, ao passo que uma linguagem semiótica, no caso geral, poderá não o estar.

linguagem só pode afirmar relações, sem implicar existência de qualquer tipo⁸; qualquer forma de existência *resultante* de uma expressão lingüística é uma operação externa à linguagem, sendo um ato, voluntário ou não, de *crença* ou de *método*. Propomos uma linguagem adequada à representação de conteúdos cognitivos, não um método de análise de problemas filosóficos, lógicos ou empíricos. Wittgenstein disse no *Tractatus* [Wittg68] que “sobre o que não se pode falar, deve-se calar”; como fundo desta dissertação, gostaríamos de propor que, quanto à existência, *sobre o que se pode falar, não se pode mais do que falar*.

2.4 A Proposta

A dissertação é um primeiro passo na investigação de uma linguagem semiótica, apropriada a representar conhecimento em geral, linguagem que, embora sendo formal, não está comprometida com a investigação do verdadeiro e do falso – o que é próprio de uma linguagem lógica (de ciências empíricas) e não de um sistema semiótico geral.

Os principais objetivos do trabalho são:

1. abordar as técnicas de modelagem conceitual de sistemas de informação como meio de formalização de conhecimento;
2. formalizar os conceitos de *estado* e de *evento* de modo a poderem ser diretamente utilizados por linguagens de programação orientadas a objetos, para emprego no processo genérico de criação de software;

⁸ Segundo o ponto-de-vista deste trabalho, qualquer fenômeno lingüístico sempre ocorre no interior de uma estrutura cognitiva, o que faz com que qualquer existência externa seja necessariamente uma inferência extralingüística – isto é, não assegurada pela própria linguagem –, como o mostram as diferenças de interpretação e as alucinações. Mesmo construções internas à estrutura cognitiva, decorrentes de regras pertencentes à linguagem, apenas têm existência como relações entre elementos, sem que nada se possa afirmar a respeito de sua existência (realidade) material.

3. efetuar uma análise crítica dos conceitos assim formalizados de *estado* e *evento*, em comparação com sua representação por meio de *máquinas de estados*, a abordagem comumente citada na literatura técnica;
4. apresentar argumentos em favor da afirmação de que não é necessário mais do que o modelo “básico” da orientação a objetos (*classe, atributo, operação, classe-paramétrica, herança de propriedades e objeto*) para **representação e análise** de qualquer ambiente ou assunto, sendo qualquer outra técnica (*máquinas de estado, diagramas de sequência, diagramas de interação* e análogos) metodologicamente desnecessária, valendo apenas como procedimento heurístico, mas não fundamentalmente pertencente ao processo de modelagem;
5. justificar que as questões acima não são meramente teóricas, são também sumamente práticas: a formalização adequada permite que estados e eventos possam ser diretamente processáveis em um modelo, ao invés de o serem indiretamente, a partir de diagramas de estados e de interação; a representação direta de estados e eventos⁹ constitui-se em fator importante para a qualidade de software.

⁹ A representação adequada faz com que estados e eventos possuam existência como conceitos diretamente pertencentes a um domínio conceitual, ao invés de serem considerados como “metaconceitos” representados em formalismos alheios àquele utilizado para a representação dos objetos básicos do domínio conceitual.

Parte II – Epistemologia

... fosse o que fosse que o homem chamasse a toda criatura viva, esse seria o seu nome a partir daí.

Gn 2.19

... e o mundo dos significados não é outro senão o mundo da linguagem.

R. Barthes, Elementos de Semiologia

3 Significado

Teoria não conteudista do significado.

Para desenvolver uma abordagem dos processos de representação a partir dos fundamentos, nossa investigação toma como ponto de partida metodológico¹⁰ a teoria semiótica¹¹ de Charles Sanders Peirce¹² (1839-1914). O grande lógico e filósofo norte-americano – e, maior ainda, semioticista – dedicou sua vida a investigar o fenômeno da significação:

Nunca me foi dado estudar o que quer que fosse - Matemática, Moral, Metafísica, Gravitação, Termodinâmica, Ótica, Química, Anatomia Comparada, Astronomia, Psicologia, Fonética, Economia, História das Ciências, viste (sic), homens e mulheres, vinho, Metrologia - a não ser como estudo de semiótica.

¹⁰ Embora este não tenha sido o ponto de partida real da investigação, que só tomou forma após uma investida direta sobre os conceitos de estado e evento.

¹¹ *Semiótica*: ciência que estuda as condições da significação [Eco76]; não pode ser confundida com *semântica*, o estudo das significações [Crys85].

¹² Uma fonte de informações sobre Peirce, na *web* (rede mundial de computadores) encontra-se em *Arisbe: The Peirce Gateway* [Arisbe01].

É importante conhecer a teoria de Peirce para que se possa interpretar adequadamente os elementos envolvidos em um processo de representação, como *modelo*, *significado* e *coisa representada*. Existe uma tendência demasiado naturalista de considerar um modelo como representação do mundo – e a tendência estabeleceu-se de modo particularmente acentuado na área de informática, talvez por pressões de ordem prática, que afastam considerações de natureza mais conceitual. Tal ponto de vista está presente mesmo em fontes que enfatizam o rigor conceitual: “*Nosso objetivo expresso é usar os conceitos da tecnologia de objetos para descrever situações do mundo*”, [Cook94], pag. 29; “*sabemos que possuímos um conceito quando o aplicamos com sucesso às coisas que estão a nossa volta*”, [Odell95] pag. 13. Embora os autores mais críticos quanto aos fundamentos declarem que as representações têm em vista *objetos da consciência*, a atitude mais amplamente difundida é relacionar os objetos da consciência à *realidade observável*.

As relações entre *modelo* e *realidade* constituem-se em um dos mais interessantes pontos em que o discurso está quase que em completo descompasso com a prática. Se os autores dos manuais de modelagem seguissem suas próprias recomendações, dificilmente eles próprios conseguiriam desenvolver qualquer modelo: não teriam onde colocar os números, uma vez que estes não fazem parte da realidade observável; nem mesmo lhes seria possível representar idéias inovadoras, de coisas ainda não existentes fisicamente. De todo modo, a crença de que modelos são um espelho do mundo empírico constitui uma ideologia pragmática de CPDs¹³, cuja mais grave conseqüência é desviar a atenção do modelador, afastando-o da representação de conceitos abstratos: a lógica de processos do sistema, por exemplo, é tema tradicionalmente deixado de lado por várias metodologias de modelagem¹⁴ conceitual

¹³ CPD, abreviatura de *Centro de Processamento de Dados*.

¹⁴ A Engenharia de Informação, na versão original de K. Finkelstein, explicitamente recomenda que se modelem primeiramente os “dados”, somente depois, já conhecidos os modelos de dados, são modelados os processos. No outro extremo, a análise estruturada, segundo a versão inicialmente proposta por C. Gane, recomenda a modelagem de “processos”, sem propriamente pronunciar-se a respeito de “dados”, uma vez que não se pode considerar a definição de depósitos de dados como uma “modelagem”; ou seja, em sua forma original, *tecnicamente*, a análise estruturada nada fala a respeito da modelagem “de dados”, enfocando apenas processos.

(inclusive a orientação a objetos); também os estados são postos fora dos modelos, tratados como um “metaconceito”, por meio de máquinas de estados. Como consultor e como modelador prático, trabalhando diretamente com profissionais dos mais variados níveis e experiência, temos observado que o condicionamento da realidade observável retira do modelador a visão de elementos essenciais a uma representação apropriada. Ou seja, uma inadequada teoria do conhecimento produz conseqüências nefastas ao sucesso da modelagem conceitual – assim, a compreensão dos elementos envolvidos nos processos de representação reveste-se de interesse prático imediato.

Contudo, não se pode ser arbitrariamente crítico com relação às dificuldades provocadas pelas diferenças de entendimento a respeito de *modelo* (*signo*), *significado* (*conceito*) e *coisa* (*objeto*). Na realidade, esta é uma das mais árduas questões da filosofia, desde a antigüidade; apenas que, nos sistemas automatizados de informação, os problemas daí resultantes tornam-se ineludivelmente práticos, situando-se além (ou aquém) do terreno prosaico da discussão literária. Esclarecer os três conceitos constitui exatamente o problema mais fundamental da semiótica. Portanto, uma questão nada trivial, e que nos leva à teoria semiótica de Peirce.

4 Teoria do Signo de Peirce

4.1 Signo, Objeto e Interpretante

Em sua forma mais simples, *signo é uma coisa que está no lugar de outra coisa*, idéia que costuma ser expressa pela fórmula latina “*aliquid stat pro aliquo*”. Segundo este conceito, amplamente difundido desde a antiguidade clássica, a *fumaça* é um *signo* do *fogo* (“*onde há fumaça, há fogo*”); em muitos casos, como na prevenção de incêndios, a fumaça será considerada como sucedânea do fogo (*está no lugar* do fogo), para as ações a serem tomadas. No caso, diremos que fumaça é o *signo* do fogo, o qual, por sua vez, é o *objeto* do signo. Do mesmo modo, uma foto é o signo da pessoa fotografada, e a palavra “*mesa*”, no entendimento de alguns, será o signo cujo objeto é uma determinada mesa.

Para facilitar a expressão, denominaremos as questões tratadas neste capítulo de “teoria do signo”, embora esteja em jogo também o objeto, e, como veremos, mais outros conceitos.

Há diferentes concepções a respeito do tipo de ligação que o signo mantém com o objeto: segundo uma corrente, o signo é associado ao objeto por meio de uma convenção, segundo outra, o signo liga-se ao objeto por uma associação natural. Não cabe discutir a questão neste trabalho, mas é necessário ressaltar que a relação entre signo e objeto – expressa sob a fórmula básica *aliquid stat pro aliquo* – só se aplica a casos excepcionalmente simples, em que uma dada coisa – *signo* – corresponde exatamente a uma outra dada coisa – *objeto*. Uma tal situação nunca ocorre nas línguas naturais, em que ambigüidade é a regra e também pode não ser fácil dizer em que consiste a existência do objeto (o exemplo clássico é o unicórnio). Apenas em algumas linguagens artificiais, como a lógica e a matemática, verifica-se correspondência unívoca entre signo e objeto. Assim, é preciso aprofundar a teoria do signo, para que ela adquira utilidade prática em sistemas reais de representação.

Para adequar a noção de signo a situações naturais de uso, é necessário fazer intervir um terceiro elemento, que analisaremos com um exemplo. A associação entre a fumaça e o fogo só pode ser considerada “natural” em virtude de nossos hábitos. Para que fumaça seja efetivamente considerada como indicadora de fogo, é necessário que alguém (p. ex., um ser humano) ou algo (p. ex., um sensor) tenha a capacidade de, ao detectar a fumaça, associá-la ao fogo (deixaremos em aberto se a capacidade consiste em “evocar” a idéia do fogo, ou em “constatar” a existência de fogo). Este terceiro elemento, que participa, juntamente com o *signo* e seu *objeto*, do ato de significação, já era mencionado por Aristóteles (“estado da alma”), por Saussure¹⁵ (“conceito”¹⁶) e por Peirce (“interpretante” [Peir35]).

Assim, de acordo com uma respeitável tradição, todo ato de significação envolve três elementos. Temos observado, ao longo de exercícios pedagógicos sobre o fenômeno da semiose, em sala de aula, que grande quantidade de alunos, sem

¹⁵ Ferdinand de Saussure (1857 – 1913), grande lingüista suíço, natural de Genebra, considerado como criador da lingüística moderna e do estruturalismo.

¹⁶ Embora, como veremos a seguir, a idéia de *objeto* não encontre lugar no pensamento de Saussure. A necessidade de explicitar o caráter diádico (*significante/significado*) das idéias de Saussure, em

instrução prévia, acaba definindo a significação em termos dessa tríade, ainda que de modo naturalmente intuitivo e informal (fato que reforça nossa crença de que a representação dos processos cognitivos deva ser abordada a partir dessa base mínima, se o problema é manter proximidade entre representação e “processos naturais”).

A terminologia que estamos adotando, desde o início do capítulo, para designar esses elementos, é a de Peirce, a qual, agora com o terceiro elemento, aparece completa: *signo*, *interpretante* e *objeto*¹⁷. Outra grande vertente do pensamento semiótico, que tem sua origem em Saussure, denomina-os *significante*, *significado* e *referente*¹⁸. A terminologia de Saussure é predominante na semiótica e na lingüística de influência francesa. Existe vastíssima literatura a respeito do ato de significação – ou do fenômeno da semiose – cuja análise ultrapassa os objetivos deste trabalho¹⁹.

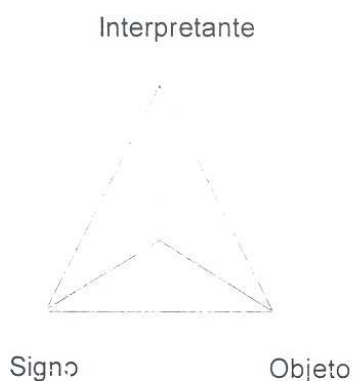


Figura 1: Tríade de Peirce

contraposição à concepção triádica de Peirce (*signo*, *interpretante*, *objeto*) foi-nos observada pelo Prof. Ricardo Gudwin, após a excelente revisão que fez deste trabalho.

¹⁷ Para maiores informações, consultar [LexSem].

¹⁸ O conceito de referente foi introduzido por R. Jakobson – [DicS79], 378 –, não fazendo parte explícita das concepções de Saussure. Efetivamente, Saussure atribuiu muito pouca importância ao *objeto* (*referente*), virtualmente excluindo-o de suas considerações. W. Nöth afirma que a idéia de objeto é alheia ao pensamento saussureano: “É irrelevante se Saussure ou outros teóricos do signo usam um termo tal como *objeto* ou *choso* em alguma passagem de sua discussão. A questão é se este termo tem um lugar sistemático na teoria semiótica. Para o sistema semiológico de Saussure, este certamente não é o caso.” [Nöth96], pag. 41.

¹⁹ Mais informações podem ser obtidas em [Eco84] [Eco75], [Todor77], [DicS79] e [Bart64].

Além da questão da terminologia, cada corrente teórica põe em relevo determinados aspectos do processo de significação. Saussure, por exemplo, como lingüista, praticamente exclui o referente de suas considerações, enfatizando o par significante/significado. Para o mestre de Genebra, o termo *signo* é empregado para designar a união entre o significante e o significado – o significante não existe sem o significado, e vice-versa – ambos ligados “*como as duas faces de uma folha de papel*” [Saus16]. Portanto, *signo*, para Saussure, tem significado diferente de *signo*, para Peirce. Nesta discussão, citamos apenas os dois grandes luminares do pensamento semiótico, e o fizemos tão somente para ilustrar como o estudo da significação está envolto em extraordinária variedade de conceitos e expressões, que mais ou menos se recobrem, e que exigiriam um verdadeiro tratado apenas para serem postos em correspondência.

4.2 Conceito de Semiose

No decorrer de muitos anos, Peirce escreveu e reescreveu inúmeras vezes sua visão sobre o fenômeno (para ele) natural da semiose, tendo-lhe dado uma formulação bastante feliz em:

Por semiose entendo uma ação, uma influência, que seja ou envolva uma cooperação de três termos, como por exemplo um signo, o seu objeto e o seu interpretante, tal influência tri-relativa não sendo jamais passível de resolução em uma ação entre duplas.

A conceituação mostra que, num ato de significação, os três elementos são inseparáveis, sem que um seja mais importante que o outro, ainda que, como esclarecem outros trechos de Peirce, o processo se inicie pela ação do signo. Um ponto importantíssimo é que nada foi dito a respeito da natureza de cada elemento; é fato estabelecido que a mesma *coisa* pode atuar ora como signo, ora como objeto, e, mesmo, como interpretante – ou seja, na semiose, a ação de cada termo, situada dentro de um esquema, é puramente posicional, nada tendo que ver com “essência” ou “natureza das coisas”. As relações entre os elementos – o próprio processo de

semiose – envolve tantas sutilezas, e contém tanta riqueza, que, nem de perto, haveria como abordar neste trabalho e sugerimos ao leitor a leitura das fontes já citadas.

Até o momento temos evitado dar uma “definição” dos elementos que, para Peirce, compõem o fenômeno da semiose, tendo desenvolvido o assunto basicamente por meio de exemplos. Entretanto, para poder justificar algumas afirmações que se seguirão, torna-se necessário caracterizar signo, objeto e interpretante, ainda que incorrendo na extrema dificuldade de oferecer o que quer que se pareça com definições (que, adiantamos, serão incompletas e demasiadamente esquemáticas).

Em primeiro lugar, deve-se ter em conta que os elementos são participantes de um ato de significação (o qual, por consequência, também poderá estender-se para um ato de comunicação). Neste contexto: (1) *signo* é o elemento que *é percebido* como uma expressão material, como palavras, desenhos ou gestos; (2) *objeto* é a *coisa evocada*, que pode não estar presente (Paris), pode ter existência puramente cultural (unicórnio), pode ser uma situação (eleição), etc.; ou seja, a *coisa evocada* não é necessariamente um objeto particular, sendo mais apropriadamente caracterizada como um “estado de coisas”; (3) interpretante é uma representação – “*uma outra representação do mesmo objeto*” [Eco76] – cuja motivação sempre remonta a algo como um conceito.

Das três noções, seguramente a mais problemática é a de *objeto*: o objeto é produzido no ato da significação (semiose), onde possui existência, não podendo ser confundido com um objeto “do mundo” – ele pode ser fruto de uma fantasia, de uma alucinação, ou pode até mesmo resultar de um ato perfeitamente lúcido de interação com “o mundo”, quando então eventualmente encontrará sua contrapartida “objetiva”, mas não há como assegurar nada a esse respeito. Portanto, o objeto da semiose não pode ser confundido com um objeto “do mundo” (não fosse assim, a história humana seria outra).

Resumindo: (1) *signo* é percebido; (2) *objeto* é evocado ou produzido no ato da semiose; (3) *interpretante* é uma representação – consideramos essa tríade caracterização razoavelmente segura para, sobre ela, tirar conclusões.

Toda a nossa abordagem da semiótica²⁰ tem em vista mostrar que seus conceitos não são abstrações destituídas de utilidade prática, mas que possuem o potencial para diretamente gerar aplicações nas atuais ciências computacionais ou cognitivas, desde que dotadas dos instrumentos adequados.

4.3 Aplicação do Conceito de Semiose

Embora a ação de cada elemento da semiose independa da substância que os componha, neste ponto da análise precisamos correr um risco e afirmar que, de algum modo²¹: (1) *signo* “tem a ver” com “recepção” de sinais, (2) *interpretante* “tem a ver” com “significado”, (3) *objeto* “tem a ver” com “produção” de sinais²², e (4) os três elementos, em conjunto, “têm a ver” com configuração de ambiente (ambiente externo), e formam, eles próprios, um ambiente (ambiente interno).

Mas, interno e externo a quê?

²⁰ Segundo as linhas da semiótica triádica de Peirce e da semiótica diádica de Hjelmslev (sob o nome de Glossemática), ambas de conteúdo verdadeiramente inspirador, se a questão é a modelagem de *estruturas cognitivas* – estruturas que: (i) possuem objetivos a cumprir; (ii) armazenam e processam conhecimento; (iii) conhecem seu ambiente; (iv) com o qual se comunicam por meio de mensagens. (Mensagem: “... sequência de sinais, organizada segundo as regras de um código: assim ela pressupõe operações de codificação e de decodificação” [DicS79], pag. 273.)

²¹ Nas análises que se seguem, é preciso ter em conta que a semiose é considerada como um fenômeno que ocorre internamente a uma estrutura cognitiva. Deste modo, não está em jogo, diretamente, o ambiente externo à estrutura, a qual se comunica com o ambiente por meio de mensagens (cujo suporte imediato são sinais – veja nota anterior).

²² Do ponto de vista da estrutura cognitiva que é sede do ato semiótico, o *objeto semiótico* é um “resultado” que, de algum modo, corresponde a um possível objeto externo (ou seja, o *objeto semiótico* corresponde a um ato de configuração de ambiente por parte da estrutura cognitiva), ao qual a estrutura eventualmente se remete por meio de sinais/mensagens. Este “(objeto)resultado” provém da ação conjunta do signo e do interpretante, sendo, porém, deles indissociado. Tal modo de interpretar o *objeto* viabiliza a representação e a implementação computacional de estruturas cognitivas. Acreditamos, em linhas gerais, que esta interpretação não está em desacordo com o conceito abstrato de objeto peirceano, uma vez que aquilo a que estamos denominando *objeto semiótico* (aqui considerado como interno à estrutura cognitiva) não pode confundir-se com um objeto “do mundo”. Toda a questão necessita de muito mais análise, sobretudo a respeito das relações entre *objeto*, em sentido peirceano, a aquilo a que se pode chamar *objeto do mundo*, em sentido físico.

A resposta vem de outro elemento, que Peirce parece que excluiu do âmago do fenômeno da semiose: o *intérprete*. Peirce costumava exprimir sua visão do mundo em termos de tríades – deste modo, sem dúvida, um quarto elemento não haveria de figurar no seio de sua concepção mais fundamental. Entretanto, para que o fenômeno da semiose possa ocorrer, é necessário ter em cena o *intérprete*²³. O *intérprete* é o lugar (formal) onde está situado o interpretante; ora, o interpretante é também entendido como uma *disposição* provocada pelo signo; portanto, o interpretante pode ser entendido como um “estado de alma” (confirmando Aristóteles) do intérprete.

Como sabemos, por Peirce, que o fenômeno semiótico é uma ação única, indecomponível, de três termos, segue-se que o lugar *formal* desta ação é exatamente o intérprete – uma vez que não há como separar os três termos que dela participam, e um desses termos reside no intérprete. Ou seja: *o intérprete é a condição lógica de existência do ato de significação*. (E assim, efetivamente, ele não pode ser acrescentado, como quarto elemento, à tríade peirceana, o que mostra a pureza conceitual do mestre²⁴.)

Como trabalho exploratório, a partir das considerações precedentes, formalizamos e implementamos uma arquitetura de agente cognitivo²⁵, em que a estrutura cognitiva é justamente um intérprete peirceano. Assim, em termos atuais, o intérprete peirceano mostra uma de suas faces: um agente cognitivo – um agente cognitivo abstrato, conceitual.

Mas a implementação demonstra um fato: as idéias semióticas de Peirce, como as de vários outros pensadores, podem tornar-se diretamente operacionais,

²³ O Prof. Gudwin discorda veementemente desta afirmação; segundo ele: “Para que possamos compreender o que é o fenômeno de semiose, pode ser conveniente introduzir o conceito de intérprete, da mesma maneira que, em lógica, é conveniente utilizar $A \Rightarrow B$, ao invés de $\sim A \vee B$ ”. Ou seja, o intérprete é um conceito facilitador!

²⁴ *Mestre*, com “semimaiúscula”, uma vez que aceitamos a ponderação do Dr. Marcos Shmeil de que *Mestre*, com maiúscula, é um significante que possui outro significado!

²⁵ Agente dotado de uma *estrutura cognitiva*, como em [ArtInt95].

produzindo resultados práticos imediatos. A partir das ciências hoje denominadas Inteligência Artificial, Engenharia do Conhecimento, Semiótica Computacional²⁶, etc., toda a área da representação e do processamento do conhecimento, aí incluído o que se conhece por “desenvolvimento de sistemas de informação”, poderá beneficiar-se de recursos que têm permanecido inexplorados e que, seguramente, farão parte do futuro das ciências do conhecimento.

4.4 Implicações da Teoria do Signo

Com sua teoria do signo, Peirce isolou o significado como uma questão em si, dissociada do problema da verdade, da correspondência com os fatos, ou da concordância com doutrinas. Ao separar significado de verdade e de realidade, Peirce lançou as bases da semiótica moderna, possibilitando uma teoria não conteudista dos modelos, com profundas implicações nos dois pólos – a filosofia e a tecnologia.

Uma representação formal, no sentido de Peirce, modela uma “crença” – independente de sua verdade ou falsidade formal ou empírica. É interessante observar que, deste modo, a teoria do significado de Peirce encontra-se (fecha um círculo) com as conclusões do arquiobjetivo Popper, quando este afirma que “*mesmo que uma teoria seja verdadeira, jamais o saberemos*” [Popper63] (citado de modo livre). Assim, Peirce eliminou o fosso existente entre “representação formal” e “linguagem natural”, abrindo caminho para a formulação de uma linguagem genérica de representação formal – linguagem semiótica – destinada a representar conteúdos cognitivos, possibilitando que uma teoria lógica da linguagem possa ser construída de modo independente de qualquer pressuposto conteudista implícito, formal ou empírico.

²⁶ Para informações sobre Semiótica Computacional (abordagem que combina semiótica, métodos computacionais e inteligência artificial), uma fonte de referências na web é *Computational Semiotics* [CompSem01].

Deve-se, ademais, reconhecer um fato: afirmar o não conteudismo de um modelo não implica afirmar que “todo” modelo é uma “criação livre do espírito”, mas implica reconhecer que qualquer atitude, com relação ao conteúdo de um modelo, é uma operação “externa” a ele – é uma operação semiótica.

Parte III – Ontologia

*O espírito do homem é uma lâmpada do Senhor,
sonda o mais íntimo do ser.*

Pr 20, 27

As coisas são o que são, serem-no é a sua irrefragável força, e a nós cabe-nos tentar compreendê-las, ajeitá-las, se possível, à oportunidade e ao interesse da ocasião, mas respeitando-as sempre, evitando sobretudo cair na tentação do avestruz, o que, na circunstância, seria fingir que as coisas, afinal, são outra coisa.

José Saramago, entrevista à revista Status,
1986

5 A Estrutura

Para encontrar a chave para os interrelacionamentos, temos de ir ainda mais fundo do que a observação direta. Precisamos não apenas ver objetos e eventos isolados mas descobrir como eles estão conectados.

Morris Swadesh, The Origin and
Diversification of Language

O conceito de estrutura, historicamente, remonta a Ferdinand de Saussure, em seu *Curso de Lingüística Geral*²⁷ [Saus16], obra em que o Mestre de Genebra conceitua a língua como um conjunto de relações abstratas, independente do modo como as palavras se manifestam. Segundo essa teoria, o material fônico é apenas um dos modos de uma língua realizar-se, não sendo, de modo algum, essencial para (a descrição de) o funcionamento da língua. Esta visão abstrata de língua, que moldou a noção de estru-

²⁷ Obra póstuma, na realidade escrita por dois alunos de Saussure, a partir de notas de aula. É interessante notar que a palavra “estrutura” não consta do *Curs*, em que aparece “sistema”, com o significado que atualmente se dá a estrutura.

tura, constitui um dos dois pilares da semiótica moderna²⁸ (o outro é a teoria do significado de Peirce, tema de capítulo anterior).

Todo o trabalho de criação de um sistema computacional, em qualquer de suas áreas de aplicação, essencialmente, constitui-se na definição formal de estruturas, daí a enorme importância que o assunto assume para nosso tema. Muito das teorias e práticas de uso corrente na informática seriam muito melhor formuladas, compreendidas e aplicadas se seus proponentes e usuários, consciente e deliberadamente, partissem da noção de estrutura. A esse respeito, pretendemos mostrar exemplos ao longo do trabalho.

Em um pequenino clássico, *O Estruturalismo* [Piag79], Piaget abordou a noção de estrutura com grande clareza e precisão. Em dois trechos, Piaget formula o que seguramente mais se aproxima de uma “definição” de estrutura – [Piag79]²⁹, pag. 8:

Em uma primeira aproximação, uma estrutura é um sistema de transformações que comporta leis enquanto sistema (por oposição às propriedades dos elementos) e que se conserva ou se enriquece pelo próprio jogo de suas transformações, sem que estas conduzam para fora de suas fronteiras ou façam apelo a elementos exteriores. Em resumo, uma estrutura compreende os caracteres de totalidade, de transformação e de auto-regulação.

Em outra passagem, Piaget diz [Piag79], pag. 9:

... convém comentar um pouco a definição proposta e esclarecer porque uma noção aparentemente tão abstrata como um sistema de transformações, fechado sobre si mesmo, pode fazer nascer em todos os domínios tão grandes esperanças.

O aspecto mais fundamental da noção de estrutura é o *de sistema de transformações fechado sobre si mesmo*: as transformações a nada recorrem que esteja fora da estrutura e seus resultados preservam a estrutura; pela lei da *totalidade*, as associ-

²⁸ Sobretudo após o trabalho de Hjelmslev, em *Prolegômenos a uma Teoria da Linguagem*.

²⁹ *O Estruturalismo* é leitura praticamente obrigatória a quem desejar aprofundar-se na noção de estrutura; o conhecimento de seu conteúdo é benéfico a todo aquele que, de algum modo, lida com produção de conhecimento.

ações entre elementos dependem do conjunto como um todo, e o *conjunto possui propriedades distintas das propriedades de elementos isolados*; com a *auto-regulação*, as estruturas são responsáveis por regulamentar suas próprias modificações, codificando as leis que regem suas transformações estruturais, e que são responsáveis pela *conservação da estrutura* e “*um certo fechamento*”.

O conceito de Piaget destaca o aspecto *operacional* de uma estrutura, como sistema de transformações. Existe, entretanto um outro aspecto de igual importância, e que se situa na raiz do pensamento saussureano: uma estrutura é também um *sistema de relações*³⁰. Greimas e Courtés [DicS79], pag. 161, retomando uma formulação de Hjelmslev, põem as *relações* em relevo com a seguinte formulação:

... consideramos uma estrutura como uma entidade autônoma de relações internas, constituídas em hierarquias.

Todorov e Ducrot esclarecem de que modo os elementos da estrutura participam das relações – [DiEL72], pag. 27:

E é isso que os saussurianos entendem quando falam de sistema ou estrutura (da língua): os elementos (lingüísticos) não têm nenhuma realidade independentemente de sua relação com o todo.

Nesta citação, os parênteses foram acrescentados por nós, para realçar a amplitude da afirmação, que possui validade para estruturas em geral, não apenas para estruturas lingüísticas.

³⁰ De um ponto de vista muito essencialista, é possível que as noções de *transformação* e de *relação* possam unificar-se sob a idéia de *função*, quando, então, o conceito de estrutura é passível de ser formulado sob um princípio único. Entretanto, o grau de abstração de tal abordagem, ainda que de grande utilidade para os sistemas computacionais, ultrapassa os objetivos do trabalho. Fomos demovidos de prosseguir nessa vertente ao constatar que mesmo filósofos metafísicos adotam a visão (que, em certo momento, consideramos como “relativamente pouco fundamentada”) do ser enquanto “*princípio de existência*” (associações) e “*centro de atuação*” (transformações) [Joliv72]. O conceito de *função* está na base da noção algébrica de *categoria*, constructo que introduz extraordinárias possibilidades à formalização de estruturas, embora de modo extremamente abstrato. Para que esta idéia seja viável, é necessário resolver alguns problemas técnicos, sobretudo ligados à maior generalidade da noção matemática de relação, em confronto com a noção de função.

Em seu conjunto, os dois últimos textos contêm elementos fundamentais, merecedores de atenção especial: (1) uma *estrutura*, ela própria, é um “objeto”³¹ – *uma entidade autônoma*; (2) uma estrutura caracteriza-se por *relações*, não pela matéria de que são feitas as coisas; (3) evidentemente, estruturas possuem *elementos*, mas a realidade (existência) deles na estrutura só se justifica pelas relações; e (4) as *relações* só adquirem sentido na *totalidade* da estrutura, e não como ligações isoladas entre elementos.

Certamente, os dois aspectos de uma estrutura (o operacional – *sistema de transformações* – e o relacional – *sistema de relações*) não terão escapado ao olho treinado do especialista em metodologias de informática, acostumado a lidar com modelos “estáticos” e “dinâmicos”: percebe-se que uma estrutura conjuga as duas faces, não separando o “estático” do “dinâmico”. Este fato mostrar-se-á fundamental a algumas de nossas conclusões, sobretudo na recusa em aceitar as miríades de modelos comumente empregadas para representar os vários aspectos de domínios conceituais.

Como foi formulado por Piaget, o conceito de estrutura seguramente é suficiente para fornecer a base conceitual a ser empregada para fundamentar a orientação a objetos, do modo como a teoria costuma ser apresentada hoje. Por exemplo, em seu grande clássico [Meyer97], *Object-Oriented Software Construction, 2nd Edition*, Bertrand Meyer proclama sua teoria dos contratos, que rege a interação entre objetos, como a técnica mais fundamental da orientação a objetos. Os constructos básicos dos contratos – *invariante*, *pré-condição* e *pós-condição* – em termos estruturais, podem ser interpretados como: *invariante*, função que codifica leis de *totalidade*; *pré* e *pós-condições*, funções que codificam leis de *auto-regulação* aplicadas a operações de *transformação estrutural*. Deste modo, ficam satisfeitas todas as condições que especificam uma estrutura, segundo a formulação de Piaget. Ou seja, os contratos de Meyer destinam-se a assegurar um “*sistema de transformações fechado sobre si mesmo*”, dotados das características de “*totalidade*” e “*auto-regulação*”; e sua teoria é sufici-

³¹ Em um sentido do termo “objeto” que será tema de análise no próximo tópico.

ente para descrever qualquer estrutura, sem que nada mais seja necessário acrescentar aos três constructos propostos. E, assim, Bertrand Meyer tem razão em sua postulação.

Muito dificilmente pode agregar-se algo ao conceito de estrutura formulado por Piaget e por Hjelmslev – exceto por um detalhe: exatamente a porta que o próprio Piaget deixou semiaberta – “*um certo fechamento*” –, porta que se abre para a escadaria que Russell tão bem enxergou³² – “*hierarquias*” – escadaria que, talvez, o próprio Hjelmslev, embora a reconhecendo, não avistasse aonde leva, nem de onde vem.

Temos agora o elemento que completa o conceito de estrutura: a noção de hierarquia. Mas, como deve ser entendida a noção de hierarquia?

Sabemos, pelo Teorema de Gödel, que, no caso geral, uma estrutura não pode codificar, ela própria, a totalidade de suas regras de transformação. Por este motivo, o fato já conhecido por Bertrand Russell, qualquer estrutura, necessariamente, terá de recorrer a uma outra estrutura, que lhe é superior, para codificar algumas de suas leis reguladoras – e assim *ad-infinitum*³³. Isto é: nenhuma estrutura é absolutamente fechada.

Assim, por necessidade lógica, toda estrutura está associada a uma hierarquia de estruturas definidora de regras de transformação.

O reconhecimento deste fato parece ter escapado a Greimas e a Courtés em seus comentários – [DicS79], pag. 162 – à noção de estrutura de Hjelmslev, uma vez que aqueles dois autores justificam a hierarquia pelo fato de uma estrutura ser “*decomponível em partes que, estando sempre relacionadas entre si, mantêm relações com o todo que constituem*”. Ora, uma tal “explicação” é verdadeira para qualquer estrutura, independente de haver ou não hierarquias.

³² Veja a seção Tema da Pesquisa.

³³ Esse processo sem fim só se interrompe como resultado de medidas de ordem pragmática, não por razões lógicas.

Um fato importante, e que precisa ser ressaltado, é que, na noção de estrutura, não está em jogo qualquer fenômeno de significação, não cabendo indagar se uma estrutura codifica um *conhecimento* ou se expressa algo *efetivamente existente* na natureza.

A noção de estrutura, segundo a formulação de Piaget e de Hjelmslev, e a hierarquia estrutural³⁴, de acordo com a interpretação de Russell, são resultados importantes, que reaparecerão posteriormente neste trabalho.

5.1 Conceito de Objeto

As análises precedentes mostram que um elemento figura em uma estrutura apenas *como* participante de associações e *como* operador de transformações³⁵; para nenhum desses modos – *participante* e *operador* – é importante o material de que o elemento é feito. Isto é: um elemento faz parte de uma estrutura em decorrência dos *papéis* que nela exerce, e não por suas características materiais. (É neste sentido que uma estrutura não é uma substância, mas um modo de organização.). Ora, um elemento de uma estrutura nada mais é do que um *objeto*.

Assim, tomando por base o conceito de estrutura, podemos conceituar objeto sem recorrer a qualquer noção de substância, ou aventar qualquer hipótese a respeito do que um objeto “realmente” é: *objeto é um feixe de associações e um centro de funcionalidade*.

³⁴ A hierarquia estrutural, a rigor, não faz parte da noção de estrutura (como faz parecer a formulação de Hjelmslev), embora a complete.

³⁵ Se conceituarmos *agente* como “algo que percebe e atua”, [ArtInt95], pag. 7, pode-se dizer *agente de transformações*; caso se exija de um agente algum tipo de comportamento racional (capacidade de realizar inferências corretas), então devemos permanecer com a expressão mais conservadora *operador de transformações*.

Em outros termos: o *feixe de associações* constitui aquilo a que se denomina *atributos* do objeto; o *centro de funcionalidade* compõe o conjunto de suas *operações*.

O conceito estruturalista de objeto possui profundas implicações teóricas e práticas, algumas das quais passaremos a abordar.

Não cabe a uma ciência das representações tomar partido a respeito da substância de um objeto, nem mesmo dizer se um objeto tem existência no mundo real ou se é uma construção mental, muito menos lhe toca questionar os motivos da existência de um objeto; este é assunto das ciências empíricas, do senso comum, da filosofia, da magia ou da religião — posições reveladoras de hipóteses conteudistas. Uma ciência das representações deve ter por objetivo, tão somente, explicitar os mecanismos de representação dos objetos, o que quer que eles sejam — e um tal programa não é pouca coisa.

A noção de atributo merece uma atenção especial: *um atributo é uma associação* de um objeto a outro objeto. Esta formulação, derivada da noção de estrutura, coincide exatamente com o conceito de atributo de Wittgenstein, como relatado no *Tractatus* [Wittg68]. Wittgenstein foi, talvez, o primeiro ser humano a deixar explícito que um atributo é uma associação³⁶ entre objetos, deste modo abrindo caminho para o conceito de sistema semiótico independente de pressupostos conteudistas (isto é, pressupostos relacionados à natureza dos objetos).

É interessante notar que a idéia de objeto como *feixe de relações*, de feição nitidamente estruturalista, é antecipada por Alfred North Whitehead³⁷ em seu esforço para responder à pergunta que tão insistentemente o perseguia — *como compreende-*

³⁶ O radicalismo deste conceito, que se afasta da tradição aristotélica tão longamente cultivada, perturbou o próprio Wittgenstein, e colocou em xeque várias de suas convicções.

³⁷ Whitehead é coautor, juntamente com B. Russell, dos *Principia Mathematica*, obra em que ambos defendem a tese de que a matemática pode reduzir-se à lógica, e que é o mais influente tratado de lógica escrito após Aristóteles. Nos *Principia*, Whitehead e Russell desenvolvem uma teoria lógica das relações.

mos a natureza? – e que ele procurou responder como físico e como lógico. Em suas perquirições [Whith68], Whitehead consistentemente aplica a idéia de objeto como feixe de relações, e de natureza como sistema de relações, realizando, a partir daí, profunda investigação sobre o conhecimento humano, onde ele buscou explicações para os fundamentos da física. Com suas respostas, Whitehead esteve no limiar de uma teoria estruturalista do conhecimento, à qual ele, de certo, teria chegado, fosse mais formalista de conceitos e menos voltado para conteúdos, como costumam ser os físicos e os lógicos³⁸.

A noção de atributo como associação constitui uma inovação radical no campo da filosofia. Segundo o conceito estruturalista, que é também a intuição ontológica de Wittgenstein, o objeto é o constructo mais básico; “materialmente” não há uma coisa mais elementar do que objeto, o qual só se explica pela associação com outros objetos³⁹, e pelas transformações por que é responsável.

Portanto – e esta é a conclusão fundamental –, um objeto não pode ser reduzido a elementos mais básicos, como qualidades, características ou propriedades *intrínsecas*. Ora, deste modo, desaparece o conceito aristotélico de atributo, em que atributo é entendido como uma *outra* coisa que “faz parte” ou “pertence” ao objeto, mas que *não é* um objeto⁴⁰.

Assim, o conceito de objeto adequado à representação de conhecimento necessariamente aparta-se de uma longa, arraigada e cultivada tradição, fundada, em

³⁸ “Nosso tema é a natureza”, pag. 168, [Whith68]. “Eu mantenho a tese mais humilde de que a natureza é um sistema. ... A mesma doutrina está essencialmente entretecida com toda a especulação física moderna. Lá por 1847 Faraday observava ...”. - pag. 163. Sabemos que o conceito de *sistema* foi atribuído a Saussure. Porém o *Cours* foi publicado em 1916, *O conceito da natureza* foi publicado em 1919, baseado em idéias desenvolvidas muito antes, quando a obra de Saussure não era conhecida, nem mesmo entre os linguistas; ademais Whitehead diz que a idéia de sistema não é nova “nem na filosofia nem na ciência”, e cita como exemplo Faraday em 1847!

³⁹ Ou seja, qualquer estrutura encontra-se envolvida em uma cadeia de associações potencialmente sem fim; a cadeia de associações só pode ser interrompida por medidas de ordem pragmática.

⁴⁰ Mesmo que “realmente” um atributo fosse diferente de um objeto, não caberia a uma representação decidi-lo, devendo dar igual tratamento tanto a atributos quanto a objetos, sem envolver-se em partidarismos “conteudistas” – uma linguagem precisa estar disponível para todos, independente de suas crenças ou decisões metodológicas.

termos teóricos, por Aristóteles, e que tem sido considerada como natural e autoevidente por período mais do que milenar.

E, mais importante ainda, o entendimento⁴¹ de que os objetos “possuem” uma *outra* coisa (que não são objetos), que são seus atributos, é plenamente reforçado pelo senso comum – e esta é a razão de a proposta aristotélica ser considerada autoevidente e inquestionável durante tanto tempo.

A noção aristotélica de atributo possui profundas conseqüências práticas, permeando, de modo acrítico, o pensamento que está na base invisível de grande número de propostas para representação de conhecimento. É o que se pode chamar de “naturalismo” em modelagem, que joga, para fora dos modelos, as abstrações que não se mostrem imediatamente como “coisas”. Assim, o empirismo ingênuo condena processos, estados e eventos a terem existência apenas como metaconceitos, apresentados por metalinguagens⁴², e não diretamente representados nos modelos em que residem os “verdadeiros” objetos – dificultando desnecessariamente todo o processo de representação.

Esperamos ter mostrado que a discussão fundamental a respeito dos conceitos de estrutura e objeto reveste-se de interesse eminentemente pragmático, de modo algum reduzindo-se à vã elocubração filosófica; seus resultados alcançam cada analista de aplicações, quem, ao sentar-se em uma cadeira para modelar, persegue um sistema que “tem que funcionar na prática”. É claro que um técnico, na tarefa de modelar aplicações, não pode revirar os princípios das técnicas de representação – do mesmo modo que um engenheiro não recorre aos axiomas da aritmética para fazer cálculos: para isso existem as metodologias.

⁴¹ Segundo sugestão do Prof. Gudwin: “..., **infelizmente**, o entendimento de que os objetos ...”!

⁴² Como as máquinas de estados.

5.2.1 Portanto, o constructo que deu nome à própria *orientação a objetos* – o *objeto*⁴³ – está assentado sobre bases conceituais de extrema solidez e possui conteúdo sumamente filosófico.

Os objetos que figuram nos sistemas de informação são os mesmos objetos com que já se deparava Aristóteles, e que atormentaram Wittgenstein.

5.2 Representação de Objeto

Para representar objetos, utilizaremos o formalismo textual da linguagem Eiffel [Meyer97] e o formalismo diagramático da linguagem UML [UML97]. Do ponto de vista da engenharia de software, a linguagem Eiffel⁴⁴ é a mais completa proposta de uma linguagem orientada a objetos, não apenas para programação, mas também para especificação conceitual de sistemas. Para representação em formato de diagramas, a linguagem gráfica UML (uma bem sucedida iniciativa de criação de um padrão gráfico⁴⁵ para a orientação a objetos) constitui opção bastante adequada. Em ambas as modalidades de representação, deve-se ter em conta aquilo que um objeto efetivamente é: *feixe de associações e centro de funcionalidade*.

⁴³ Deve-se notar que esse conceito de objeto é genérico, não estando restrito à orientação a objetos. Apenas estamos sublinhando a fundamentação filosófica da orientação a objetos, como base formal para a proposição de uma linguagem genérica de representação de conhecimento.

⁴⁴ Entretanto, para simulação de estruturas cognitivas, em que é necessária a redefinição de classes em tempo de execução, possivelmente a ferramenta mais adequada seja a linguagem Python, tão fervorosamente defendida pelo Dr. Edson Scalabrin, da PUCPR.

⁴⁵ Embora os proponentes da UML – Unified Modeling Language passem ao largo dos aspectos conceituais de embasamento da tecnologia que propõem, reproduzindo, em sua linguagem, a Torre de Babel dos formalismos da orientação a objetos. Mesmo que a UML tenha sido formalizada por meio de uma metalinguagem (SCL – Semantic Constraint Language), essa formalização não ultrapassa o âmbito da expressão lingüística, não podendo ser considerada uma fundamentação em termos de idéias fundamentais: a UML incorpora uma grande quantidade de soluções “bem sucedidas”, mas sem empreender uma análise de fundamentos – ou seja, aborda aquilo que está “da tecnologia para a frente”, e não “da tecnologia para trás”.

5.2.1.2 Operações

No exemplo, as operações (transformações estruturais) que podem ser realizadas por um *cidadão* estão a cargo das operações `casar_com` e `set_conjuge`. Deve-se observar que a operação de criação de uma instância de *cidadão*, `make`, recebe um tratamento especial, pois pôr um objeto em existência não se pode considerar como algo pertinente ao objeto, embora pertença à classe.

⁴⁸ A classe `NOME` exibe um dos recursos mais poderosos da orientação a objetos, ao encapsular todo o tratamento aplicado a nomes, o que vai muito além de definir que `nome` é uma sequência de caracteres (`string`).

5.2.1.3 Classe Exemplo - CIDADAO

```
class CIDADAO
inherit
    PESSOA_FISICA
creation make
feature - associações
    nome: NOME
    conjugue: CIDADAO-- esposo ou esposa
    endereco: STRING
    emprego: PESSOA_JURIDICA
feature -- funcionalidade
    solteiro: BOOLEAN is
        do
        end -- solteiro
    casar_com(benzinho: CIDADAO) is
        do
        end -- casar_com
    pode_casar_com(outro: CIDADAO): BOOLEAN is
        do
        end -- pode_casar_com
    set_conjuge(outro: CIDADAO) is
        do
        end -- set_conjuge
invariant
    matrimonio_consistente: solteiro or conjuge.conjuge =
Current
end -- CIDADAO
```

5.2.1.4 Observações sobre Classe

Que tipo de coisa uma classe define? Em primeiro lugar, como já se sabe, define estruturas: associações, operações e regras. Em segundo lugar, define apenas estruturas; isto é, uma classe *não define* significado intuitivo ou “mental” de um

objeto, nem mesmo possui qualquer relação com definição de dicionário para objeto – uma classe não define nem significados nem essências. Uma prova de que classe não define (diretamente) significados é que, se seus termos forem consistentemente substituídos por outros (p. ex., mediante uma operação “*change all*”), a especificação estrutural ficará preservada (alunos o sabem, ao trocar nomes de variáveis em programas), embora o significado percebido possa alterar-se.

Em termos semióticos, uma classe é um *signo*, no sentido de Peirce – qualquer associação de classe a significado, ou de classe a objeto, é uma operação realizada por um observador (intérprete). Este tema será desenvolvido no próximo capítulo, ao abordarmos o conceito de modelo.

5.2.2 Representação Gráfica

A representação gráfica, em linguagem UML, exibe a hierarquia conceitual a que pertence a classe CIDADAO, em que figuram as classe PESSOA_FISICA e PESSOA_JURIDICA, citadas na especificação textual, e a classe PESSOA, acrescentada para tornar o exemplo mais ilustrativo (Figura 2).

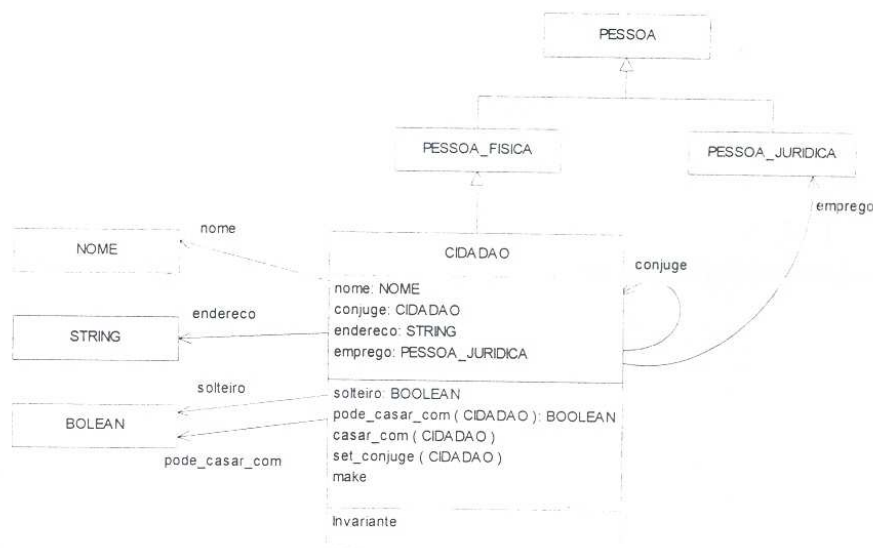


Figura 2: Classe CIDADAO

Basicamente, uma classe exibe três seções (a seção do invariante não é um padrão usual UML⁴⁹), como é ilustrado pela classe CIDADAO: (1) nome da classe – CIDADAO; (2) atributos – lista de atributos; (3) operações: lista de funções e operações; (4) invariante: condições booleanas caracterizadoras da estrutura. Para não sobrecarregar o exemplo, apenas a classe CIDADAO foi mostrada com todas as seções.

A Figura 2 apresenta o *diagrama de classes* da UML, que costuma ser imprópriamente denominado *modelo estático* na teoria-padrão da orientação a objetos. Ora, no momento em que o diagrama mostra as operações, não há como falar em “*modelo estático*”, uma vez que as transformações, a parte dinâmica, também estão representadas; assim, é mais apropriado utilizar a denominação *modelo estrutural* (aqui há uma confusão entre *modelo* e *diagrama*, que esclareceremos posteriormente, no próximo capítulo).

Para que o diagrama possa ser bem interpretado, é necessário desenvolver uma análise a respeito das associações que nele figuram; para fazê-lo, tomaremos para exemplo a associação emprego – as conclusões dela decorrentes serão válidas para as demais associações.

(1) Em primeiro lugar, devemos reconhecer que a associação está representada duas vezes, uma como atributo, outra sob a forma de uma seta⁵⁰; ou seja, a linguagem UML congrega dois modos de representação – gráfico e textual –, sendo, assim, um formalismo híbrido;

(2) De todo modo, a classe CIDADAO contém uma *referência* à classe PESSOA_JURIDICA, referência dada pelo atributo emprego;

⁴⁹ Embora a UML prescreva a possibilidade de criação de novas seções, como a aqui utilizada para representação do invariante.

⁵⁰ Ao invés de “seta”, pode-se também utilizar “arco orientado”; damos preferência ao primeiro termo para realçar o caráter intuitivo e visual de uma notação gráfica; ademais, também pretendemos sublinhar que, quanto a associações, o único conceito *teórico* com que estamos lidando é o de *atributo*, os demais sendo deixados à informalidade.

(3) A referência recebe uma segunda representação sob a forma de uma seta; neste caso, a direção da seta, indicada por sua ponta, é um elemento importante para a representação estrutural em formato gráfico, pois indica que classe contém a referência e que classe é referenciada⁵¹;

(4) O rótulo – emprego – colocado próximo à extremidade da seta, junto à classe referenciada, costuma ser denominado “*papel semântico*”; temos assim, o verdadeiro caráter de um papel semântico, do modo como é utilizado nos diagramas estruturais gráficos⁵²: trata-se de um atributo;

(5) O formalismo gráfico é efetivamente redundante: poder-se-ia utilizar unicamente o estilo textual; é preciso, entretanto, reconhecer que o formato gráfico é extremamente útil e intuitivo, facilitando a visualização das associações; porém, considerado o seu caráter redundante, pode-se reduzir o uso das setas a apenas o necessário a garantir a clareza e legibilidade dos diagramas⁵³.

⁵¹ Portanto, explicitar a direção da seta *não* constitui uma *opção de implementação*, como muitos pensam, nada tendo que ver com *caminhos de acesso* para otimização de processamento de bancos de dados; trata-se tão somente de uma definição estrutural, em nível puramente conceitual.

⁵² Não confundir com outro sentido de *papel semântico*, que será desenvolvido ao tratarmos de predicados.

⁵³ Uma ilustração do caráter intuitivo do formato gráfico é a facilidade de visualização de relacionamentos do tipo *agregação e composição*, os quais, no formato textual, são especificados por meio de invariantes de classes (neste caso, regras estruturais definidoras do tipo de associação) – a mesma observação vale para a definição de multiplicidade de associações. É interessante observar que cada classe (representada textual ou graficamente) é definidora, automaticamente, de uma associação *n*-ária (não binária, se $n > 2$), uma vez que seus atributos definem associações (efetivamente, o nome de um atributo é o nome de uma associação). De modo geral, no formato textual, as características das associações (isto é, atributos) são especificadas por meio de invariantes de classes. Esses invariantes possuem definições típicas, que os tornam prontamente identificáveis, embora sem o reconhecimento imediato do formato gráfico, a menos que se recorra a uma padronização de nomes ou comentários. Deve-se, assim, observar que o formalismo textual contém todo o potencial expressivo dos formalismos gráficos, ultrapassando-os, porém, nos aspectos de especificação de funcionalidade, ao permitir a definição de pré e pós-condições. Entretanto, o formalismo gráfico, embora com menor capacidade expressiva, considerados seus recursos atuais, exibe um apelo intuitivo bastante desejável à facilidade de uso. Quanto à modelagem de associações, é interessante observar que há uma alternativa à representação de associações diretamente como atributos (ou setas, dotadas de qualificativos ou não): as associações podem ser modeladas como estados de um conjunto de objetos (segundo o conceito de estado proposto neste trabalho). Este modo de se entender uma associação abre perspectivas não contempladas pelos métodos usuais (graficamente motivados): por exemplo, todas as associações tornam-se automaticamente históricas. Não pretendemos explorar semelhantes possibilidades neste trabalho, considerados

O formalismo gráfico da orientação a objetos constitui uma modalidade de rede semântica bastante bem sucedida. Entretanto, o formato textual de representação proporciona extraordinária precisão e capacidade expressiva à definição de modelos.

seus objetivos, mas apenas sublinhar a grande riqueza de idéias que se esconde por trás dos formalismos de representação, gráficos ou textuais.

6 O Modelo

6.1 O que é um Modelo

Se consideramos o material de que um modelo é feito, *modelo é um signo*; isto é, *uma forma material que participa de um ato de significação*. Neste sentido, modelo é um signo do qual se exige uma “certa semelhança”, ou uma certa “proximidade de funcionamento”, com relação ao objeto referenciado⁵⁴.

Mas, de certo, para nossas finalidades, modelo deve ser um signo dotado de alguma complexidade, e cujo interpretante possa ser compreendido como ligado a um conhecimento ou teoria. Assim, considerada sua complexidade, um modelo ultrapassa os limites de uma operação semiótica singular e espontânea⁵⁵ – como o é o ato semió-

⁵⁴ Uma classe especial de modelo é o *ícone*. *Ícone*, para Peirce, é um signo que, por alguma propriedade, exibe uma semelhança com o seu objeto semiótico.

⁵⁵ O Prof. Gudwin, com veemência, discorda de tal abordagem! Segundo ele, um modelo “não ultrapassa, mas sim exige que o processo de semiose disponibilize toda sua potencialidade!”. De um modo geral, o Prof. Gudwin considera, com razão, que o tratamento por nós dado à conceituação de modelo deve-se mais a Umberto Eco do que a Peirce.

tico elementar –, precisando ser considerado como uma construção, em sentido mais amplo. E, como participante do ato semiótico, um modelo compartilha a sorte dos demais signos compartilhados: é uma convenção cultural – um modelo deve sua existência a convenções que só fazem sentido no seio de uma comunidade.

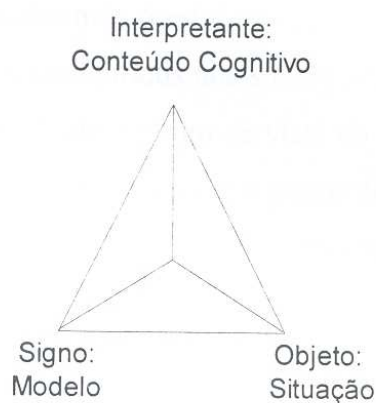


Figura 3: Modelo e Semiose

Assim, como construção cultural, modelo não pode estar associado a conhecimento consistente ou completo, nem mesmo a conhecimento lógico – modelo como expressão de conhecimento científico, como teoria “lógica”, é apenas um caso particular. Portanto, mesmo para sistemas de informações, um modelo representa “convenções” de determinadas sociedades, nada – ou quase nada – tendo que ver com “espelho da realidade”. A semiótica mais recente de Umberto Eco [Eco76] – posterior àquela em que ele se dispunha a “*exorcizar a erva daninha do referente*” [Eco64] – desenvolve-se nesse sentido. Eco frisa a importância de considerar-se um signo como uma unidade cultural; neste sentido, modelo é a unidade cultural por excelência. Como conhecimento formalizado, aplica-se ao modelo todo o aparato conceitual desenvolvido para análise do conhecimento, como é o caso, por exemplo, da filosofia de Popper [Popper63] [Popper71].

Para diferenciar modelo de um signo qualquer, propomos que *modelo* seja entendido como um *signo dotado de estrutura*.

No capítulo precedente, estrutura foi abordada sob o ponto de vista operatório e relacional, sem fazer recurso explícito a significado; deste modo, pode-se falar em

estrutura do signo, sem que daí resulte que a estrutura deva sua existência “lógica” ao significado do signo, mas que seja, ela própria, condição prévia para a formação do significado. Assim, em um modelo, encontram-se as teorias semióticas e as teorias ontológicas (estruturais): significado e funcionalidade reunidos em um mesmo local.

Até este momento, analisamos o estatuto “formal” de um modelo, como integrante da semiose. Entretanto, como todos nós somos produtores e usuários de modelos, é preciso analisar modelo desde o ponto de vista do intérprete, aquele lugar lógico onde está situado o interpretante; mas sob o ponto de vista do intérprete especial que é o ser humano. Ou seja, precisamos situar *modelo* no interior do *processo cognitivo humano*⁵⁶.

Lévi-Strauss, segundo Piaget [Piag79], pag. 87, ao refletir sobre “*o primeiro princípio fundamental ... que, por detrás das relações ‘concretas’, buscará a estrutura subjacente ... que só pode ser obtida pela construção dedutiva dos modelos abstratos.*” (negrito nosso), situou, de forma lapidar, o modo como um modelo se insere no processo cognitivo humano – *Anthropologie structurale*, pag. 28, in [Piag79], pag. 87 :

Se, como cremos, a atividade inconsciente do espírito consiste em impor formas a um conteúdo e se essas formas são fundamentalmente as mesmas para todos os espíritos, antigos e modernos, primitivos e civilizados ... é necessário e suficiente atingir a estrutura inconsciente, subjacente a cada instituição e a cada costume, para obter um princípio de explicação válido para outras instituições e outros costumes, com a condição, naturalmente de prolongar bastante a análise.

O texto de Lévi-Strauss ensina que: (a) *modelo é uma forma imposta a um conteúdo* (cognitivo); (b) *modelar, ou representar, é uma atividade inconsciente do espírito humano*; (c) *existem padrões fundamentais de modelos – princípio de explicação válido para outras instituições e outros costumes* – que independem de cultura.

⁵⁶ Segundo nosso entendimento, um interpretante semiótico pertence a um processo (cognitivo) situado nas estruturas cognitivas de um intérprete; com base nesta idéia (e em outras contidas neste trabalho), definimos e implementamos um modelo de agente cognitivo, em pesquisa a ser aprofundada futuramente.

Conceituar modelo como forma *imposta a um conteúdo* é compatível com a visão peirceana da semiose, em que o signo contribui para “formar” o interpretante, sem contudo o determinar: *modelo* é a *forma* (signo), *interpretante* é o *conteúdo* cognitivo, sem que se especifique de que modo a forma se impõe ao conteúdo. Entender um modelo como *princípio de explicação* derivado de uma *estrutura inconsciente* equivale a realçar o caráter estrutural de um signo, o que condiz com nossa proposta de considerar *modelo* como uma *estrutura dotada de significado* (ou, em termos mais precisos, *signo dotado de estrutura*).

A discussão sobre modelo, tomando por base Lévi-Strauss, passou-se em linhas quase exclusivamente saussureanas, valorizando a relação *signo-interpretante* (ou, *significante-significado*, segundo Saussure), sem menção ao *objeto semiótico* (*referente*, para a atual escola saussureana, após Jakobson). Entretanto, na tríade peirceana, o *objeto semiótico* é precisamente o *resultado lógico* da ação conjunta dos dois outros elementos (*signo* e *interpretante*), dos quais, contudo, não se separa (embora sendo *resultado*). Assim, o *objeto semiótico* esteve presente em toda a discussão sobre modelo, ainda que não mencionado.

Existe um fato que necessita ser esclarecido, para que haja compreensão adequada a respeito do objeto semiótico: o objeto é *produzido* pelo ato de significação, não podendo ser confundido com um objeto pertencente ao mundo físico exterior ao intérprete. Tal fato ocorre tanto para intérpretes humanos quanto para intérpretes computacionais – em que o objeto é produzido dentro das estruturas cognitivas do intérprete, como configuração de memória (humana ou computacional); em um segundo momento, o objeto poderá ser associado ao que se entende serem objetos “do mundo”⁵⁷, por meio de canais de transmissão de sinais e outros processadores alheios ao fato semiótico.

⁵⁷ Para interpretação do conceito de *objeto semiótico* aqui exposto, solicitamos ao leitor rever a leitura de **4.3 Aplicação do Conceito de Semiose**, sobretudo as notas explicativas. Do ponto de vista de uma estrutura cognitiva, o *objeto semiótico* não pode estar situado externamente (nada garante a existência de um objeto externo, embora se possa falar de objeto segundo um ato semiótico situado no interior da estrutura cognitiva), portanto é necessário distinguir o objeto *produzido* pelo ato de signifi-

6.2 Modelo e Realidade

Um modelo expressa conhecimento, convenções, fatos culturais, crenças, nada se podendo afirmar a respeito da “realidade” desses conhecimentos. Talvez a refutação mais direta a respeito de um modelo (como forma de teoria) efetivamente representar a realidade tenha vindo de Karl Popper [Popper63] por meio de uma argumentação que, aproximadamente, desenvolve-se do seguinte modo: só se pode afirmar que uma teoria é verdadeira se ela apresentar respostas corretas (concordantes com os fatos) às perguntas que lhe são dirigidas; ora, a quantidade de perguntas potencialmente existentes é infinita, não havendo qualquer critério para fixação de um número adequado; como não é possível formular infinitas perguntas para teste de uma teoria, segue-se que, mesmo que uma teoria seja verdadeira, jamais o saberemos. Sabe-se que Popper foi demasiadamente otimista a respeito de seus critérios para julgar “respostas corretas” e “concordância com os fatos”, de todo modo, em linhas gerais, sua argumentação expressa bem a impossibilidade de se julgar a verdade empírica de uma teoria, e, por esta via, de um modelo.

Ainda que um modelo não seja um meio para exprimir verdades absolutas, pode-se pretender que ele seja uma construção proposta para atender a determinadas finalidades. A esse respeito há uma interessante citação de Peirce em [Eco79a], pag. 27: *“A idéia de significado é tal que envolve alguma referência a um propósito”*. Eco continua, então: *“... o realismo ... de Peirce deve ser visto na perspectiva do seu pragmatismo: a realidade não é um simples Dado, mas antes um Resultado. Para compreender o que o significado de um signo deve produzir como Resultado, eis a noção de Interpretante Final.”*. Passagem importantíssima sobre a “realidade” como

cação (interno à estrutura cognitiva) em contraposição ao objeto *presenciado*, que é um objeto hipotético, externo à estrutura cognitiva. Essa associação, de um objeto *produzido internamente* com referência a um *objeto presumido* externo passa-se em um “segundo momento”, uma vez que a associação pressupõe a existência do objeto interno à estrutura cognitiva e se processa por meio de mecanismos alheios ao ato semiótico, através de dispositivos não diretamente pertencentes à estrutura cognitiva (canais de transmissão e outros dispositivos). Toda a base de nossa argumentação baseia-se na auto-

“resultado”, idéia que talvez mostre a fonte do “resultado” da “conta-por-um” de Badiou [Badiou96], que também produz a “realidade como resultado”. Podemos estar diante não apenas da gênese de algumas das idéias de Eco, mas também das de Badiou, em um ponto fundamental. Também a “operação fundamental do espírito” de Piaget, que é aplicar “uma forma a um conteúdo” [Piag79] enquadra-se sob esse princípio proposto por Peirce.

Esta não é apenas uma posição metodológica – modelo como resultado, não como expressão da “verdade”: um modelo, como expressão de um conhecimento, está profunda e intrinsecamente formado pela cultura, pelo contexto, pelos meios de representação e pelas finalidades (e por mais alguns fatores). Assim, podemos nos afastar da idéia de um modelo universal. Em um sentido amplo, este tema é desenvolvido com grande propriedade em *A Estrutura das Revoluções Científicas*, por Thomas Kuhn [Kuhn62], que mostra o caráter inseparável das teorias e das visões do mundo; nesse livro Kuhn ampliou idéias do grande lingüista Edward Sapir, em sua tese a respeito da incomensurabilidade das visões do mundo, condicionadas pelas estruturas lingüísticas. Os aspectos mais radicais da proposta de Sapir são questionáveis, porém um fato pode ser dado como aceito: o modo como um modelo é formulado depende inelutavelmente das finalidades e dos vários fatores ambientais e culturais.

Deste modo, pode-se aceitar que o critério adequado para julgamento da “objetividade” de um modelo é o da adequação às finalidades, conforme os meios disponíveis – ainda que se “creia” que o modelo *espelha* a realidade (a saudável atitude de todo aquele que investiga um domínio). Vieram de Karl Popper, em *Conjecturas e Refutações* [Popper63], os critérios para se julgar a adequação de um modelo – critérios análogos aos já citados para os testes de validade de uma teoria: todo conhecimento, em sua origem, é hipotético e tem como finalidade solucionar problemas. O único modo de testar a validade de uma teoria (ou seja, um modelo) é por meio de

nomia da estrutura cognitiva (sede do ato semiótico), com relação ao que quer que se entenda ser o ambiente a ela externo.

testes de refutação; como não se podem fazer os infinitos testes potencialmente existentes, segue-se que não existe o conceito de teoria correta, apenas o de teoria confiável (aquela que mostrou-se bem sucedida em uma quantidade “suficientemente grande” de teste de refutação).

Kuhn mostrou que as idéias de objetividade de Popper (aí também incluída a objetividade do que se entende por refutação de uma teoria) são demasiadamente idealistas, não tão “objetivas”. Entretanto, pode-se tomar como bem fundamentado que todo conhecimento é originalmente hipotético, e que não existe o conceito de conhecimento “correto”, mas apenas confiável. Se também dermos por estabelecido que todo conhecimento é expressão de uma visão de mundo, segundo Sapir, podemos abandonar as idéias de modelo universal e de modelo perfeito (segundo Popper, mesmo se uma teoria fosse perfeita, jamais o saberíamos, dada a impossibilidade da realização de infinitos testes). Ou seja, podemos trabalhar de modo mais humildemente humano, aceitando modelos adequados a finalidades, dependentes de ambiente, resultantes de nossa experiência – e que de modo algum são infensos a revisões.

6.3 Crítica: Modelo, Ontologia e Base de Conhecimento.

Este tópico mostra como um trabalho atual [SwTa99], publicado em um dos principais jornais científicos – IEEE –, exhibe uma conceituação passível de reparos, o que só se torna visível em decorrência de uma análise conceitual a partir dos fundamentos, como a feita neste trabalho.

A expressão *ontologia* tem adquirido relevância na área de inteligência artificial como “um conjunto de conceitos ou termos que podem ser usados para descrever alguma área do conhecimento ou construir uma representação dela.”, [SwTa99]. Neste sentido, *ontologia* é sinônimo de *modelo*. Porém, o modo como os termos *base de conhecimento* e *ontologia* têm sido usados em conjunto mostra a necessidade de um esclarecimento de conceitos a partir dos fundamentos. Vejamos: “... uma ontolo-

gia provê uma estrutura básica ou armadura em volta da qual uma base de conhecimento pode ser construída. Uma ontologia provê um conjunto de conceitos e termos para descrever algum domínio, enquanto que uma base de conhecimento usa aqueles termos para representar o que é verdade sobre algum mundo real ou hipotético. Assim, uma ontologia médica pode conter definições de termos como “leucemia” ou “doença terminal”, mas não contém asserções de que um determinado paciente tem alguma doença, embora uma base de conhecimento deva.”, [SwTa99].

Segundo a proposta acima, *ontologia* possui um uso mais geral do que *base-de-conhecimento*. Ora, um ambiente de enfermaria é um domínio tão legítimo quanto o é uma descrição mais genérica da área médica, não sendo visível o motivo pelo qual o fato de um doente apresentar uma moléstia não possa figura em uma ontologia da enfermaria. A questão da generalidade é discutível e pode tornar inútil a diferenciação entre ontologia e base-de-conhecimento: em ambiente hospitalar, provavelmente, enfermarias são um assunto tão genérico quanto o são os demais temas médicos. Assim, caracterizar ontologia como hierarquicamente mais genérica é uma questão de senso estético, não podendo servir de base para uma caracterização teórica, ou mesmo prática. Portanto, a “verdadeira” distinção entre ontologia e base-de-conhecimento é que aquela descreve “*algum domínio*”, enquanto que essa representa “*o que é verdade sobre algum mundo real ou hipotético*”. Ora, mesmo no sentido de [SwTa99], do qual ainda não nos afastamos, ontologia e base-de-conhecimento têm a mesma utilidade: representar (ou descrever). Deste modo, a chamada *base-de-conhecimento* não descreve nenhuma *verdade* ao representar que um certo paciente apresenta uma certa doença (pois a enfermaria pode estar vazia! – sem que tal fato invalide a representação). Portanto, não se pode afirmar que uma base-de-conhecimento descreva uma verdade, em sentido *real* (um doente *real*); portanto, a “verdade” da base-de-conhecimento é a verdade *hipotética* que lhe restou, de um *possível* doente. Ora, por raciocínios análogos pode-se mostrar que também uma ontologia pode descrever “verdades” ao descrever doenças, ficando para o progresso da medicina dizer se são doenças “reais”, ou mesmo se “existem” doenças. Ou seja, no sentido de [SwTa99], ontologias ou bases-de-conhecimento podem ambas descrever verdades ou não – estando iguais neste sentido. Assim, também não há diferenciação entre ontologia e

base-de-conhecimento com base na “verdade” que uma delas descreva (na realidade, nem ontologias nem bases-de-conhecimento descrevem verdades, mas apenas *crenças* em verdades, sempre passíveis de questionamento empírico ou em um mundo possível). Entretanto, há um sentido em que uma base-de-conhecimento pode descrever uma “verdade” – é quando *realmente* há um doente internado e os dados do doente são registrados “de acordo com a descrição”. Ou seja, a “verdade” reside, primeiramente no objeto (pessoa internada) e, num segundo momento, na interpretação dessa pessoa como sendo um doente descrito. Este raciocínio vale também para mostrar a verdade de uma doença. Portanto, nem ontologias nem bases-de-conhecimento, no sentido de [SwTa99] podem ser diferenciadas por uma suposta hierarquia conceitual, nem pela verdade que descrevem.

Deste modo, propomos que o termo *ontologia* seja aplicável apenas a descrições genéricas – aquilo a que chamamos *modelo intencional* – enquanto que *base-de-conhecimento* seja reservado ao *modelo extensional*, comumente em *formato armazenado* em um banco de dados (podendo haver bases-de-conhecimento de doentes, como também de doenças).

Este é um exemplo de como a terminologia usada nas várias áreas dos sistemas de informação tem-se ressentido da falta de uma análise de conceitos a partir dos fundamentos, análise que forneça critérios que permitam ordenar conceitos de um modo coerente e menos impressionista do que tem ocorrido habitualmente.

Há também um outro sentido em que o conceito de ontologia precisa ser revisito. Como observamos acima, o termo ontologia pode ser utilizado para descrever uma área do conhecimento, o que quer que uma tal área seja, e, como também já observado, segundo essa aceção, ontologia é sinônimo de modelo intencional. Existe, entretanto, a crença de que uma ontologia representa os aspectos estáticos de um domínio, sendo uma teoria “do conteúdo”, em contraposição a teorias que descrevem os aspectos dinâmicos das áreas representadas, ou teorias “dos mecanismos” – “*Ontologias são quintessencialmente teorias de conteúdo, porque sua principal contribuição é identificar classes específicas de objetos e relações que existem em algum domínio.*” [SwTa99]. De acordo com essa abordagem, os “mecanismos” (regras, “frame language

ges”, redes neurais etc.) são representados por outros tipos de teorias, sendo uma alternativa ao uso de ontologias. Esta separação em *teorias estáticas* (cf. ontologias) e *teorias dinâmicas* (cf sistemas de regras) padece de miopia: os modelos representam *tipos de objetos*, mas também descrevem e executam *transformações*, o que permite dotar seus objetos de capacidade de gerar transformações; assim, não apenas objetos individuais são naturalmente dotados de capacidade operatória, mas também podem ser definidos objetos complexos, como redes neurais, que são, eles próprios ontologias altamente dinâmicas. Ou seja, ontologias, de acordo com os princípios deste trabalho, expressam, ao mesmo tempo, teorias estáticas e dinâmicas, não havendo a dicotomia “teorias de conteúdo” e “teorias de mecanismos” (como afirmado em [Sw-Ta99]). Este trabalho apresenta uma ontologia de redes neurais dotada de grande capacidade operatória; uma rede neural, definida por essa ontologia, pode ser incluída como atributo de um (tipo de objeto) *professor*, por exemplo, dotando-o de capacidade de tomar decisões. Portanto, uma ontologia especifica tantos os aspectos estáticos quanto dinâmicos de um domínio, de modo integrado. A aparente separação entre as duas abordagens (identificar *conteúdo* com *estática* e *mecanismo* com *dinâmica*), na realidade, resulta de fatores históricos, de pesquisadores filiados a diferentes tradições culturais que se desenvolveram separadamente, mas que, na realidade, se abordadas com os métodos adequados de conceituação (e representação) constituem um pseudoproblema. Esta miopia resulta da adesão excessiva a determinadas técnicas e ferramentas, e a uma atitude demasiadamente estreita com relação ao que significa modelar um domínio (ato que não se restringe a formular uma taxionomia⁵⁸).

⁵⁸ Entendendo-se *taxionomia* como *classificação hierárquica*

Parte IV – Fenomenologia

Examinai tudo com discernimento: conservai o que é bom.

1 Tes 5, 21

... e a Verdade é apenas a memória de tudo quanto se encontrou pelo caminho.

Maurice Merleau-Ponty, Signos

“indicar o objeto de que se fala”) e *predicado* (“afirmar algo a seu respeito”) [Di-EL72], pag. 200. Ou seja, qualquer sentença declarativa (enunciado) possui dois termos: (1) um *objeto* – o *sujeito* – que é o foco da atenção e do qual se diz algo; e (2) tudo o mais que é dito (a respeito do sujeito) – o *predicado*.

Esta dicotomia *sujeito/predicado* influenciou, desde a antiguidade até nossos dias, todas as disciplinas formais, não apenas a lógica e a gramática. De acordo com o conceito, predicado é algo que se atribui ao sujeito.

Noam Chomsky, por exemplo, retoma a dicotomia, para fazer dela a base de toda a sua teoria da sintaxe (isto é, de toda a sua teoria da linguagem). Seu postulado fundamental – $S \rightarrow SN\ SV$ –, para a estrutura de uma sentença, é uma reinterpretação do conceito aristotélico. O esquema afirma que S – *sentença* – pode ser substituída pela seqüência $SN\ SV$ – respectivamente, *sintagma nominal* e *sintagma verbal*. A Figura 4 ilustra graficamente este modo de organizar uma sentença, aplicado a “João deu um livro a Maria”. Implicitamente, tem-se que o sintagma nominal, SN , quando ligado diretamente à base da sentença, S , faz o papel de sujeito – e tudo o mais é o predicado.



Figura 4: Sentença tradicional

Portanto, não é de estranhar que a idéia de que os objetos *possuem* propriedades seja encarada como natural e fora de questão pela literatura técnica de informática: “*Objects (also) have observable properties*”, [Cook94], pag. 29.

Na realidade, a concepção de Aristóteles é muito mais ampla e sutil do que esta compreensão largamente difundida⁵⁹. Entretanto, dadas as finalidades do trabalho, não poderemos abordar determinados matizes do conceito aristotélico, por não ser possível mostrar suas conexões com as questões aqui levantadas, cujo tratamento é primordialmente técnico.

7.2 Conceito Lógico de Predicado

Frege apartou-se da análise lingüística, baseada na separação *sujeito/predicado*, inaugurando o conceito moderno de predicado (lógico). E o fez ao definir predicado como um tipo especial de função. Assim, o predicado *Obeso* (*João*) é um modo particular de associar *João* a um dos valores *V* – verdadeiro – ou *F* – falso. No caso geral, *Obeso* (*X*) é outra forma de se representar *Obeso*: $X \rightarrow \{ V, F \}$. O mesmo vale para predicados com mais de um argumento, como *Maior_que* (*X*, *Y*), que define uma relação entre *X* e *Y*, no caso, convencionalmente, *X* ser maior do que *Y*. Nesses exemplos, (1) caso João efetivamente possa contar-se entre os obesos, temos *Obeso*: *João* $\rightarrow V$, e (2) *Maior_que*: (*4*, *5*) $\rightarrow F$. Por essas considerações, segundo Frege, predicados particulares (isto é, instâncias de predicados), como *Obeso* (*João*) e *4 > 5* são nomes de um valor-verdade, *V* ou *F*.

O tratamento que Frege aplica aos predicados faz parte do esforço empreendido por esse grande lógico para representar, talvez compreender, o mundo. Portanto, o predicado de Frege precisa ser entendido como um modo de modelar a realidade, empírica ou abstrata⁶⁰, que nos cerca, e não como simbolismo destituído de sentido, ao estilo da matemática puramente formal e simbólica.

⁵⁹ O conceito de Aristóteles tem em vista a distinção *tema/rema*, da qual a dicotomia *sujeito/predicado* é um dos aspectos particulares. O contraste *tema/rema* está na base da fundamentação lingüística do conceito de *estado* que desenvolvemos neste trabalho.

⁶⁰ Talvez a dicotomia de partida para um lógico.

Entretanto, embora a concepção lógica de Frege mostre-se particularmente feliz para representar verdades eternas, como *Maior_que*: $4 > 5 \rightarrow F$ ou *Todos os homens são mortais*, *Sócrates é homem*, *logo ...*, deixa muitas questões em aberto quando trata de asserções como *Obeso (João)*, cujo valor-verdade – e a própria condição de verdade – pode variar com o tempo e que não oferece meios para indicar em que consiste efetivamente a obesidade de João – elementos primordiais à representação de verdades transitórias e circunstanciais. Portanto, temos de nos aprofundar um pouco nas idéias de Frege se quisermos nos beneficiar de seu conceito de predicado para a representação de conhecimento.

Em primeiro lugar, é preciso ter em vista a atitude de um lógico ante o fenómeno da representação e da significação. Para tanto, analisemos o conceito fregeano de predicado no contexto da teoria semiótica de Peirce, como mostrado na Figura 5.

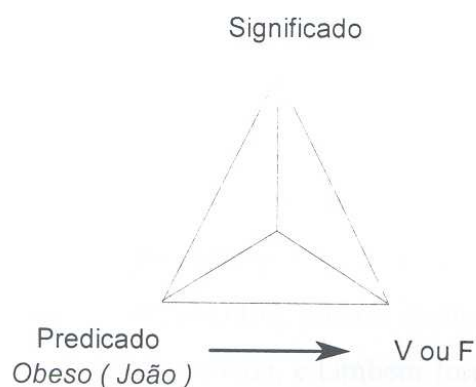


Figura 5: Predicado lógico

De acordo com a figura, a expressão significativa do predicado – *Obeso (João)* – é aplicada a determinados objetos, no caso, um valor-verdade (*V* ou *F*), deixando claramente à parte, como não problemática, qualquer questão relacionada a significado. Acostumados às verdades eternas, que parecem derivadas diretamente dos objetos, independentes do observador, os lógicos movem-se com firmeza sobre conceitos que podem ser expressos em termos de extensão. Não por acaso, Frege evita a noção

movediça de significado e utiliza a idéia segura de função, assim aplicando um *objeto* (expressão do predicado) a outro *objeto* (valor-verdade). Ou seja, considerados os elementos envolvidos no processo de significação⁶¹, existe uma marcada preferência pelos dois elementos dotados de extensão – o *signo* e o *objeto* – e uma virtual exclusão do elemento não formalizado pela lógica, o *significado*.

Entretanto, para representar conhecimento, e para tornar a representação operacional, sob a forma de sistema de informação (software), um problema torna-se ineludível: como codificar, de algum modo, o significado pretendido, por meio da forma que é o predicado.

Assim, o tratamento dado pela lógica aos predicado – como uma função que aplica valores argumentais a valores-verdade, a partir de um sentido que permanece implícito – é claramente insuficiente às finalidades da representação de conhecimento. Não apenas o sentido/significado é excluído da representação, mas também o predicado lógico não representa de modo explícito as condições em que o predicado/enunciado, ele próprio, pode ocorrer (instanciação do enunciado).

Nossa investigação gira em torno de como dotar o próprio predicado dos meios necessários à formalização da semântica. Essas características diferenciam um predicado genérico (que estamos investigando) de um predicado lógico: a estrutura do predicado, além da exibição dos argumentos, precisa explicitar as circunstâncias em que a instância do predicado pode ser criada, e também formalizar a condição lógica que caracteriza o predicado.

⁶¹ No caso, a tríade da semiose peirceana: *signo*, *interpretante*, *objeto*.

7.3 Conceito Semântico de Predicado

De acordo o conceito que adotamos, predicado é uma expressão sempre ligada a um significado, e, ademais, deve proporcionar meios para formalizar uma parte desse significado. Assim, o predicado “*João deu um livro a Maria*”, que provisoriamente⁶² reescreveremos *Deu* (*João*, *livro*, *Maria*), será entendido de acordo com a Figura 6.

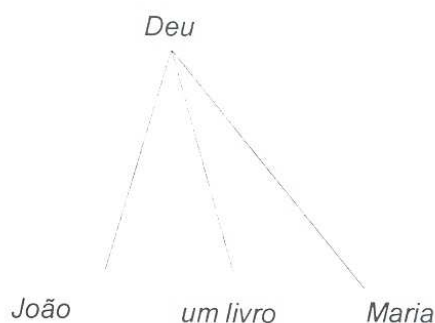


Figura 6: Predicado e seus componentes

A figura mostra:

- O predicado: *Deu* – organizador da predicação e portador do (de uma parte do) significado da predicação;
- E seus dependentes imediatos: *João*; *um livro*; *Maria* – que exercem funções que completam o sentido do predicado (*João*: *agente*; *livro*: *objeto*; *Maria*: *beneficiário*)

Este modo de representar um ato de significação, por meio de um predicado e de um elenco de termos que completam seu sentido, aparta-se da dicotomia tradicional *sujeito/predicado* – e é especialmente indicado à nossa finalidade de representação de conhecimento, exatamente pela ênfase que atribui ao conteúdo semântico. Cada termo ligado ao predicado nele exerce um *papel semântico* – no exemplo acima:

João exerce o papel de *agente*; livro, o papel de *objeto*; Maria, o de *beneficiário*. Um fato a ser destacado é que não se atribui função privilegiada a nenhum elemento, como ocorre ao *sujeito* da gramática tradicional. Esta é uma decisão plena de implicações que não poderão ser analisadas aqui, mas que, como mínimo, libera o predicado, como formalizador de um ato de significação, de estar comprometido com um conteúdo semântico particular, como é o caso do sujeito⁶³.

Um outro exemplo, mais completo, de representação de um predicado, juntamente com seus argumentos, em que são exibidos os papéis semânticos, é “*Hoje Maria envia o relatório para o cliente de Araucária*” – *Envia* (*Maria, relatório, cliente, Araucária, hoje*) – em que a estrutura de papéis semânticos encontra-se exemplificada na Figura 7.

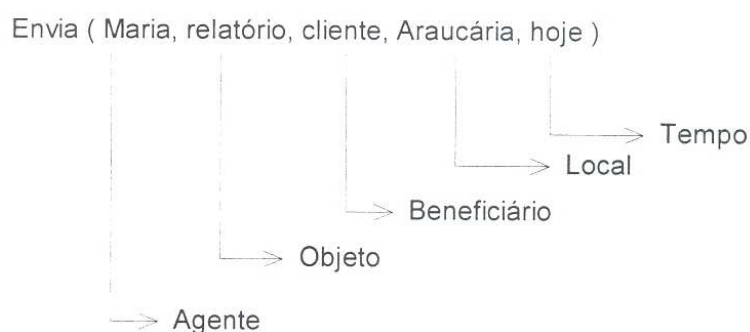


Figura 7: Predicado e papéis semânticos

A figura exhibe uma estrutura extremamente simplificada de papéis semânticos, apenas a título de ilustração.

Este modo de representação de predicado, basicamente semântico, é utilizado por diversas correntes de pensamento lingüístico, como nas várias versões das gramá-

⁶² Até analisarmos o modo de interpretar um predicado.

⁶³ Na realidade, o sujeito da gramática tradicional está associado a uma grande quantidade de noções, como se observa em: (1) “João fechou a porta” – “João” é *agente*; (2) “A chave fechou a porta” – “chave” é *instrumento*; (3) “A porta fechou-se” – “porta” é *paciente*, sem designação de agente. Os exemplos mostram a inadequação de se tomar a forma gramatical superficial como base para formalização de conhecimento, como fazem determinadas metodologias.

ticas de casos – [Ander71], [Fill68] –, gramáticas funcionais – [Tesnière] – e, em geral, em abordagens de base semântica, como em [Chafe75].

Ademais, esta representação dos predicados é compatível com a interpretação dos predicados como é feita na lógica formal, em que a expressão *Deu (João, livro, Maria)* encontra paralelo exato em um predicado lógico escrito sob a forma $P(x_1, x_2, \dots, x_n)$. O paralelo vai além da mera forma de se escrever um predicado, alcançando mesmo a compreensão do predicado como função lógica: “*A proposição constitui a unidade semântico-pragmática de uma língua natural, dado que só nela – nas suas formas completas ou nas suas formas reduzidas, como um mero **sim** ou um mero **não** – se realiza a totalidade das funções lingüísticas*” [Peres84], pag. 36. (No original, *proposição* corresponde a *predicado/predicador*, segundo estamos usando neste trabalho; os negritos são acréscimos nossos.)

Deve-se observar que, na teoria lingüística, o que estamos chamando aqui de *predicado*, para evitar ambigüidades, sobretudo com o conceito tradicional de predicado da gramática tradicional, muitas vezes é denominado *predicador*. Também deve-se ter em conta que o conceito de *predicador*, na lingüística, não se restringe a verbos, podendo aplicar-se também a adjetivos e a expressões relacionais como *Pai_de*.

Pode-se, assim, afirmar que o conceito lingüístico de predicado, utilizado por numerosas correntes lingüísticas que valorizam os aspectos semânticos, é compatível com o conceito lógico de predicado (predicado como função lógica), porém alargando este último com um riquíssimo tratamento relacionado com significado, como se torna patente pelas análises que abordam os papéis semânticos.

O conceito de predicado desenvolvido neste trabalho muito deve aos resultados da lógica e da lingüística. Nossa abordagem interpreta esses resultados de um modo que, acreditamos, contenha algo de pessoal, e acrescenta um tratamento que tem permanecido alheio às ciências mais tradicionais, mediante o uso de técnicas formais pertencentes às ciências da informação.

Todo o tema possui amplas implicações teóricas e práticas na representação de conhecimento. Muito da pesquisa lingüística, no campo das linguagens naturais, gira em torno dos papéis semânticos e este conceito é essencial para o uso dos predicados na representação do conhecimento, sobretudo na Inteligência Artificial. Não enveredaremos pelo tema neste relato, uma vez que pretendemos fazê-lo em uma investigação mais ampla, na formalização do conceito de predicado para uso em processos gerais de formalização de conhecimento, em seguida ao atual trabalho. Entretanto, até o momento, temos realizado algumas incursões de aplicação da idéia. Um dos projetos, de modelagem de um Sistema Mercadológico de uma empresa de consultoria e assistência técnica (a ser exemplificado mais detalhadamente em capítulo posterior), mostra como o conceito de predicado e papéis semânticos facilitou a estruturação das funções internas, por meio de um projeto inteiramente desenvolvido sobre o conceito formal de predicado

8 Interpretação de um Predicado

Abordaremos, agora, a questão de como um predicado deve ser entendido. Antes de fazê-lo, porém, é necessário deixar claro o papel dos predicados nos processos de representação de conhecimento.

Para as análises que se seguem, é preciso ter em conta que estamos nos apartando da interpretação que a gramática tradicional faz de predicado. Ou seja, nos afastamos da noção de que, em uma asserção, predicado é tudo aquilo que não é o sujeito. Para nós, os predicados possuem a forma $P(x_1, x_2, \dots, x_n)$, de acordo com o formalismo lógico, e também, em linhas gerais, de acordo com a interpretação que as várias correntes da gramática funcional fazem de predicado. Assim, a asserção “*O cliente quita o débito*” não será decomposta em (*sujeito = o cliente*) e (*predicado = quita o débito*); mas, provisoriamente, será formalizada por *Quitar(cliente, débito)*.

Existem muitos modos de se formalizar um modelo e, neste particular, a área de sistemas de informação é especialmente rica, bastando recordar os modelos “*relacionais*”, “*E/R*”, “*de objetos*”, aos quais se somam “*DFDs*” “*diagramas de atividades*” e toda a quantidade de formalismos cuja diversidade as mais recentes propostas de padronização não fazem mais do que consagrar. A variedade dos modos de repre-

sentação reflete a diversidade de culturas técnicas e científicas: *gramáticas, sistemas de "frames", redes semânticas* – e muitos outros formalismos – são representantes bem sucedidos dos diferentes recursos que as várias áreas do conhecimento utilizam para representar seus objetos de estudo.

Entretanto, qualquer que seja o formalismo, os objetos por eles representados também podem ser representados sob a forma de predicados⁶⁴, em um formato mais simples e menos dependente de formatos gráficos.

Deste modo, pelos motivos acima, *um predicado é um formalismo mínimo de representação*; e, ademais, é um formalismo suficientemente abstrato para proporcionar distanciamento de hábitos adquiridos ao lidar com representações predominantemente gráficas⁶⁵. Deste modo, consideramos que: (1) toda formalização de conhecimento pode ser feita por meio de predicados; (2) dentre os formalismos disponíveis, os predicados são o modo de representação mais simples; e (3) os vários formalismos são diferentes formas de os predicados se manifestarem⁶⁶.

Agora, voltemos a considerar a asserção “*O cliente quita o débito*”, que “provisoriamente” foi traduzida em termos do predicado *Quitar(cliente, débito)*. Mas, se estamos empregando uma doutrina bem fundamentada, que conjuga lógica e teorias lingüísticas modernas, por quê a ressalva “provisoriamente”? Para responder é necessário retornar a um estágio anterior, dando uma explicação que ainda estamos devendo: em que consiste, precisamente, a fraqueza da dicotomia tradicional *sujeito/predicado* (fraqueza que nos levou a adotar ponto de vista alternativo)?

⁶⁴ No sentido que estamos empregando, um predicado reduz-se a uma expressão formal, sem ser considerada explicitamente a proposição a ele correspondente; parte considerável das idéias deste trabalho consiste em codificar semântica diretamente na forma dos predicados, sem recurso a outro meio de representação de significado.

⁶⁵ É o que ocorre, por exemplo, quando se consideram as associações binárias como fundamentais – não há qualquer motivo para fazê-lo, exceto o condicionamento adquirido no uso contínuo de representações gráficas.

⁶⁶ Segundo o conceito que advogamos, predicados codificam operações e funções, modelando também os aspectos comportamentais. Paralelamente a este relato, estamos investigando um conceito mais amplo de predicado, que faz dele uma expressão “não linear”, deste modo tornando-o particularmente adequado (completo) à modelagem de conhecimento.

Segundo orientações metodológicas de ampla circulação⁶⁷, a forma gramatical de uma asserção proporciona um modo de detecção dos objetos de interesse e de ter-se informações seguras a respeito da estrutura desses objetos. Assim, a afirmação “*O cliente quita o débito*” informa que existe um objeto de interesse – *cliente* – e que uma das operações desempenhadas por um cliente é *quitar débitos*. Uma tal visão dos fatos pode ser representada por meio de uma classe da orientação a objetos, como é mostrado na Figura 8, que descreve um cliente genérico, um de seus atributos e as operações por que ele é responsável (entre as quais, *Quita débito*).

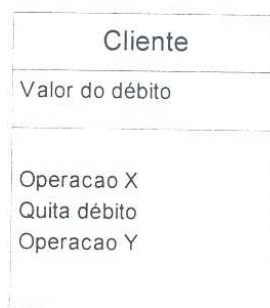


Figura 8: Classe, atributos e operações

Porém, acabamos de cair em uma cilada verbal (ao representar *Quita débito* como operação de *Cliente*)! O débito, eventualmente, pode ser quitado por algum outro objeto⁶⁸, que não seja exatamente um cliente, nem mesmo o próprio cliente beneficiado pela quitação – o que obriga a procurar por outros tipos de objetos que também possam “quitar débitos” de clientes. Este tipo de abordagem, advindo de uma verbalização inadequada das questões a resolver, complica extraordinariamente as soluções, e, pior, deixa o responsável pela modelagem sem critérios seguros de análise.

A inadequação da solução ocorre em consequência de um pressuposto: a análise gira em torno de *objetos* e de *propriedades observáveis de objetos*. O pressuposto é praticado implícita e informalmente por muitos modeladores, mas, também, é ado-

⁶⁷ Caso da metodologia OMT [Rumb94], que recomenda uma formulação literária dos problemas a resolver – o *Enunciado* – como etapa inicial do processo de análise.

⁶⁸ *Agente*, como diria o Dr. Marcos Shmeil, da PUCPR.

tado como “fundamentação” teórica por algumas metodologias de análise conceitual, como a metodologia Syntropy [Cook94].

De todo modo, sejam os pressupostos implícitos ou explícitos, a origem da conceituação inadequada reside em um condicionamento lingüístico muito profundo. A estrutura das línguas indo-européias, que se reflete na dicotomia sujeito/predicado, condiciona a que o mundo de seus falantes se organize em termos de sujeitos/objetos que fazem coisas – não é, pois, de estranhar que tal modo de conceber a realidade esteja presente nas metodologias de análise e na estrutura dos modelos delas resultantes. Esta visão do mundo encontra-se tão profundamente arraigada no modo de pensar dos modeladores (isto é, dos produtores de conhecimento) que a grande complexidade dos modelos é aceita como absolutamente natural.

Entretanto, existe um outro modo de entender as coisas. A asserção que estamos analisando, “*O cliente quita o débito*”, ao invés de ser interpretada em termos de “*o cliente faz alguma coisa*”, pode ser compreendida como “*alguma coisa acontece*”; nestes novos termos, a afirmação pode ser parafraseada por “*ocorre a quitação do débito do cliente*” – ou, em formato de predicado, *Quitação(cliente, débito)*.

O predicado *Quitação(cliente, débito)* é um predicado lingüístico autêntico – naturalmente interpretado como um acontecimento, um evento ocorrido –, fato que proporciona todo o aparato teórico já elaborado para representação semântica dos predicados, em especial os papéis semânticos dos argumentos. Considerado como um predicado lingüístico, o *cliente* é o *beneficiário* da *Quitação* que tem o *débito* por *meta*; como falta ainda o *agente* da *quitação*, nada nos impede de postular a existência de um agente arbitrário, que pode ser qualquer objeto que preencha as condições de ser um pagador (ou seja, o *agente* é definido por um critério de seleção aplicável a um objeto, sem que seja realmente importante conhecer a classe a que pertence tal objeto – fato que dota o modelo de extraordinária flexibilidade). Temos, assim, o predicado completo: *Quitação(cliente, débito, pagador)*, cuja estrutura encontra-se exibida na Figura 9.

Portanto, a interpretação que propomos como mais adequada para um predicado, sob o ponto de vista da representação de conhecimento, é a de que “*algo acontece*”⁶⁹, e esta interpretação é radicalmente diferente de considerar-se que “*alguém faz algo*”⁷⁰.

Deste modo, nossos predicados serão sempre formalizados por meio de substantivos, que dêem idéia de ocorrência de determinados fenômenos, e não por meio de verbos – *Quitação(cliente, débito, pagador)*, e não *Quitar(cliente, débito, pagador)*. Utilizaremos verbos exclusivamente para denominar *operações*, aquilo que de modo inequívoco e sem exceções constitua a “responsabilidade” de um objeto. Temos, assim, um marcador léxico para diferenciar formalmente o que se entenda ser um evento, do que se defina ser uma operação: substantivo para o primeiro, verbo para o segundo.

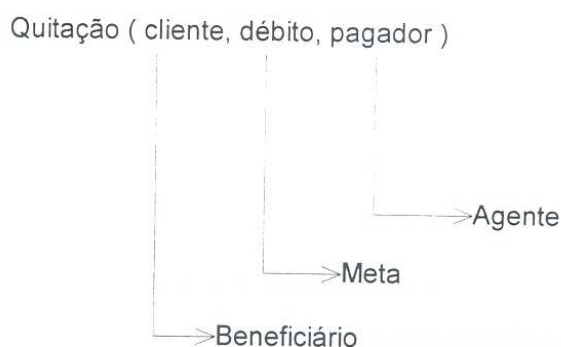


Figura 9: Interpretação de um predicado

⁶⁹ De acordo com a ciência linguística, este modo de interpretar predicado é conhecido como interpretação “ergativa”, embora, os predicados linguísticos das interpretações ergativas também costumem ser expressos como verbos, e não como substantivos, segundo propomos. Cremos que a interpretação ergativa seja psicologicamente mais adequada à formalização dos processos cognitivos humanos, embora não possamos fundamentar a afirmação neste trabalho, considerados seus objetivos.

⁷⁰ O Prof. Gudwin interpreta a diferença entre os pontos de vista a respeito de predicados do seguinte modo: (i) “*algo acontece*” – descrição de uma situação observada; (ii) “*alguém faz algo*” – causas que levam à ocorrência de uma situação, descrição causal. No atual momento, não temos condições de adotar nem de contestar essa análise; apenas observamos que mesmo predicados ergativos podem ser interpretados de maneira causal, como em “*Pedro manda pintar a casa*”, que pode ser ergativamente interpretado como “*Ordem(Pedro, Pintura(casa, alguém))*”, em que o predicado causal *Ordem()* modaliza o predicado de ação *Pintura()*.

Este é o motivo da ressalva anterior, ao dizer que utilizávamos *Quitar* como forma provisória: (1) a forma verbal contém resquícios de uma interpretação inadequada de predicado em termos da ação de objetos isolados; (2) e, ademais, pode induzir a que um predicado, expresso como um verbo, passe a ser considerado como pertencente à funcionalidade de alguma classe, assim introduzindo erros de representação.

É importante observar que *predicado*, no sentido que estamos empregando, de fato, *estabelece uma associação n-ária* entre objetos. Este é um fato primordial: a natureza relacional de um predicado faz dele o elemento natural para representar estruturas – efetivamente, predicado coincide com o conceito de objeto estrutural, no sentido que estabelecemos anteriormente.

Por fim fechamos o cerco tão longamente construído: predicado codifica semântica e predicado codifica estrutura – ou seja, predicado é a forma natural de representação de um modelo (modelo: estrutura dotada de significado). E predicado é uma forma mínima, destituída de todo o aparato empírico das representações basicamente gráficas.

Sabemos que o predicado lingüístico alarga o predicado lógico; nosso objetivo é ampliar o predicado lingüístico, dotando-o da funcionalidade necessária para que ele efetivamente possa constituir-se no elemento fundamental de uma linguagem genérica de representação de conhecimento. O atual trabalho apresenta idéias introdutórias nesta direção.

9 Exemplo de Aplicação de Predicado e Papéis Semânticos

Para ilustrar o potencial de aplicação do conceito de predicado dotado de papéis semânticos, apresentamos um exemplo colhido da análise de um projeto real. Toda a estrutura do projeto baseou-se no conceito de predicado. Apresentamos um predicado aleatoriamente escolhido, com a finalidade de ilustrar o grande potencial da aplicação de papéis semânticos à teoria e à prática da representação de conhecimento. O exemplo mostrado é puramente conceitual, mas sua implementação segue-se diretamente das especificações da modelagem.

O projeto consistiu na especificação de um sistema de administração mercadológica para uma empresa de prestação de serviços de consultoria e assistência técnica. Inicialmente, constatou-se uma grande variedade de funções, atribuições e serviços. Entretanto, a análise, baseada baseada nos conceitos de predicado e papéis semânticos, mostrou que a diversidade de elementos exprime uma estrutura, de certo modo, bastante simples: todas as operações do Sistema Mercadológico foram reduzidas aos dois únicos grupos de operações fundamentais:

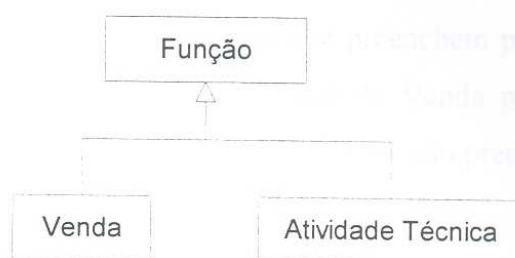
- Atividades_Técnicas: Atendimentos, Comunicação, Processamento e Ocorrência;
- Venda (de qualquer Produto, material ou imaterial).

Qualquer função do SEBRAE, o cliente para o qual foi projetado o sistema, pode reduzir-se a um desses dois tipos.

Tais grupos de operações, na realidade, possuem caráter genérico. As análises mostram que são típicos de uma empresa que dê ênfase à prestação de serviços.

Tecnicamente, a redução das operações do Sistema SEBRAE (no caso, a relação do SEBRAE com seu mercado, em sentido amplo) a uma estrutura elementar proporciona que todo o sistema de informações possa ser gerado a partir de relativamente poucos padrões, reduzindo significativamente a complexidade do software. Cada função do Sistema está localizada em um único lugar, disponível para ser utilizada pelo serviço que dela necessitar, fato que também reduz a complexidade da implementação.

No exemplo, como se pode observar, deixamos de seguir a prática habitual de empregar verbos para nomear predicados: Venda, em lugar de Vender; Atividade_Técnica, em lugar de Executar_Atividade_Técnica. Assim o fazemos pelas razões



apontadas no capítulo anterior.

Figura 10: Aplicação do conceito de predicado

Apresentamos, a seguir, como ilustração, a especificação textual do predicado Venda. Deve-se destacar o uso dos papéis semânticos para definir formalmente as

características dos objetos que podem exercer determinadas funções; assim, a função Vendedor só pode ser preenchida por um objeto da classe Funcionário.

9.1 Venda

- Venda (Objeto_Mercantil (Produto); Vendedor (Funcionário); Comprador (Algum); Local_Abstrato (Algures))

9.1.1 Conceito

Venda, efetuada num local Algures, de um Produto⁷¹ existente, vendido por um Funcionário (Analista_de_Negócio ou Analista_de_Informação) comprado por Algum comprador.

Pode-se dizer que a única função de Produto é poder figurar como Objeto de Venda. Sob o ponto de vista do Sistema Mercadológico, Produto é tudo aquilo que pode ser vendido pelo SEBRAE.

9.1.2 Exemplo de Aplicação

O predicado genérico Venda pode ser especializado para representar um tipo particular de venda, como mostrado no próximo exemplo; é importante observar a compatibilidade de características entre os objetos que preenchem papéis semânticos: papéis semânticos de uma versão mais especializada de Venda precisam ser compatíveis com aqueles pertencentes a uma versão mais genérica do predicado.

⁷¹ Produto é definido como um modo de uso de determinados objetos ou serviços técnicos – especificamente, serem comercializados. Ou seja, determinados elementos podem ser considerados “em sua condição de” serem comercializados, de serem “tratados como” Produtos. Quaisquer atividades (técnicas) ou coisas podem ser consideradas Produto, desde que diretamente comercializadas ou intermediadas pelo SEBRAE, e que cumpram determinadas condições. Não apenas *objetos básicos* são **Produtos** (isto é, podem ser comercializados), mas também *eventos e serviços*.

- Venda (Objeto_Mercantil (Produto); Vendedor (Analista_de_Negócio); Comprador ("Grupo Buaiz"); Local_Abstrato (SEBRAE))

9.1.3 Classe

A classe abaixo, em linguagem UML, formaliza o predicado genérico Venda, mostrado anteriormente de modo textual.

Embora em formato “próximo a uma representação gráfica”, o predicado a seguir é puramente conceitual; a característica bidimensional do formalismo empregado exhibe os recursos disponíveis para representação do predicado como objeto pertencente a uma estrutura: atributos, operações e regras estruturais *do predicado* são representados de modo natural.

Venda	
[Objeto_Mercantil; Vendedor; Comprador; Local_Abstrato]	
{Função_SEBRAE}	
•Data_de_Realização:	Data
•Data_de_Pagamento_Prevista:	Data
•Data_de_Pagamento_Efetiva:	Data
•Criar_Venda ()	
•Invariante //	não definido

Parte V – Ética

Introdução da Regra em uma Estrutura

em toda sabedoria está a prática da Lei.

Eclo, 19, 21

A estrutura constitui uma hierarquia que apenas conhece sua própria ordem, e para a qual só existe um acesso possível: para reconhecê-la segundo seu princípio inerente, cumpre proceder de maneira ao mesmo tempo dedutiva e irreversível, descendo gradativamente dos fatos mais abstratos (gerais, simples) para os fatos cada vez mais concretos (particulares, complexos).

L. Hjelmslev, Ensaio Lingüísticos

Estado

Naquela noite

Alarguei as angústias dos vales

E pisei o húmus da alma.

Só então o Sol nasceu

10 Noção Tradicional de Estado

Parte significativa da conceituação atual a respeito de estados⁷², de modo acrítico, gira em torno da idéia de que os *estados constituem características visíveis pertencentes aos objetos*. Faz parte da cultura da orientação a objetos denominar coletivamente os valores dos atributos de um objeto por “o estado do objeto” – uma breve passada em manuais da área basta para comprovar a afirmação. Mesmo autores que pretendem desenvolver uma fundamentação formal (“*State is a collection of associations an object has with other objects and object types*”[Odell95], pag. 112) apenas repetem o conceito empírico, embora de modo mais sofisticado. A idéia, na realidade, deriva da concepção atribuída a Aristóteles de atributo como característica implicitamente ligada a um objeto, e, também, de que estados, de algum modo, são atributos. Portanto, uma idéia bastante respeitável.

⁷² Uma das propostas deste trabalho é que eventos são, “na realidade”, um tipo particular de estado. Assim, embora este capítulo mencione somente estados, suas considerações aplicam-se igualmente a eventos.

James Odell [Odell95] [Odell96] e Steve Cook [Cook94] sintetizam de modo bastante feliz, sob diferentes pontos de vista, o que pode ser chamado de teoria-padrão da orientação a objetos – assim, serão extensivamente citados neste trabalho. Referências de mais ampla circulação, de interesse histórico e prático a respeito da teoria-padrão, são [Rumb94] [Booch94], [Coad95] e [Jacob92]. Considerações de ordem metodológica, com enfoque formal sobre a teoria-padrão podem ser encontradas em [Odell96]. [Eco76] desenvolve uma abordagem à representação de conhecimento sob o ponto de vista semiótico (teoria da significação), cujo conteúdo é importante para profissionais de sistemas de informação.

Um avanço com relação a essa tradição encontra-se em [UP99], pag. 431 – “(State is) a condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.” – em que estado está desvinculado da noção de valor de atributo. Entretanto, tal como no parágrafo anterior, estado é entendido sob o ponto de vista de um objeto particular – e este é o ponto relevante.

Esse princípio, de primado do objeto na constituição de um modelo, está presente também na análise da funcionalidade do objeto. No caso, o fundamento teórico tem sido influenciado pela abordagem behaviorista, centrada na análise do *comportamento observável* de um objeto que recebe estímulos e produz respostas [Cook94]. Tal enfoque é até mesmo considerado como o cerne da análise de um modelo, e baseia-se em uma visão demasiado naturalista de como descrever a “realidade”.

(Deixaremos de lado o fato de que os estados não são efetivamente representados, mas são reconhecidos, pela análise dos valores de atributos, por um observador externo ao modelo.)

Assim, temos uma premissa infiltrada, de modo acrítico, no modo como a modelagem é usualmente entendida: os objetos *possuem* seus estados. Esta premissa está presente na aplicação das máquinas de estados finitos – versão mais difundida, *statecharts* de David Harel [Harel87] – à análise dos estados. Um fato, porém, precisa ser ressaltado: este não é um pressuposto das máquinas de estado, mas do modo como

elas costumam ser empregadas na modelagem orientada a objetos – “*Statecharts capture the following information: a list of the events of interest to an object type; ...*” [Cook94], pag. 92; “... *there actually exists simultaneously one instance of a state-chart for each object...*”, pag. 111.

Mas, será tal modo de entender estado, sob o ponto de vista de objetos particulares, adequado à representação de um domínio conceitual?

Para responder, analisemos uma situação prática, a título de exemplo: uma nota fiscal é considerada vencida não apenas quando sua data de pagamento está atrasada, mas também se inexistente pagamento pendente, ou contestação judicial, ou o serviço está inconcluso, ou o cliente não é “especial” etc.. Ou seja, uma nota é considerada vencida dependendo do “estado” de um conjunto de objetos. E o (estado de) *vencimento* de uma nota pode ter atributos (como *responsável pela constatação*) e funcionalidade (como *emissão de aviso*), que não são *características pertencentes à* nota.

Esta caracterização abstrata de um estado – *propriedade ou funcionalidade de um “conjunto de objetos”* – está em descompasso com a visão tradicional de objeto portador de seus estados. No exemplo acima, nota fiscal não é portadora de seus estados. E qualquer representação dos estados de nota fiscal, por tomar como foco uma nota fiscal particular, estaria envolta em notável complexidade e dificuldade de implementação.

Chegamos a uma conclusão: *o conceito vigente de estado é insuficiente para fazer face a situações reais de representação de um domínio*. O problema só não se torna explícito porque os estados não são *diretamente* representados ou processados, uma vez que residem em máquinas de estados externas, ou estão embutidos em programas que processam os modelos. Como os estados não são visíveis, as dificuldades são atribuídas a outras causas – a mais freqüente ligada à percepção de que os modelos são “realmente muito complicados”.

De todo modo, consideramos que a teoria vigente de sistemas de informação trata o conceito de estado de modo puramente empírico e informal, e, principalmente, também inadequado.

11 Conceito Formal de Estado

A formalização dos conceitos de estado e de evento – de modo implementável em linguagens de programação – constitui a principal contribuição deste trabalho. Faremos, a seguir, uma análise detalhada do conceito de estado. Primeiramente, desenvolveremos uma discussão intuitiva do conceito, necessária não apenas para a compreensão teórica, mas também para que a idéia possa ser utilizada na modelagem prática. Depois, apresentaremos a definição formal de estado, implementada em linguagem Eiffel e em Object Pascal. Na continuação, serão mostrados exemplos de aplicação. A discussão a respeito de eventos será bastante sumária, pois a amplitude do tema exige atenção especial – e é objeto de pesquisa em andamento.

11.1 Base Intuitiva

Do ponto-de-vista intuitivo, *estado significa um conceito, ou uma condição, aplicável a um objeto.*

A Figura 11 ilustra um critério aplicado a determinado ser humano, segundo o qual ele é, ou não, considerado obeso. Suponhamos que o critério seja definido por uma fórmula que combine *peso* e *altura*. Caso a combinação de peso e altura do João esteja acima de um determinado valor, ele será considerado obeso, caso contrário, será considerado não obeso (talvez magro).

É importante observar que não estamos definindo uma propriedade do João, mas um conceito em que ele eventualmente se enquadre, pois, se a fórmula for alterada, João poderá ter sua condição mudada para não obeso, mesmo que seu peso e altura não se alterem. Ou seja, por si, os valores dos atributos do João nada informam sobre sua condição de obesidade. Muitas pessoas podem até não ter idéia do que seja obesidade, mesmo tendo peso e altura.

A Figura 11 mostra um *estado* em sua forma mais elementar: uma condição aplicada a um único objeto. O estado também pode ser representado como um predicado lógico, como *Obeso (João)*, que é uma forma análoga à do diagrama – embora predicado lógico não seja noção equivalente ao conceito de estado que estamos propondo, assunto que será tema de outro trabalho.



Figura 11: Estado - Representação gráfica

Assim, temos uma representação gráfica para estados, conforme o exemplo da figura acima: o retângulo maior representa um objeto que é o estado, e o retângulo menor representa um objeto que é regido pelo estado. Por comodidade, diremos que o objeto regido pelo estado é o *objeto do estado*.

Verifica-se, deste modo, que o formalismo gráfico da orientação a objetos pode ser utilizado para representar estados, por meio das denominadas “classes paramé-

tricas”, como na figura. O exemplo, e o trabalho em geral, utiliza a linguagem gráfica UML – Unified Modeling Language, para representações gráficas.

A forma gráfica mostra, entretanto, uma perspectiva mais ampla: toda a tecnologia de objetos pode ser utilizada para o tratamento de estados – uma vez que, como é ilustrado pela Figura 11, **um estado é um objeto**.

11.2 Observações sobre o Conceito

Alguns fatos necessitam ser destacados a respeito do conceito de estado, antes de passar a definições mais formais:

- Estado *Obeso* (*João*) aplica-se a João mesmo que João não seja obeso; caso João não seja obeso, diz-se que o estado “*não está ativo*”. Em geral: um estado encontra-se *ativo* se a regra que o caracteriza é satisfeita pelo objeto; e *não ativo* caso contrário. Portanto, um estado pode ocorrer, mesmo não estando ativo.
- Consideremos o objeto que é o estado *Obeso* de João. Este estado pode ter propriedades, e em geral as tem – por exemplo, a data em que João tornou-se obeso (isto é, a data em que o estado tornou-se ativo). Ou seja, um estado possui propriedades que não são propriedades pertencentes a seu objeto, mas são propriedades (mais ou menos transitórias) *aplicadas* a seu objeto.

11.3 Representação Gráfica

A Figura 12 exhibe a representação gráfica completa de um estado. Tanto o estado quanto seu objeto descendem do objeto mais geral existente, ao qual denominamos *Coisa*. Este objeto genérico deve ser entendido como puramente conceitual, mas co-

mumente recebe implementações nas linguagens da orientação a objetos; Eiffel, por exemplo, denomina-o ANY, Object Pascal chama-o de TObject, e Java referencia-o por Object. O nome dado é pouco relevante, o aspecto fundamental é o fato de um estado ser um objeto de pleno direito, como qualquer outro.

Como já foi mencionado, *atributos* e *operações* de um estado devem ser entendidos como características *aplicadas ao seu objeto*, mas *não pertencentes ao objeto*. A *condição_de_estado* especifica em que circunstâncias o estado encontra-se *ativo*.

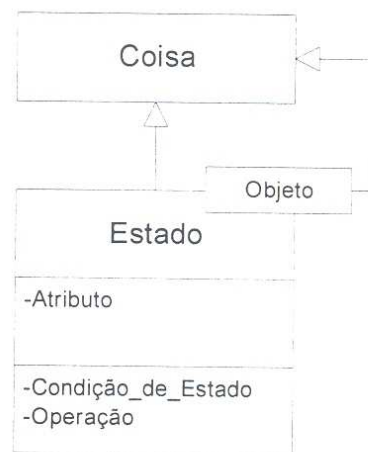


Figura 12: Estado - Representação gráfica completa

O aspecto mais importante a ser ressaltado é que o *objeto do estado* não se restringe, em geral, a um objeto simples, mas a um conjunto ordenado de objetos. Este é o caso mais geral de um estado: definir propriedades de um conjunto de objetos.

Toda a análise precedente mostra que, ao contrário da teoria-padrão, em que os objetos *possuem* estados, temos a situação inversa: nossos estados é que, efetivamente, são responsáveis por seus objetos. Segundo o conceito, um objeto pode estar sendo regido por um estado sem mesmo o perceber (e esta é a situação geral). Ademais, o conceito mostra que um determinado objeto pode estar submetido a vários estados ao mesmo tempo, sem que o fato produza maiores complicações. Outro fato relevante é que, por exemplo, um estado não envolve necessariamente objetos que mantenham, entre si, relações “visíveis”, no sentido de vínculos observáveis em um

diagrama de classes; em outros termos, um estado pode envolver objetos sem qualquer vínculo observável “natural” no domínio, um estado define uma reunião puramente convencional de objetos.

11.4 Aspectos Semânticos do Conceito de Estado

Abordaremos agora, de modo particular, os aspectos semânticos do conceito de estado, segundo a perspectiva deste trabalho. É preciso, porém, observar que os vários temas aqui desenvolvidos têm por objetivo justamente a proposta de uma tal semântica. Assim, o presente capítulo não constitui mais do que um resumo das idéias que se encontram ao longo de todo o texto (desculpe-nos o leitor, então, algumas repetições).

Na apresentação que se segue, para simplificar, designaremos a semântica de estado que estamos desenvolvendo por “o conceito de estado”, ou, resumidamente, “o conceito”⁷³.

11.4.1 Semântica Fundamental de Estado

Estado é uma condição aplicada a um objeto.

Ou seja: estado **não** é uma característica intrinsecamente pertencente a um objeto, ou nele diretamente observada – como, por exemplo, o fato de o objeto “ser verde”, ou “ser áspero”, ou “medir 3 m de largura”, ou “possuir 4 pernas”, ou “chutar bola”. Tais características, para sermos consistentes, são os (valores dos) **atributos** do objeto, e o conjunto desses atributos pertencentes ao objeto constituem (ou pertencem a) uma **estrutura**⁷⁴. Portanto, aquilo a que os livros de orientação a objetos cos-

⁷³ Este modo de expressão não se deve a qualquer veleidade, mas apenas à necessidade de simplificação da escrita, para evitar repetições.

⁷⁴ Aqui cabe um comentário: (i) os *valores* dos atributos de um objeto particular – *instância* – constituem uma estrutura; (ii) mas, também, uma *descrição* das características de um objeto – *classe* – compõe uma estrutura. Assim, sob o ponto de vista da representação, para cada objeto estão em jogo

tumam chamar de “o estado do objeto” (o conjunto dos valores dos atributos), mais apropriadamente, deve denominar-se “a estrutura do objeto”.

Para interpretar adequadamente o conceito de “estado como *condição*”, desvinculando-o de qualquer noção de “característica intínseca” ou de “propriedade observável”, analisemos alguns exemplos.

Poucos anos atrás, as autoridades médicas dos EEUU alteraram as condições para que uma pessoa seja classificada como diabética – em nossos termos, classificada como pertencente ao estado *Diabético* (*pessoa*). Deste modo, pessoas anteriormente consideradas “saudáveis” (sob o aspecto da glicemia) passaram a ser enquadradas como diabéticas. Ou seja: legalmente, a condição glicêmica de uma pessoa alterou-se, sem que, necessariamente, tenha variado qualquer medida relacionada a suas características físicas – o *estado* de saúde de uma pessoa pode ter-se alterado, sem que suas condições físicas necessariamente tenham-se modificado.

O exemplo ilustra que uma “condição” (o estado) não está diretamente ligada a características pertencentes a um objeto, ainda que, eventualmente, tenham algo “a ver” com essas características.

Entretanto, a questão envolve outros fatores. Todos sabemos que um objeto pode ser considerado “morno” – estado *Morno* (*objeto*) – para determinadas pessoas, mas não por outras. Ou seja, o estado de um objeto “tem a ver” não somente com características do objeto, mas com características ligadas à “relação” entre observador e objeto.

Outros elementos podem ser acrescentados. Usualmente, não há questionamento a respeito da cor de um objeto – por exemplo, um objeto “ser vermelho” não

duas estruturas: (i) *estrutura extensional*, que é a “composição” do objeto, e (ii) *estrutura intensional*, que descreve a estrutura extensional, isto é, que descreve o objeto particular. Embora exista um paralelismo entre as duas estruturas, elas correspondem a “mundos” diferentes, de modo algum havendo uma ligação “natural” entre *descrição* e *objeto descrito*. Todo o tema merece uma discussão mais pormenorizada, que, no entanto, extrapola os objetivos do trabalho. De todo modo, não se pode confundir *objeto* com *descrição de objeto*.

costuma suscitar maiores indagações. Mas, será que, em ambientes escuros, objetos “vermelhos” ainda serão considerados como possuidores da cor vermelha? É fato conhecido que a cor de um objeto varia de acordo com o grau de iluminação do ambiente. Assim, existem motivos para considerar-se a cor como algo não intrinsecamente pertencente a um objeto, mas como um estado a ele ligado – no exemplo, *Vermelho (objeto)*. Ou seja, o estado *Vermelho (objeto)* “tem a ver” com características do objeto, mas também com condições ambientais. Entretanto, como existem pessoas daltônicas, podemos concluir que a caracterização do estado *Vermelho (objeto)* “tem a ver” com características do objeto, do ambiente e do observador.

Os exemplos mostram que um estado não está ligado exclusivamente à situação de objetos particulares, mas a um “estado de coisas”, cujos contornos, muitas vezes, tornam-se difíceis de definir.

Vejamos, por exemplo, o estado de uma “pessoa casada” – *Casado (pessoa)*. Evidentemente, em nossa cultura⁷⁵, a caracterização do estado não se liga (nem mesmo indiretamente) a qualquer característica física pertencente ao “objeto” pessoa em questão, como também independe de particularidades de observadores específicos e do ambiente físico. É claro que há a questão de saber se existem erros de cartório, ou se em outro país a pessoa ainda seria considerada casada, etc..

Portanto, os exemplos mostram que a formulação de um conceito genérico de estado vai além da enumeração de características previamente definidas: (i) o único pressuposto é que existe um objeto em foco, e que (ii) esse objeto satisfaz, ou não, uma condição lógica⁷⁶ a ele aplicável (ou não).

Os exemplos tiveram por finalidade básica, além de ilustrar o conceito de estado, enfatizar a independência implícita que existe entre estados e características de objetos de quaisquer tipos.

⁷⁵ Brasileira, ou “ocidental”.

⁷⁶ O modo como o objeto satisfaz à condição lógica pode ser generalizado para além dos limites de uma condição booleana; nossa proposta para fazê-lo depende de análises ainda em andamento.

12 Definição Formal de Estado

Passaremos agora à formalização rigorosa do conceito de estado, por meio da especificação de uma classe em linguagem Eiffel.

Embora especificadas de modo similar ao das linguagens de programação, as definições formais de estados e eventos que propomos mostram-se efetivamente como puramente formais (isto é, abstratas e genéricas) e destituídas de aspectos de implementação. Os únicos constructos empregados são aqueles pertencentes à noção geral de estrutura: descrição (classe), associação (atributo, função), e regra de transformação (operação). Nada é colocado que especifique um conceito pertencente a uma área de aplicação ou que seja recurso proveniente de uma tecnologia particular. Qualquer acréscimo de conteúdo é feito através de classes herdeiras – por meio do mecanismo de herança – que agregam os detalhes ligados a domínios conceituais e a ambiente tecnológico.

A definição de estado deste capítulo demonstra que conceitos abstratos e de cunho filosófico podem ser diretamente implementados em sistemas de informação. Não existe a dicotomia *conceitualização/implementação*, ou *teoria/prática*, desde que seja utilizada a aparelhagem teórica adequada.

A linguagem Eiffel mostra-se especialmente dotada para produção de especificações formais, sobretudo graças a um mecanismo (*design by contract* [Meyer97]) que facilita a especificação de classes em graus decrescentes de abstração. Este é o motivo de nossas definições terem sido desenvolvidas inicialmente em Eiffel⁷⁷. Entretanto, as especificações que propomos podem ser implementadas em qualquer linguagem da orientação a objetos. Muitas das implementações aqui exibidas foram feitas em Object Pascal, via Delphi (embora tal linguagem não ofereça o recurso das classes paramétricas, úteis para definir parâmetros formais, mas de modo algum necessárias às implementações).

Em Eiffel, estado foi definido por meio da classe ESTADO [G], cuja especificação completa encontra-se a seguir. De acordo com a notação, um determinado estado – descrito por ESTADO – contém uma referência – descrita por G – a um objeto particular. Ou seja, ESTADO é uma classe “container” de objetos – exatamente os objetos aos quais o estado se aplica.

O ponto fundamental da classe ESTADO [G] é a definição da *lógica de estado*⁷⁸, implementada pela rotina *atualiza_estado*, que controla todo o comportamento do estado, a partir da *condicao_de_estado*, condição booleana que define em que circunstâncias o estado encontra-se ativo. Isto é: a rotina *atualiza_estado* define o comportamento de um estado no momento em que ele é acionado, dependendo das condições (definidas em *condicao_de_estado*) satisfeitas pelo objeto quando:

⁷⁷ Devido à sua estratégia de compilação, atualmente Eiffel é a linguagem mais apropriada para implementação das propostas deste trabalho: Eiffel gera código executável apenas para as características de uma classe que sejam efetivamente referenciadas em uma compilação – assim, as classes podem ser definidas de modo genérico e completo, sem que o fato implique a geração de código desnecessário a execuções específicas.

⁷⁸ Os termos em itálico que figuram no parágrafo referenciam operações e funções pertencentes à classe ESTADO [G], especificada neste capítulo; as referências não foram acrescentadas ao parágrafo para não prejudicar a naturalidade da leitura.

- O objeto acaba de entrar no estado;
- O objeto já se encontra no estado;
- O objeto está para sair do estado;
- E até mesmo o estado encontra-se inativo para o objeto.

A classe ESTADO [G] foi exaustivamente testada em diversas aplicações (por exemplo, em todas as citadas no atual relato), sem terem sido encontrados erros em seu funcionamento. Por se tratar de uma classe absolutamente genérica (na realidade, trata-se de um *frame* definidor de uma lógica), as rotinas específicas precisam ser redefinidas pelas classes herdeiras. Não é demais observar que o objeto genérico G pode ser absolutamente qualquer estrutura, simples ou complexa, o que definitivamente afasta a idéia de que o conceito de estado está associado a objetos individuais (digo, simples).

Um fator importante, a ser destacado, é que a posição de parâmetro formal G (*objeto do estado*) é ocupada por uma *referência* a um objeto, e não diretamente pelo próprio objeto.

O código a seguir contém a especificação completa de um estado genérico, amplamente comentada. Porém, o funcionamento da classe poderá não ficar evidente a partir apenas dessa especificação, que envolve muitos detalhes. Por este motivo, após a apresentação da classe, faremos uma análise pormenorizada de suas principais partes.

12.1 Estado: Formalização em Eiffel

A classe ESTADO [G] encontra-se extensamente comentada, motivo que nos leva a apresentá-la sem notas explicativas, além das já apresentadas ao longo do trabalho.

```
deferred class ESTADO [G]
    -- padrão genérico de estado
    -- um estado define propriedades de um conjunto de objetos,
    -- conjunto referencido pelo objeto G, que é composto
    -- ou simples
feature {ANY}
    ---- objeto regido pelo estado
    objeto: G
    ---- fim objeto regido pelo estado
feature {ANY}
    ---- características usadas pela lógica de estado
    condicao_de_estado: BOOLEAN is
        -- expressão booleana que caracteriza o estado,
        -- o estado ocorre se a condição é verdadeira
        deferred
    end -- condicao_de_estado
    estado_ativo: BOOLEAN
        -- se verdadeiro, estado está ativo; não ativo, se falso
    estado_anterior: BOOLEAN
        -- valor de estado_ativo quando do último acionamento do estado
        -- salva valor de estado_ativo, antes de este ser atualizado
    acao_de_entrada is
        -- ações a executar apenas a cada nova entrada no estado,
        -- mas não quando o estado já se encontra ativo
        deferred
    end -- acao_de_entrada
    acao_de_saida is
        -- ações a serem executadas a cada saída do estado
        deferred
```

```

end -- acao_de_saida
regra_de_estado_ativo is
    -- ações a serem executadas sempre que o estado for acionado
    -- e estiver ativo
    -- colocação da situação nas regras do estado (que são as
    -- definidas por esta feature)
    deferred
end -- regra_de_estado_ativo
---- fim características usadas pela lógica de estado
regra_de_estado_inativo is
    -- ações a serem executadas sempre que o estado for acionado
    -- e estiver inativo
    deferred
end -- regra_de_estado_inativo
feature {ANY}
---- features acionadas pela lógica de estado
texto_estado_ativo: DESCRICAO is
    -- texto padrão referente a estado ativo
    do
        Result.set_descricao("--Estado ativo--")
    end -- texto_estado_ativo
texto_estado_n_ativo: DESCRICAO is
    -- texto padrão referente a estado não ativo
    do
        Result.set_descricao("--Estado nao ativo--")
    end -- texto_estado_n_ativo
---- fim features acionadas pela lógica de estado
feature
---- features de uso externo
frozen atualiza_estado is
    -- define a lógica de estado, não realiza outro processamento
    -- não pode ser redefinida por classes herdeiras
    -- controla:
    --     (a) condições de ocorrência do estado
    --     (b) execução das ações de ocorrência do estado
    --     (c) execução das ações de entrada no estado
    --     (d) execução das ações de saída do estado

```



```

--      (e) execução das ações de não ocorrência do estado
--      (f) situação do estado no último acionamento
do
    estado_anterior := estado_ativo
    if condicao_de_estado    -- estado se configura
    then if not estado_ativo -- então é entrada no estado
        then acao_de_entrada
            estado_ativo := True
        end
        regra_de_estado_ativo -- aplica regras do estado
    else if estado_ativo     -- então é saída do estado
        then acao_de_saida
            estado_ativo := False
        end
        regra_de_estado_inativo
    end
end -- atualiza_estado
print_estado is
    -- impressão padrão do estado
    -- ilustra lógica de uma rotina de estado
    -- acionada externamente
do
    if estado_ativo
    then print(texto_estado_ativo.out)
    else print (texto_estado_n_ativo.out)
    end
end -- print_estado
make_estado(n_objeto: G) is
    -- inicializa ESTADO que rege
    -- (isto é, define regras de uso para)
    -- um objeto G de classe Genérica
--    require
--        objeto /= Void
do
    objeto := n_objeto
end -- make_estado
end - ESTADO

```

12.2 Formalização em Object Pascal

A classe abaixo exibe uma implementação operacional de Estado, com procedimentos de recuperação, de modo apropriado ao uso em sistemas de informação reais.

```
type TEstado = class(TObject)
// padrão genérico de estado
protected
    FObjeto: TObject; //objeto regido pelo estado
    FTexto: TTexto;
    FEstado_Ativo: Boolean;
    FEstado_Anterior: Boolean;
    procedure Restaura_Valores; virtual; abstract;
    procedure Salva_Valores; virtual; abstract;
    function Invariante: Boolean;
    function Criar_Precond(ParObjeto: TObject): Boolean;
    function Criar_Poscond(ParObjeto: TObject): Boolean;
    procedure Aciona_Prerreq; virtual;
    function Aciona_Precond: Boolean; //virtual;
    function Condicao_de_Estado: Boolean; virtual;
    function Acao_de_Entrada_Poscond: Boolean; virtual;
    procedure Acao_de_Entrada; virtual;
    function Regra_de_Estado_Ativo_Precond: Boolean; virtual;
    function Regra_de_Estado_Ativo_Poscond: Boolean; virtual;
    procedure Regra_de_Estado_Ativo; virtual;
    procedure Acao_de_Saida; // virtual;
    procedure Regra_de_Estado_Inativo; //virtual;
    procedure Mensagem_Estado_Ativo;
    procedure Mensagem_Estado_Inativo;
    function Aciona_Poscond: Boolean; // virtual
    procedure Aciona_Posreq; // virtual
public
    property Texto: TTexto read FTexto write FTexto;
```

```

property Estado_Ativo: Boolean read FEstado_Ativo;
property Objeto: TObject read FObjeto;
procedure Aciona;
constructor Criar(ParObjeto: TObject); virtual;
end; // Testado

procedure TEstado.Aciona;
// Lógica estatal; não pode ser redefinida por herdeiras
begin
// Início tratamento de Pré-requisito
try // Início try ... finally ...;
    // prepara procedimento de recuperação
try//try ... except ...
    Aciona_Prerreq;
except
    on Exception do
        ShowMessage('Exceção: pré-requisito de estado');
    end; //fim - try ... except ...
finally
    // procedimento de recuperação
    // em caso de falha, se necessário
end; // Fim try ... finally
// Fim tratamento de Pré-requisito
// Início tratamento de Pré-condição
try//try ... except ...
    if not Aciona_Precond then raise
        EException_P.Create
            ('Exceção: pré-condição de estado');
except
    on Excecao: EException_P do
        begin
            Excecao.Procedimento;
        end;
    end; //fim - try ... except ...
// Fim tratamento de Pré-condição

```



```

// Início do corpo da operação
try // try ... finally ...
    try // try ... except ...; corpo da operação
        // Início lógica de Estado
        FEstado_Anterior := FEstado_Ativo;
        if Condicao_de_Estado
        then begin;
            if not FEstado_Ativo
            then begin Acao_de_Entrada; FEstado_Ativo := True;
end;
                Regra_de_Estado_Ativo;
            end
        else begin
            if FEstado_Ativo
            then begin Acao_de_Saida; FEstado_Ativo := False; end;
                Regra_de_Estado_Inativo;
            end;
        // Fim lógica de Estado
    except
        //Exceções do corpo da operação
        on Exception do
            ShowMessage('Exceção: lógica de estado');
        end; // Fim try ... except
    finally
        // procedimento de recuperação, se necessário
    end; // try ... finally
// Fim do corpo da operação
// Início da pós-condição
try // try ... except ...
    if not Aciona_Poscond then raise
        EException_P.Create
        ('Exceção: pós-condição de estado');
except
    on Excecao: EException_P do
        begin

```

```

        Excecao.Procedimento
    end;
end; // try ... except
// Fim da pós-condição
// Início dos pós-requisitos
try // try ... finally
    try // try ... except
        Aciona_Posreq;
    except
        on Exception do
            ShowMessage('Exceção: pós-requisito de estado');
        end; // Fim try ... except
    finally
        // procedimento de recuperação, se necessário
    end; // try ... finally
// Fim dos pós-requisitos
end; // Estado.Aciona;

```

12.3 Exemplo de Aplicação do Conceito de Estado

O diagrama abaixo ilustra um exemplo de aplicação de estado em um sistema de recursos humanos. Não foi acrescentado código fonte, uma vez que o objetivo é apenas mostrar a aplicabilidade da idéia (grande quantidade de código fonte foi acrescentada a outras partes do trabalho).

Os sistemas de recursos humanos incluem-se entre os de modelagem mais complexa, devido à grande variedade de tipos de contrato, regimes de trabalho, objetos de trabalho etc.. Em grande parte, a complexidade da representação resulta da falta de ferramentas conceituais de análise e representação, como as que propomos. O exemplo ilustra uma série de funções (Funcionário, Função Mercadológica, Analista de Negócio, entre outras) que recebem uma organização natural sob a forma de estados.

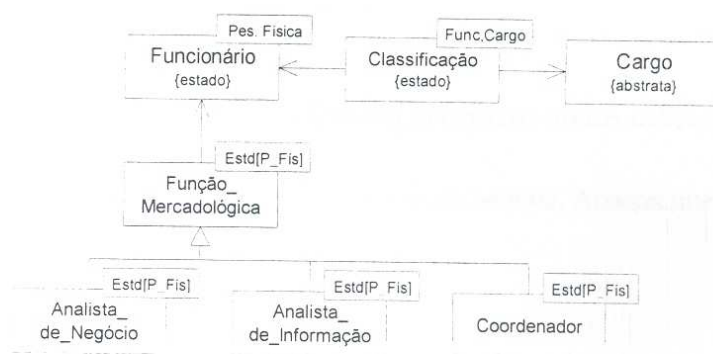


Figura 13: Aplicação do conceito de estado

Evento

Muitos sentirão alegria

Porque deles é o primeiro milagre

E darão o óbulo do fariseu com ares humildes

Muitos não compreenderão

Porque suas inteligências vão somente até os processos

E já existem nos processos tantas dificuldades...

Vinicius de Moraes, Acontecimento

13 Noção Tradicional de Evento

Na teoria tradicional, os eventos são definidos como “transição de estados”: “*An event is a noteworthy change in state*”, [Odell95], pag. 126; “O efeito de um evento detectado é causar alterações de estados em um ou mais objetos.”, [Cook94] (pag. 137). Segundo tal teoria, um evento ocorre instantaneamente (como mudança de estados) e só estará no modelo por suas manifestações, ou seja, pelas alterações que produz nos objetos. Deste modo, interpretado como ocorrência instantânea é negado ao evento qualquer direito de figurar em um modelo: um evento só possui “o antes” e “o depois”, o evento propriamente dito não existe.

Portanto, do ponto de vista conceitual, a situação dos eventos pode ser considerada mais grave do que a dos estados: estados não são definidos, e eventos são definidos a partir de estados – ou seja, a conceituação de eventos é definitivamente posta de lado⁷⁹.

⁷⁹ Estamos nos referindo aos eventos “instantâneos” – eventos conceituados como “transições de estados” –, segundo a teoria que comumente figura nos textos da orientação a objetos.

A teoria do evento “instantâneo” baseia-se, por um lado, em um pressuposto ontológico: a idéia de que objetos, de algum modo, devam ser “entes”. Assim, não há lugar para eventos em um modelo de “coisas”, ainda que de coisas ideais; só há lugar, mesmo que de forma indireta, para “estados de coisas”. O evento instantâneo, sem duração, apenas detectável por seus efeitos, fica fora do modelo. Por outro lado, a caracterização de evento como algo “que não existe” evidencia uma grande dificuldade conceitual: quem quer que se proponha a definir evento, formalmente, terá de lidar com o conceito de estado – como estado não está definido⁸⁰, evento não pode ser definido, e, assim, é excluído das considerações.

É interessante observar que a concepção de evento ocorrendo instantaneamente é contrária ao conceito intuitivo, uma vez que os eventos, no “mundo real”, possuem duração; comumente podemos dizer que um evento “está acontecendo” – uma sirene “está soando”; um bolo “está sendo feito” – sem que se possa falar em mudança “instantânea” de estado.

⁸⁰ Até o momento, não encontramos uma definição conceitualmente adequada de estado nas fontes teóricas e práticas da orientação a objetos: basicamente os estados são tratados como um conceito do senso comum, e este conceito de senso comum contém, como pressuposto implícito, a noção aristotélica de atributo como algo “pertencente” ao objeto. É esse conceito basicamente empírico (não definido) de estado que cria dificuldades à definição de evento (daí a nossa hipótese de o evento “instantâneo” corresponder a uma espécie de válvula de escape, pela qual o evento é apresentado e, logo após, excluído das considerações). Nossa abordagem propõe uma conceituação – e uma definição formal – para estado, e, a partir desta, uma conceituação e uma formalização de evento. Evidentemente, pode-se partir da conceituação de evento, para daí, derivar o conceito de estado.

14 Definição Formal de Evento

Neste trabalho, é dado um tratamento bastante preliminar a evento – todo o tema, de grande riqueza, faz parte de pesquisa que estamos desenvolvendo. De todo modo, apresentamos a especificação formal de evento em linguagem Eiffel.

A característica relevante da classe EVENTO [G] é que o objeto do evento é uma lista de estados; idéia cuja fundamentação será desenvolvida posteriormente, como já observado.

A definição a seguir é uma especificação formal, segundo o formalismo da orientação a objetos, mas também é uma definição operacional, ou seja, uma implementação.


```

deferred class EVENTO [G -> LISTA[ESTADO[ANY]]]
  -- padrão genérico de evento
  -- um evento é um estado que regulamenta outros estados,
  -- que são os estados pertencentes à lista
  inherit
    ESTADO [G]
    rename
      objeto as lista_de_estados
    end
end -- class EVENTO

```

14.1 Especificação Formal de Evento com Implementações

A classe a seguir, em Object Pascal, implementa a mesma definição formal apresentada em linguagem Eiffel, mas acrescenta os procedimentos de recuperação necessários a uma implementação real. Embora contendo mecanismos de recuperação necessários a implementações, a especificação constante deste capítulo preserva as características de abstração e generalidade de aplicação, sendo assim uma especificação puramente formal, uma vez que não incorpora qualquer definição ligada a domínios específicos ou a soluções técnicas.

```

type TEvento = class (TEstado)
  protected //protected
    function Invariante: Boolean;
    function Criar_Precond(ParObjeto: TLista_de_Estados): Boolean;
  public
    constructor Criar(ParObjeto: TLista_de_Estados);
end; // TEvento

function TEvento.Invariante: Boolean;

```

```

begin
    Result := inherited Invariante;
    if Result then
        begin
            try // try ... except ...
                if not (FObjeto is TLista_de_Estados)
                    then raise EException_P.Create
                        ('Violado Invariante de Evento: Objeto precisa ser
TLista_de_Estados');
                if not (FObjeto as TLista_de_Estados).Invariante then
                    raise EException_P.Create
                        ('Violado Invariante de Evento: um dos parâmetros não é
estado');
            except
                on Excecao_3: EException_P do
                    begin
                        Excecao_3.Procedimento;
                        Result := False;
                    end;
            end; // fim - try ... except ...
        end; // begin
    end; //TEvento.Invariante

```

15 Conseqüências das Noções Tradicionais

A interpretação tradicional, que acabamos de discutir, levou a que estados e eventos sejam considerados como tendo uma natureza essencialmente distinta da dos objetos que diretamente compõem um domínio conceitual. Esta separação de noções (estados e eventos de um lado, objetos do outro) tem raízes muito profundas, que só poderemos abordar plenamente após uma análise do conceito de estado que estamos propondo. De todo modo, o entendimento de que estados e eventos “não são objetos autênticos” levou a que sua representação seja encarada como naturalmente diferente da representação dos “verdadeiros” objetos. Na situação atual da teoria, para atender à diferença de interpretação, a modelagem orientada a objetos utiliza basicamente três tipos de representação. O principal deles é o *modelo estrutural* (muitas vezes impropriamente denominado *modelo estático*), que representa os objetos, juntamente com suas associações e categorizações; a teoria deste modelo encontra-se magistralmente exposta em [Meyer97]. Estados e eventos são modelados por *diagramas de estados*, em uma versão extremamente difundida devida a David Harel [Harel87], cujo tratamento didático mais completo encontra-se em [Cook94], e que tem fontes remontando diretamente aos autômatos finitos. Também são utilizados *diagramas de seqüên-*

cia e e diagramas de colaboração, explanados em [UML97], para representar o modo como um objeto atua sobre o outro (coloquialmente, trocas de mensagens). Dependendo do autor, cada modo de representar pode assumir diferentes formas, fato que faz da orientação a objetos uma espécie de torre de babel dos formalismos de representação. Atualmente encontra-se em curso uma bem sucedida iniciativa de padronização, por meio da linguagem UML – Unified Modeling Language [UML97b] (que, deve-se acrescentar, contém uma grande quantidade de outros modelos e diagramas, além dos já citados). Entretanto, mesmo com a padronização da linguagem, permanece intocada a questão básica: para modelar estruturas de objetos são necessários três modelos. (Uma observação sobre a grande variedade de modelos utilizados pela orientação a objetos dá direito a uma reflexão: coisas complicadas são sintomas de coisas mal entendidas.)

Uma modelagem orientada a objetos típica:

- inicia-se com a formulação de um modelo estrutural dos tipos de objetos básicos do domínio,
- em seguida, tomando por base esse modelo estrutural, constroem-se máquinas de estados e diagramas de interação, para representar estados e eventos,
- no momento em que o projeto torna-se relativamente estável, os estados e eventos ganham o direito de migrar para os diagramas estruturais⁸¹, “como se fossem objetos”, em geral perturbando os objetos “autênticos” lá estabelecidos, e danificando os modelos já desenvolvidos.

Um projeto real de sistema de informação, aí incluída a modelagem dos objetos, envolve muitas outras atividades além das descritas acima. Mas o processo des-

⁸¹ Comumente, os estados “migram” para os modelos estruturais sob a forma de atributos, e os eventos sob a forma de métodos. Ambas as práticas constituem “estratégias de implementação”, destinadas a incluir os “efeitos” de estados e eventos nos modelos estruturais, mas não se constituem diretamente em “representação” de estados e eventos.

creve as linhas gerais, e ilustra um fato: estados e eventos não têm direito próprio de figurar nos modelos estruturais.

Nesse processo, um fato já chama a atenção: estados e eventos são modelados duas vezes; uma, em seus próprios (!) modelos (diagramas de estado e interação), outra, quando são posteriormente introduzidos nos modelos estruturais - seja (i) nos mesmos modelos dos objetos básicos, seja (ii) em modelos estruturais próprios, ou ainda (iii) em ambos. Costumeiramente, o fato é considerado tão natural e inevitável que sequer desperta atenção da maioria dos teóricos e práticos da modelagem. Pensa-se mesmo que tal procedimento constitui a própria técnica de modelagem. O processo é ensinado em escolas de todos os níveis como teoria-padrão, inquestionável. A complexidade e a rigidez dos modelos resultantes são consideradas como fazendo parte das complicações do mundo, nada tendo que ver com a técnica empregada. Por suposto, o sucesso ou insucesso dos modelos produzidos deve-se à habilidade do modelador, que, se aplicar corretamente as metodologias disponíveis, chegará a bons resultados.

Esta é a situação geral da teoria e da prática da modelagem dos sistemas de informação, como ocorre em virtualmente todos os quadrantes. Nossa investigação nasceu, exatamente, de uma grande insatisfação com a necessidade do uso de tantos modelos e com a complexidade dos resultados até então obtidos (se no mundo só existem objetos⁸², por que não basta o modelo de objetos para modelar qualquer assunto? - se os objetos são simples, por que os modelos são tão complexos?). Na continuação, mostraremos de que modo estados e eventos podem ser formalizados como objetos, deste modo reduzindo a necessidade de formalismo a apenas o modelo estrutural (ao invés de três modelos). Também exploraremos algumas conseqüências lógicas da formalização, em que observaremos alguns fenômenos que podem lançar luz sobre a quase inevitabilidade da teoria dos três modelos.

⁸² Em um conceito amplo de objeto, que também inclui estados e eventos.

16 Implicações dos Conceitos de Estados e Eventos

A principal implicação teórica da formalização dos estados e eventos é a introdução rigorosa do conceito de nível lógico (na realidade, nível semiótico) na modelagem de sistemas de informação, uma vez que um estado – e daí um evento – modaliza (isto é: introduz condições de uso) de objetos que lhe são logicamente subordinados. Tal fato permite – virtualmente, obriga a – separar, de modo formal, dois aspectos fundamentais da modelagem de sistemas:

1. a representação dos objetos básicos pertencentes a um domínio conceitual, e
2. a representação de regras e conhecimento que se aplicam a esses objetos básicos do domínio.

Na realidade, se considerarmos as políticas que atuam sobre um ambiente, a representação da dicotomia *regra/objeto básico* não fica restrita a apenas dois planos conceituais, existindo uma hierarquia de níveis, em que cada nível:

1. define seus próprios objetos básicos

2. e também define regras aplicáveis aos níveis inferiores.

A hierarquia de níveis é logicamente ilimitada, tanto para cima quanto para baixo; a limitação deve-se a razões de ordem pragmática; e o conjunto dos níveis efetivamente definidos constitui o domínio conceitual do sistema representado (Figura 14).

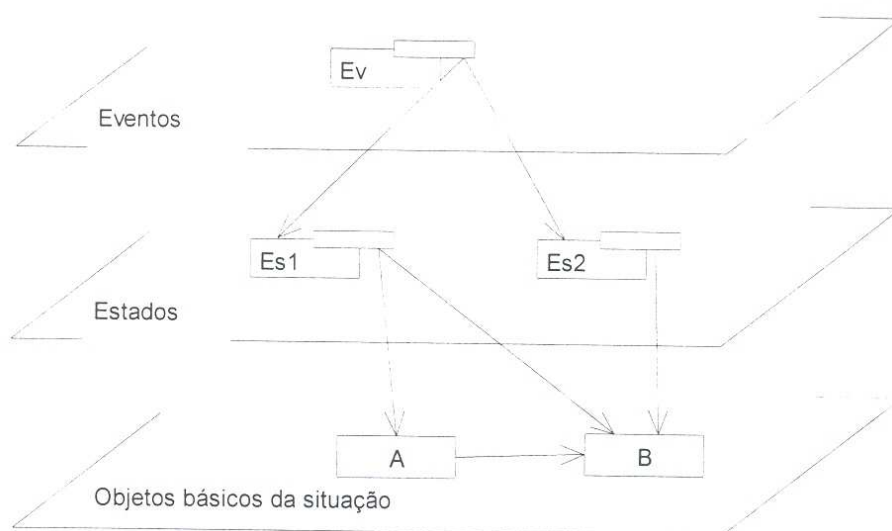


Figura 14: Domínio - Níveis Lógicos de Representação

Esta questão lógica e metodológica – confusão de níveis lógicos de representação – constitui-se em problema de *qualquer* formalismo semiótico, não ficando restrito à orientação a objetos. Na teoria e na prática atuais de modelagem dos sistemas de informação o tema é virtualmente ignorado⁸³. Os manuais – e também os projetos de pesquisa (veja-se o trabalho de Harel para automatizar suas máquinas de estado [Harel00]) – ensinam a colocar toda a modelagem em um único plano; isto é, níveis estruturais diferentes são representados no *mesmo* plano conceitual. Semelhante processo leva a soluções de grande complexidade, além de serem teoricamente inviáveis, como nos ensinam os resultados da lógica. De acordo com o Teorema de

Gödel, as regras aplicáveis a uma estrutura não podem ser definidas, por procedimentos gerais, dentro da própria estrutura. A consequência prática deste princípio é que colocar toda a modelagem em um único plano lógico (modelagem monoplana) constitui-se em um empreendimento fadado ao insucesso. Isto demonstra a fraqueza teórica da modelagem de sistemas de informações como é comumente praticada hoje. A raiz do tratamento inadequado está em não se entenderem os fundamentos lógicos dos processos de representação, e à informalidade a que são relegados estados e eventos. Um modelo de sistema de informação precisa ser pensado como uma hierarquia de estruturas, cada nível estrutural, como já foi observado, codificando regras que se aplicam aos níveis inferiores (Figura 14); para desenvolver essa hierarquia de metaestruturas, é fundamental a definição explícita do que são estados e eventos – objetivo deste trabalho .

Um fato precisa ser destacado. A formalização mostra que estados e eventos residem em níveis lógicos diferentes dos níveis a que pertencem os objetos por eles regidos (Figura 14). Esta é uma lei que independe do formalismo utilizado para modelagem. Deste modo, **inevitavelmente** os estados e eventos **teriam de ser expulsos** do nível dos objetos básicos, mesmo na modelagem tradicional, em que o problema é ignorado. Entretanto, daí **não** se conclui que esses seres errantes (pelos métodos tradicionais) devam migrar para um formalismo diferente⁸⁴ daquele utilizado para representar qualquer objeto.

⁸³ Mas não pelo Dr. Alcides Calsavara, que, em seu trabalho *Virtuosi: Máquinas Virtuais para Objetos Distribuídos* [Cals00], propõe uma implementação formal de eventos.

⁸⁴ Como o são as máquinas de estados e os diagramas de sequência e interação.

Parte VI – Pragmática

Assim, pois, é por seus frutos que os reconheceis.

Mt 7, 20

17 Aplicação de Evento – Análise de Estrutura Organizacional

Este exemplo ilustra a aplicação de eventos e papéis semânticos em um projeto conceitual destinado a organizar as atividades mercadológicas de uma empresa prestadora de serviços de consultoria e assistência técnica. As técnicas deste trabalho se mostraram um poderoso instrumento de análise de funções organizacionais.

O Projeto tem por objetivo:

- (a) formalizar a estrutura de relacionamento da empresa com ser mercado; e, a partir desta análise organizacional,
- (b) definir os procedimentos técnicos de atendimento à demanda por serviços e
- (c) especificar os procedimentos de administração do processo mercadológico.

Como base da organização funcional, foi definida uma função de altíssimo nível, a Função_SEBRAE, classe a partir da qual definiram-se as atividades mercadológicas da empresa, especializadas em dois únicos grupos de operações fundamentais:

- Atividades_Técnicas: Atendimentos, Comunicação, Processamento e Ocorrência;
- Venda (de qualquer Produto, material ou imaterial).

Qualquer função da empresa pode reduzir-se a um desses dois tipos. Tais grupos de operações, na realidade, possuem caráter genérico. As análises mostram que são típicos de uma empresa que dê ênfase à prestação de serviços.

Tecnicamente, a redução das operações do sistema (no caso, a relação da empresa com seu mercado, em sentido amplo) a uma estrutura elementar proporciona que todo o sistema de informações possa ser gerado a partir de relativamente poucos padrões, reduzindo significativamente a complexidade do software. Cada função do sistema está localizada em um único lugar, disponível para ser utilizada pelo serviço que dela necessitar, fato que também reduz a complexidade da implementação.

O material abaixo foi retirado do projeto original⁸⁵ [Sebrae99] e acrescenta comentários explicativos a respeito das soluções adotadas.

No exemplo a seguir, Função_SEBRAE é um evento; Meta, Administrador, Beneficiário e Local_Abstrato são papéis semânticos; Objetivo_de_Negócio, Alguém, Algum e Algures são classes definidoras de posições argumentais (dos papéis semânticos) dotadas de invariantes (condições booleanas) destinados a selecionar os objetos adequados ao preenchimento dos respectivos papéis semânticos.

⁸⁵ Por esse motivo, serão mantidas as convenções gráficas do texto original, que diferem das convenções adotadas neste trabalho, basicamente, pelo emprego de colchetes – [] – para demarcar os parâmetros de predicados, ao invés de parênteses – () –, como aqui utilizados.

17.1 Função_SEBRAE

- Função_SEBRAE [Meta [Objetivo_de_Negócio]]; Administrador [Alguém]; Beneficiário [Algum]; Local_Abstrato [Algures]]

17.1.1 Conceito

Função_SEBRAE reúne uma classe de Processos (basicamente Vendas e Atividades_Técnicas) que concorrem para a realização de um determinado Objetivo_de_Negócio, que é uma Meta do SEBRAE, sob comando de um Administrador autorizado, para benefício de Algum Beneficiário, tipicamente um Setor_da_Economia, Segmento_de_Mercado ou Negócio. Função_SEBRAE tem por base um Local_Abstrato [Algures], por exemplo, o SEBRAE, um posto de serviço, ou a Internet.

Função_SEBRAE caracteriza um grupo de Processos do SEBRAE, dedicados a um determinado tema, oferecidos ao mercado ou à coletividade em geral.

A classe Função_SEBRAE define completamente uma área funcional do SEBRAE, podendo especificar e manipular suas Atividades_Técnicas, Clientes e Produtos. Por consequência, definindo também as Vendas que podem ser feitas. Eventualmente uma Função_SEBRAE que não lide diretamente com Clientes e Produtos deve simplesmente deixá-los sem definição; assim fica registrada a inexistência de Clientes e Produtos para a Função.

Este Projeto define várias Funções_SEBRAE, especializadas em tarefas definidas, as principais sendo: Bolsa_de_Negócios, Informações_sobre_Produtos, Levantamentos_Técnicos, Diagnose, Banco_de_Soluções, Planejamento, Atendimento_de_Balcão, Administração_de_Treinamento. Cada uma dessas funções tem autonomia para definir suas Atividades_Técnicas, Clientes e Produtos (as operações técnicas que realizam tais tarefas são herdadas da classe mais genérica Função_SEBRAE). Outras Funções podem ser definidas com facilidade.

Este é um aspecto importante do Projeto: uma Função_SEBRAE implementa uma atividade mercadológica completa, que contém sua administração técnica e comercial. Ou seja, o Projeto define Funções de natureza mercadológica, voltadas para o atendimento do mercado.

Além disso, é possível centralizar determinadas atribuições em uma das Funções já definidas. Por exemplo, especificar a definição ou criação de Produtos unicamente na classe Informações_sobre_Produtos. Então, as demais Funções utilizam os Produtos definidos por esta. De todo modo, o Sistema contempla apenas Funções de natureza mercadológica.

17.1.2 Exemplo de Aplicação

Exemplos de Funções_SEBRAE: Bolsa_de_Negócios, Banco_de_Soluções, Informações_sobre_Produtos, Diagnose_Empresarial, Atendimento_de_Balcão, Administração_de_Treinamento.

A classe apresentada abaixo, Bolsa_de_Negócios é descendente de Função_Sebrae, a partir de Atividades_Técnicas. Isto é: Função_Sebrae <|-- Atividades_Técnicas <|-- Bolsa_de_Negócios (em que o símbolo <|-- denota a relação de especialização/generalização de classes).

- Bolsa_de_Negócios [Apresentação [Painel_de_Negócios]; Administrador [Gerente]; Beneficiário ['Setor_Industrial'⁸⁶; Local_Abstrato [Algures]]

⁸⁶ O tipo de letra diferenciado advém de uma convenção gráfica do projeto original, denotando um valor passado diretamente como parâmetro.

17.1.3 Classe

Função_SEBRAE [Meta; Administrador; Beneficiário; Local_Abstrato] { Venda, Atividade_Técnica }
• Vendas: Lista [Venda []] /*Vendas próprias da Função_SEBRAE • Operações: Lista [Atividade_Técnica []] /*Atividades_Técnicas próprias da Função_SEBRAE • Clientes: Lista [Cliente] /* Clientes da Função_SEBRAE • Produtos: Lista [Produto] /*Produtos comercializados pela Função_SEBRAE • Agenda_da_Função: Agenda /*especificação de uma Agenda adequada às finalidades da Função_SEBRAE
• Criar_Função_SEBRAE () • Exibir_Dados_Cadastrais_Cliente () • Histórico_Atividade_Técnica_Cliente () /*histórico dos Atendimentos a um dado Cliente • Histórico_Vendas_Cliente () /*histórico das Vendas a um dado Cliente • Salvar_Função_em_BD • Recuperar_Função_de_BD /* a operação abaixo manipula Qualquer objeto particular que pertença a uma Função_SEBRAE. executando uma de suas Operações; o objeto Qualquer é acessado por meio de uma Atividade_Técnica. • Executar_Operação (Atividade_Técnica; Objeto_Qualquer.Operação_X; Parâmetros) /*por exemplo, criar um Agendamento ou exibir uma Agenda /* as operações abaixo manipulam as Atividades_Técnicas pertencentes a uma Função_SEBRAE • Incluir_Atividade_Técnica (Atividade_Técnica []) • Excluir_Atividade_Técnica (Atividade_Técnica []) • Localizar_Atividade_Técnica (Atividade_Técnica [], Parâmetros_de_Seleção) • Processar_Atividade_Técnica (Atividade_Técnica.Operação_X; Parâmetros) /* as operações abaixo manipulam as Vendas pertencentes a uma Função_SEBRAE • Incluir_Venda (Venda []) • Excluir_Venda (Venda []) • Localizar_Venda (Venda [], Parâmetros_de_Seleção) • Processar_Venda (Venda.Operação_X; Parâmetros) /* as operações abaixo processam conjuntos de objetos, as Listas de Objetos de uma Função_SEBRAE, sem alterá-los (alterações só podem ser feitas através de uma Atividade_Técnica) • Emitir_Relatório [Lista_de_Objeto_Qualquer, Parâmetros_de_Seleção] /* inclui Mala_Direta

• Exibir_em_Tela [Lista_de_Objeto_Qualquer, Parâmetros_de_Seleção]
• /* Invariante não definido

17.1.4 Diagrama

Diagrama que representa as funções do SEBRAE.

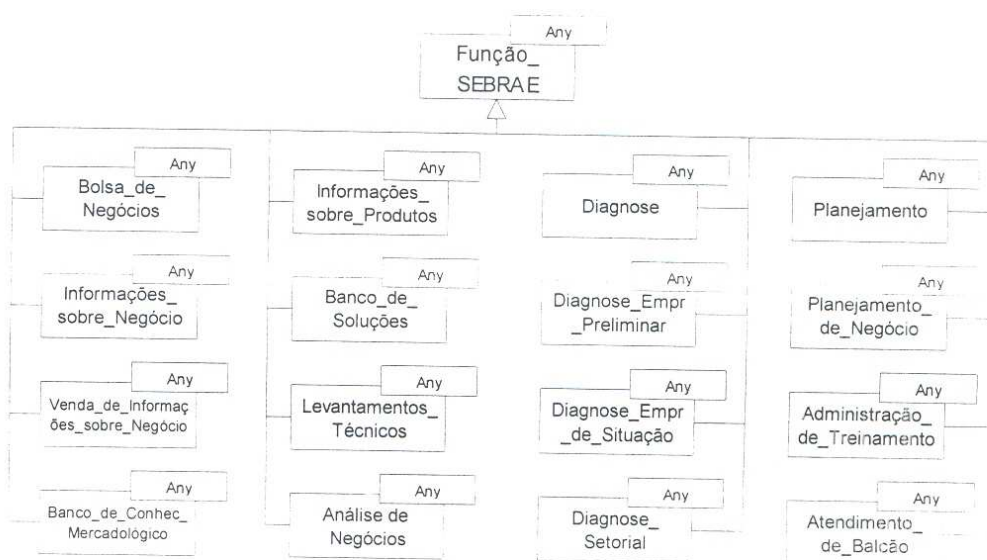


Figura 15: Estados e hierarquia de funções

18 Aplicação de Estados e Eventos – Rede Neural

Este exemplo mostra a construção de uma rede neural, com topologia definida em tempo de criação da rede, inteiramente construída sobre os conceitos de estado e de evento. O conceito básico da aplicação, o *Neurônio*, é definido como o estado de um *Corpo*⁸⁷. Mostraremos a definição das interfaces das classes; o código fonte completo não consta do trabalho devido à grande quantidade de páginas, entretanto, encontra-se à disposição dos interessados, podendo ser consultado mediante solicitação ao autor.

Todas as classes encontram-se extensamente comentadas, motivo pelo qual não acrescentaremos notas explicativas suplementares.

Na arquitetura do sistema, as Topologias de Redes referenciam as Arquiteturas; as Arquiteturas, por sua vez, referenciam os Neurônios, que se organizam em Perceptrons.

⁸⁷ Um *Corpo*, como definido na rede neural implementada, é uma célula dotada de um *axônio* e de uma quantidade indefinida de *dendritos*, também possuindo outras características, como *peso*; o

18.1 Topologia de Redes Neurais – Diagrama

O diagrama exibe a taxionomia das topologias neurais implementadas, maiores informações podem ser obtidas em [Gall94]; cada topologia é descrita por uma classe própria e é dinamicamente instanciada em tempo de execução.

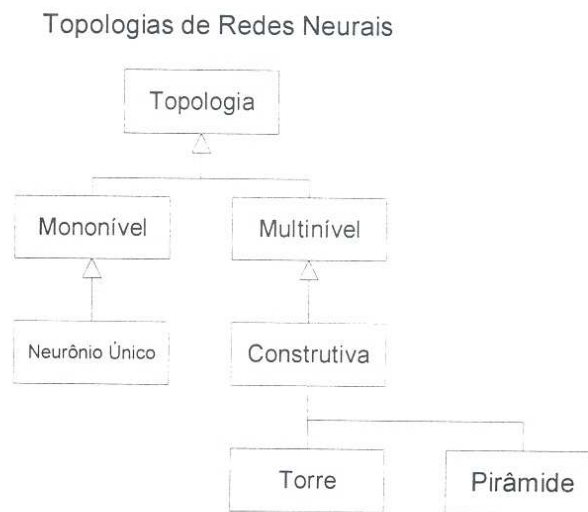


Figura 16: Topologia de redes neurais

Corpo é inerte, somente ganhando vida em virtude da participação em um estado, o *Neurônio* (os termos em itálico, próprios das redes neurais, podem ser consultados em [ArtInt95]).

```

    // topologia adequada (vide classes descendentes):
    // (a) treinado até um máximo de PQt_Iteracoes;
    // (b) que melhor satisfaz PExemplos e Pexem
    // plos_Paradigmaticos;
    // (c) utilizando o algoritmo do Bolso com Catraca.
    // --->> A função é genérica; devido ao polimorfismo,
    // funciona
    // para: perceptron de célula única, construtivos torre
    // e pirâmide;
    // pode ser extendida para outros casos (Jordan e Elman,
    // por exemplo)
    constructor Criar(PQt_Sensores: Integer; PQt_Niveis: Integer); virtual; abstract;
end; // TTopologia_Neural

type TUni_Celular = class(TTopologia_Neural)
    public
        constructor Criar(PQt_Sensores: Integer; PQt_Niveis: Integer); override;
end; //

type TTipo_Torre = class(TTopologia_Neural)
    public
        constructor Criar(PQt_Sensores: Integer; PQt_Niveis: Integer); override;
end; // TTipo_Torre

type TTipo_Piramide = class(TTopologia_Neural)
    public
        constructor Criar(PQt_Sensores: Integer; PQt_Niveis: Integer); override;
end; // TTipo_Piramide

```

18.3 Arquiteturas de Redes Neurais

// Construção de arquiteturas de redes neurais.

```

    constructor Criar(PQt_Sensores: Integer; PQt_Niveis: Integer); override;
end; // Tpiramide

```

18.4 Perceptron

```

// O modelo de Perceptron adotado nesta unit é:
// (segundo Gallant - Neural Network Learning and ...)
// (a) MLP - Multilayer Perceptron;
// (b) com valores discretos, {+1, -1, 0}, para
//     entradas de células e funções de ativação;
//     verdadeiro: +1; falso: -1; desconhecido: 0;
// (c) A computação da rede segue a ordem das células

interface

uses Classes, Estado_Delphi, Patterns_Basicos, Neural_1;

type TPerceptron = class(TEstado)
    // Rede Neural com Sensores e Neurônios
protected
    function Invariante: Boolean;
    // FObjeto é L[ L[Sensor], L[Neuronio] ]
    procedure Regra_de_Estado_Ativo; override;
    // Aciona cada Neuronio, na ordem da Lista
    function Get_Sensores: TList;
    function Get_Neuronios: TList;
    function Saida: Real;
    // Falta generalizar para saída multineural (isto é:
    // multivalores)
public
    property Sensores: TList read Get_Sensores;
    property Neuronios: TList read Get_Neuronios;
    procedure Dispara_Neuronio(PEntradas: TStringList; PNeuronio: TNeuronio);

```

```

    constructor Criar_Basico(PSensores: TList; PNeuronios:
TList);
end; // TPerceptron

type TExemplo = class(TObject)
    // Primeiro elemento da lista FValores é o classificador:
    // (Gallant, pag. 67)
    // +1 : classificação correta
    // -1 : classificação incorreat
protected
    FValores: TStringList;
    function Classificador: Integer;
        // FValores.Strings[0]
    function Valores_Exemplo: TStringList;
        // FValores.Strings[1 até k]
    function String_de_Valores: String;
        // Lista de valores do exemplo, em formato de caracteres
public
    procedure Exibir;
    constructor Criar(PValores: TStringList);
        // Configuração da lista de criação: [classificador, va-
lor 1, ..., valor k]
        // ou seja: classificador do exemplo + valores que en-
tram para sensores
end; // TExemplo

type TTreinamento = class(TEstado)
    // Treinamento de um Perceptron (com Bolso e Catraca):
    // (1) para satisfazer uma lista de Exemplos de Trabalho,
    // (2) com base em uma lista de Exemplos de Treinamento.
    // O algoritmo de treinamento (Regra_de_Estado_Ativo) apli-
ca-se a:
    // (1) 1LP unicelular: neurônio único;
    // (2) MLP: Torre;
    // (3) MLP: Pirâmide.
    // Falta generalizar para:

```



```

// (1) 1LP multicelular;
// (2) lista vazia de exemplos de treinamento.
// [O algoritmo aplica-se a Pirâmide, obtendo Torre e célula
única
// como casos particulares, ainda que a solução tenha per-
corrido o
// caminho inverso: célula, Torre, Pirâmide]
protected
FSucesso_Pesos_Salvos: Integer;
// Número de elementos da última sequência bem sucedida
FPesos_Bolso: TStringList;
// Pesos do Perceptron retidos pelo Algoritmo do Bolso
FSucesso_Pesos_Bolso: Integer;
// Número de vezes consecutivas em que FPesos_Bolso foi
bem
// sucedido durante determinada seção de treinamento
FTreino_Pesos_Bolso: Integer;
// Número de elementos da última sequência consecutiva
que
// gerou inclusão de pesos no bolso
function Invariante: Boolean;
// FObjeto é Perceptron, L[Exemplo_Trabalho],
L[Exemplo_Treinamento]
procedure Regra_de_Estado_Ativo; override;
// Percorre a lista de Neurônios de um Perceptron, trei-
nando
// cada Neurônio, segundo a ordem dos Neurônios na lis-
ta;
// O algoritmo é apropriado para Neurônios dotados da
função Sign,
// como em Gallant (falta analisar se funciona em ou-
tros casos).
function Get_Perceptron: TPerceptron;
function Get_Exemplos: TList;
function Get_Treinamento: TList;
public
property Qt_Valores_no_Bolso: Integer read FTrei-
no_Pesos_Bolso;

```

```

        // Quantidade de valores no bolso para o último nível
treinado

    constructor

        Criar_Basico(PPerceptron: TPerceptron; PExemplos: TList;
PTreinamento: TList);

        // PExemplos é L[ TExemplo ]
end; // Ttreinamento

```

18.5 Neurônio

```

// Modelo básico de neurônio

interface

uses Classes, Estado_Delphi, Patterns_Basicos{,
Form_Exibir_Neuronio};

type TFilamento = class(TObject)
    private
        FValor: Real;
        // Sinal que transita pelo Filamento
        // (sinal de entrada ou saída)
end; // TFilamento

type TSensor = class(TObject)
    protected
        FSinal: String;
        FFilamentos: TList;
        procedure Set_Sinal(PSinal: String);
        // Decodifica Sinal, atribuindo-o a todos os filamentos
    public
        property Sinal: String {read FSinal} write Set_Sinal;
        function Filamento(PFilamento: Integer): TFilamento;
        constructor Criar(PQt_Filamentos: Integer);
end; // TSensor

```

```

type TFibra = class(TObject)
// Dendrito ou Axônio
private
    FPeso: Real;
    FFilamentos: TList;
public
    property Peso: Real write FPeso;
    constructor Criar; virtual;
    procedure Set_Filamento(PFilamento: TFilamento);
        // Inclui Filamento
end; // TFibra

type TDendrito = class(TFibra)
private
    function Saida: Real;
public
end; // TDendrito

type TAxonio = class(TFibra)
public
    constructor Criar; override;
end; // TAxonio

type TCorpo = class(TObject)
private
    FAxonio: TAxonio;
    FDendritos: TList;
    FPeso: Real;
    // Default = 0
    // Contribuição de FCorpo a FNeuronio
    procedure Soma_Pesos(PPeso_Corpo: Real; PPesos_Dendritos:
TStringList);
        // Soma de Pesos durante Treinamento (MLP com valores
discretos?)
public
    property Peso: Real write FPeso;

```

```

    constructor Criar(PPesos: array of Real);
    constructor Criar_Basico(PQt_Dendritos: Integer);
    function Filamento(PDendrito: Integer; PFilamento: Integer): TFilamento;
    function Filamento_Saida(PFilamento: Integer): TFilamento;
    function Quantidade_Dendritos: Integer;
    procedure Set_Filamento_Axonal(PQt_Filamentos: Integer);
        // O corpo é criado com 1 filamento axonal (filamento 0)
        // esta procedure acrescenta novos filamentos axonais
end; // TCorpo

```

```

type Tligacao_Neural = class (TEvento)
    private
        function Get_Mensagem: TMensagem;
        function Get_Filamento_Origem: TFilamento;
        function Get_Filamento_Destino: TFilamento;
        function Invariante: Boolean;
        procedure Regra_de_Estado_Ativo; override;
    public
        constructor Criar_Basico(PFilamento_Origem: TFilamento;
        PFilamento_Destino: TFilamento);
end; // Tligacao_Neural

```

```

type TNeuronio = class(TEstado)
// Neurônio é o estado de um Corpo
// As Ligacoes_Neuralis estão sendo interpretadas como
// "conhecidas" pelo Corpo (atributos dele)
    private
        FLigacoes_Recepcao: TList;
        // Lista das ligações neurais de recepção de sinal
        function Invariante: Boolean;
        // Invariante: FObjeto é TCorpo
        procedure Regra_de_Estado_Ativo; override;
        function Corpo: TCorpo;
        function Theta: Real; virtual;
        function Entrada: Real;

```



```

        // Versão simplificada de Entrada, do modo padrão da te-
oria
        function Ativacao: Real; virtual; abstract;
    public
        procedure Exibir;
        procedure Set_Ligacao(PLigacao: TLigacao_Neural);
        procedure Set_Peso(PPeso: Real);
            // Atribui Peso ao Corpo
        procedure Set_Todos_Pesos(PPesos: TStringList);
        function Saida: Real;
            // Versão simplificada de Saida: função real
        function Pesos: TStringList;
            // Lista de pesos dos Dendritos e do Corpo do Neurônio
            // Pesos[0] = peso do Corpo; demais pesos são dos den-
dritos
            // definido como TStringList para superar dificuldades
            // com arrays de números reais
        function Quantidade_Dendritos: Integer;
        procedure Exibir_Pesos;
        procedure Atualiza_Pesos(PPeso_Corpo: Real; PPe-
sos_Dendritos: TStringList);
            // Atualização de Pesos durante Treinamento (discreto?)
        constructor Criar(ParObjeto: TObject); override;
end; // TNeuronio

```

19 Desenvolvimento Futuro

O potencial de aplicação das idéias apresentadas neste trabalho, por um público mais amplo, depende de maior formalização das propostas e do desenvolvimento de metodologias e de material de ensino, que tornem os princípios apresentados mais acessíveis a técnicos aplicadores. Na continuação, propomos alguns projetos com essa finalidade. Acrescentamos também a proposta de dois projetos de natureza mais teórica – Semântica de Predicados Modais e Gramáticas de Línguas Naturais; como também incluímos um projeto de interesse prático, a Especificação e Implementação de Agentes Cognitivos.

Metodologia de Modelagem de Sistemas de Informação

Proposta de uma metodologia baseada em eventos e nos papéis semânticos de seus participantes; um aspecto positivo da abordagem é que ela implica a organização quase que obrigatória dos modelos em níveis conceituais, e a modularização dos conceitos empregados; uma metodologia deste tipo leva a um grande rigor conceitual na representação de domínios, mas é extremamente exigente em relação ao prepa-

ro técnico dos modeladores, devendo ser fortemente apoiada por recursos de especificação; uma tal metodologia fornece recursos conceituais para explorar o potencial da orientação a objetos como ferramenta de especificação de sistemas em diferentes níveis de abstração.

Formalização dos Papéis Semânticos

Sistematização e implementação de diferentes estruturas de papéis semânticos, para uso na especificação de eventos/predicados; este é um dos recursos a serem utilizados para formalizar a estrutura dos argumentos de eventos/predicados, sendo importante meio para a codificação de semântica em modelos de sistemas de informação e em modelos semânticos em geral.

Especificação de Processos

Especificação e implementação de padrões de eventos que permitam o tratamento formal de concorrência de processos e séries temporais; esta questão está intimamente associada ao tratamento de padrões de eventos como máquinas de inferência lógica a partir do controle de variáveis ambientais; acreditamos ser esta uma das mais poderosas aplicações das técnicas propostas.

Especificação Declarativa de Sistemas

A especificação de estados e eventos, segundo as técnicas propostas, possibilita que grande parte das definições de sistemas seja feita de modo declarativo (definição de expressões booleanas e procedimentos que se enquadram em uma lógica previamente existente: a lógica de estado); tais recursos podem ser estendidos para complementar linguagens imperativas, dotando-as de comandos declarativos para a especificação de eventos (os quais, de acordo com a proposta de meto-

dologia acima citada, tornam-se os elementos fundamentais da modelagem semântica), permitindo que a modelagem de sistemas possa concentrar-se na especificação lógica dos processos.

Especificação de Estruturas de Informação

Criação de uma biblioteca genérica de estruturas de informação (ou de dados) que parta da definição genérica de grafo e, por especialização, especifique as demais estruturas (lineares e não lineares); uma abordagem alternativa mais abstrata é partir dos conceitos de coleção não-ordenada, conjunto e relação de ordem (linear e não linear), e daí derivar as estruturas acima citadas; uma tal conceituação, e suas respectivas implementações, além de teoricamente mais adequada do que várias propostas em uso, pode ser utilizada como material de ensino nos estudos de graduação da área de informática (basicamente, sistemas de informação, ciência da computação, engenharia da computação).

Semântica de Predicados Modais

A formalização dos conceitos de estados e eventos possibilita um modo de abordar predicados do tipo *Pedro acredita que Maria não quer ir à escola*, por meio da formalização, com simplicidade, dos vários níveis de modalização envolvidos, como em: (*Crença* (*Pedro*, *Não* (*Vontade* (*Maria*, *Ida* (*Maria*, *escola*))))); deste modo, a semântica de cada predicado – o que inclui a seleção e o papel semântico dos argumentos – pode ser analisada isoladamente, como também pode ser analisado o comportamento conjunto dos predicados e de seus argumentos; eventualmente o tratamento modal possa ser estendido à análise dos quantificadores $\forall$ e $\exists$.

Gramática de Línguas Naturais

Um projeto particular do autor (uma intenção oculta por trás de toda a investigação) é a aplicação das idéias deste trabalho à formalização da teoria gramatical das línguas naturais, sob o ponto de vista de um agente cognitivo; acreditamos ser possível desenvolver um tratamento integrado (não entendemos como a análise lingüística não seja tratada de modo integrado) das morfologia, sintaxe, semântica e pragmática das linguagens naturais, tomando por base o conceito de predicado situado em uma estrutura cognitiva de tipo peirceano.

Linguagem Genérica para Formalização de Conhecimento

Definição e implementação de uma linguagem genérica para especificação de conhecimento (linguagem semiótica), a partir de uma perspectiva computacional; este trabalho é o primeiro passo em tal direção⁸⁸.

Especificação e Implementação de Agentes Cognitivos

Continuação de um projeto já iniciado, que resultou na implementação de uma estrutura cognitiva de um agente planejador, e que pode resultar na especificação de um modelo genérico de agente cognitivo.

⁸⁸ Para a apreciação de uma linguagem desse tipo, deve-se traçar uma analogia com as línguas naturais (português, inglês etc.). Uma língua natural é evolutiva e historicamente adaptada à expressão de conteúdos cogitáveis por seres humanos, ainda que os conceitos específicos possam variar de pessoa para pessoa, segundo a experiência, os pressupostos e os objetivos de cada um. Também uma linguagem semiótica deve apresentar essa característica de generalidade: precisa proporcionar os meios para representação de qualquer conteúdo cognitivo, sem intervir ou propor critérios para o julgamento de qualquer tipo de realidade ou adequação desse conteúdo – ou seja, deve ser uma linguagem de “mecanismos”, e não de “conteúdos”.

20 Conclusão

– Por quien Dios es, Sancho – dijo a esta sazón don Quijote –, que concluyas con tu arenga; que tengo para mí que si te dejasen seguir en las que a cada paso comienzas, no te quedaría tiempo para comer ni para dormir; que todo le gastarías en hablar.

Miguel de Cervantes Saavedra, Don Quijote de la Mancha

Resumo dos principais resultados metodológicos.

O trabalho mostra (fato insuficientemente visível de dentro da teoria-padrão) que a teoria normal dos sistemas de informações está envolvida em soluções artificiosas e excesso de formalismos por falta de uma abordagem realizada a partir de seus fundamentos. Teóricos que atuam na área erguem construções sem considerar problemas de base [UP99] [Odell95] [Cook94]. Assim o fazendo, adotam a atitude de físicos aplicadores ou de engenheiros - edificam teorias sobre conceitos puramente empíricos, considerados de modo acrítico, e (como esperamos ter mostrado) inadequados.

Algumas implicações da pesquisa:

1. A teoria-padrão é *incompleta*, considera somente eventos que são *transições* de estados (*transição de estado* chega a ser tomada como *definição* de *evento*). A formalização que propomos amplia o conceito para abranger também eventos que são *conjunções* ou *disjunções* de estados e/ou eventos, além de alargar o próprio conceito de transição. Ademais, a teoria-padrão determina (seguramente por dificuldades na conceituação de estado) que os eventos ocorrem instantaneamente. No nosso caso, os eventos podem ter duração (na realidade o conceito de *tempo* é aí inexistente, fazendo sentido falar-se apenas em *sucessão* - num senso mais amplo, talvez *transformação*).
2. Sob um prisma conceitual, a teoria-padrão pressupõe que os objetos simplesmente “possuem” seus estados e são conscientes deles - pressuposto considerado auto-evidente e nunca questionado. Esta é uma hipótese *desnecessária* (se nos ativermos ao “possuem” acima, é falsa) ao que se possa entender por interpretação da realidade, além de ser absolutamente *insuficiente* para a solução de virtualmente todas as situações práticas (exceto exemplos de livros didáticos). Tem-se aqui o ponto mais fundamental de nossa proposta, aquele que mostra as conexões das técnicas de modelagem de conhecimento com resultados de ciências afins.
3. Para finalidades de implementação (não como visão conceitual), as abordagens mais refinadas da teoria-padrão (em que estados são modelados como objetos que manipulam objetos básicos, evitando-se o caminho inverso) adotam a solução de implementar diretamente máquinas de estados (local onde habitualmente vivem estados, eventos e transições, quando da análise conceitual) como modelos estruturais. Entretanto, máquinas de estados são *ferramentas de trabalho*⁸⁹ que reúnem os mesmos conceitos dos modelos estruturais, só que de um modo *incoerente*: (i) diferenciam estados e condições de

⁸⁹ Como observa o Prof. Gudwin, são ferramentas não apenas de *análise*, mas também de *síntese*.

guarda - que são a mesma coisa -; (ii) em muitas circunstâncias, confundem eventos e operações - que são coisas distintas.

4. Isto ocorre porque as máquinas de estados originaram-se da teoria da computação, com a finalidade de implementar algoritmos, em que conta a “execução cega”, somente depois foram aplicadas às técnicas de representação, em que estão em jogo “interpretação” e “julgamento”. Assim, a própria idéia dessas máquinas é alheia a trabalhar com situações abertas ou complexas; quando se aplicam a problemas de modelagem, é sempre necessária uma simplificação artificial. Por exemplo, apenas lidam adequadamente com estados definidos para objetos únicos, tornando-se complicadas e antinaturais para estados que se referem simultaneamente a vários objetos - neste caso são necessárias várias máquinas e a modelagem torna-se pouco realista (“antinaturais” e “pouco realista” fazem sentido segundo nossa proposta). Deste modo, a implementação direta das máquinas de estados leva, em situações normais, a uma modelagem *não correspondente* à realidade pretendida. Conceitos de uma tradição nem sempre se ajustam adequadamente a outra - não há traduções unívocas genéricas entre máquinas de estados e diagramas estruturais.
5. Existe portanto uma questão de princípio. Encarar a solução de estados, eventos, transições etc. como sendo uma questão de transportar diretamente máquinas de estados para dentro de modelos estruturais implica preservar os problemas apontados nas duas objeções precedentes. Por razões básicas, não se podem compatibilizar, mediante procedimentos gerais, máquinas de estados com modelos estruturais. Nenhuma solução *regular* pode daí derivar. Acrescente-se que o enfoque pressupõe aplicar um tratamento genérico a estados e eventos, como se tais conceitos estivessem bem esclarecidos mesmo na teoria-padrão (acreditamos poder mostrar que a teoria-padrão toma esses conceitos como dados, sem fundamentá-los). Assim, as propostas que têm por finalidade traduzir diretamente e algoritmicamente máquinas de estados em modelos estruturais sempre o fazem recorrendo a *constructos arbitrários*, teoricamente *pouco consistentes*; e os resultados mostram-se de escasso valor

prático. Como seria de se esperar, neste ponto de máxima dificuldade ocorrem os maiores investimentos conceituais de solução da teoria padrão - uma proposta bem elaborada encontra-se em [FerRb98]. Entretanto, este é um programa fadado ao insucesso - ao menos quanto à simplicidade, coerência conceitual interna, compatibilidade de resultados com outras áreas do conhecimento científico e aplicabilidade prática. Tal fato só não é facilmente reconhecível por falta de um critério que permita apreciar a viabilidade dos programas de pesquisa e a qualidade dos resultados obtidos - critério que só pode ser proporcionado por uma visão a partir dos fundamentos. Qualquer resultado positivo conseguido neste ponto levará, retroativamente, à reavaliação dos procedimentos de modelagem em vigor.

6. Nossa abordagem demonstra que a modelagem de estados e eventos pode ser executada como uma modelagem regular da orientação a objetos, mediante procedimentos ordenados, sem recurso a expedientes *ad-hoc*. Não é necessário mais do que o modelo normal de objetos para todo o processo de modelagem. Máquinas de estados e diagramas de interação são úteis, mas pertencem a processos de descoberta (nem sempre necessários), não fazem parte da técnica fundamental de modelagem de objetos (sempre necessária).
7. A proposta simplifica e organiza o processo de modelagem, graças à utilização de uma hierarquia aberta de estruturas dotadas de funções especializadas (com solução distinta da implementação direta de máquinas de estado), sem necessidade de alterar o modelo de objetos básicos para inclusão de estados e eventos. Em consequência, esclarece também o conceito de *sistema*.
8. A formalização dos conceitos de estados e eventos levou à criação de *padrões* (*patterns*), que são esquemas de soluções reaplicáveis em contextos específicos. Esses patterns evidenciam a existência de famílias de estruturas que independem do nível metaestrutural em que ocorrem - fato talvez revelador de uma classificação de categorias de eventos reais. Do ponto de vista prático, a lógica reutilizável dos patterns, somada aos métodos de análise da abordagem,

possui implicações na engenharia de software, principalmente em especificações formais de sistemas.

9. A abordagem desenvolve uma apreciação crítica dos métodos utilizados, entre os quais: (i) conceito estruturalista de modelo e objeto - possivelmente o que concede maior liberdade semântica, permitindo uma caracterização puramente formal de conteúdo, independente de substância; esta concepção fornece a base para considerar-se um modelo como produto independente de interpretação; a interpretação pertenceria a procedimentos heurísticos e pragmáticos, não seria necessária ao modelo considerado como estrutura; (ii) discussão filosófica dos pressupostos fundamentais da proposta e também dos comumente avocados pela teoria-padrão - para a teoria-padrão não são necessários mais do que o empirismo imediato (“realidade observável”) e o behaviorismo (“comportamento” de objetos como resposta a estímulos); tais pressupostos, uma vez expostos, são logo abandonados pelos autores; no entanto, produzem consequências danosas aos espíritos mais crédulos que se atêm à autoridade da palavra escrita (esses, quanto melhor aprendem, pior modelam); (iii) apresentação da modelagem como uma ontologia, refinada e/ou revisada por métodos semióticos; compreensão do processo de modelagem, nos contextos ontológico e semiótico.

21 Bibliografia

Os números que compõem os indicadores da referências (final dos textos da primeira coluna) indicam, com o maior grau de precisão possível, o ano em que a versão original foi escrita, que pode não coincidir com o ano em que foi escrita a versão citada (segunda coluna).

1. [Arisbe01] Arisbe: The Peirce Gateway; www.door.net/arisbe/arisbe.htm
2. [ArtInt95] Artificial Intelligence – A Modern Approach; Stuart Russell, Peter Norvig; Prentice Hall; 1995
3. [Badiou88] O ser e o evento; Alain Badiou; Jorge Zahar Editor, Editora UFRJ; 1996.
4. [Bart64] Elementos de semiologia; Roland Barthes; Editora Cultrix Ltda.; 1992
5. [Booch94] Object-oriented analysis and design with applications; Grady Booch; Addison-Wesley Publishing Company; 1994.
6. [Cals00] Virtuosi: Máquinas Virtuais para Objetos Distribuídos; Alcides Calsavara; PUC-PR, Centro de Ciências Exatas e de Tecnologia; 2000
7. [Chafe75] Significado e estrutura lingüística; Wallace L. Chafe; Livros Técnicos e Científicos Editora S. A.; 1979

8. [Coad95] Object models: strategies, patterns, and applications; Peter Coad, David North, Mark Mayfield; Prentice-Hall; 1995.
9. [CompSem01] Computational Semiotics; Ricardo R. Gudwin; UNICAMP;
www.dca.fee.unicamp.br/~gudwin/compsemio
10. [Cook94] Designing object systems: object-oriented modelling with Syntropy; Steve Cook, John Daniels; Prentice Hall International; 1994
11. [DicS79] Dicionário de Semiótica; Algirdas Julien Greimas, Joseph Courtés; Editora Cultrix
12. [DiEL72] Dicionário enciclopédico das ciências da linguagem; Oswald Ducrot, Tzvetan Todorov; Editora Perspectiva S. A.; 1988
13. [Eco64] A estrutura ausente; Umberto Eco; Editora Perspectiva S. A.; 1987
14. [Eco76] Tratado geral de Semiótica; Umberto Eco; Editora Perspectiva; 1991.
15. [Eco84] Semiótica e Filosofia da Linguagem; Umberto Eco; Editora Ática S. A.; 1991.
16. [FerRb98] Padrão State Reflexivo: Refinamento do Padrão de Projeto State para uma Arquitetura Reflexiva; Luciane L. Ferreira, Cecília M. F. Rubira; UNICAMP; Instituto de Computação; 1998
17. [Gall94] Neural Network Learning and Expert Systems; Stephen I. Gallant; The MIT Press, 1994
18. [Harel00] On the inheritance of state-based object behavior; David Harel;
www.eiffel.com/tools/usa/; 2000
19. [Harel87] Statecharts: a visual formalism for complex systems; Science of Computer Programming 8; 1987
20. [Hausd37] Set Theory; Felix Hausdorff; Chelsea Publishing Company; 1962.
21. [Hjel59] Ensaios Lingüísticos; Louis Hjelmslev; Editora Perspectiva S. A.; 1991
22. [Jacob92] Object-Oriented Software Engineering - A Use Case Driven Approach; Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Övergaard; Addison-Wesley Publishing Company; 1992
23. [Joliv72] Tratado de Filosofia III: Metafísica; Livraria Agir Editora; 1972
24. [Kuhn62] A Estrutura das Revoluções Científicas; Editora Perspectiva S. A.; 1990
25. [LexSem] Computational Semiotics Group - DCA-FEEC-UNICAMP_Léxico.htm

26. [Meyer97] Object-Oriented Software Construction, Second Edition; Bertrand Meyer; Prentice Hall PTR; 1997
27. [Nöth96] A semiótica no século XX; Winfried Nöth; ANNABLUME editora, comunicação; 1996
28. [Odell95] Object-oriented methods: a foundation; James Martin, James Odell; Prentice-Hall, Inc.; 1995
29. [Odell96] Object-oriented methods: pragmatic considerations; James Martin, James Odell; Prentice-Hall, Inc.; 1996
30. [Peir35] Semiótica; Charles Sanders Peirce; Perspectiva; 1977
31. [Peres84] Elementos para uma Gramática Nova; João Andrade Peres; Livraria Almedina – Coimbra – Portugal; 1984
32. [Piag79] O Estruturalismo; Jean Piaget; DIFEL - Difusão Editorial S. A.; 1979
33. [Popper63] Conjecturas e Refutações (O Progresso do Conhecimento Científico); Karl R. Popper; Editora Universidade de Brasília, 1982
34. [Popper71] Conhecimento Objetivo – Uma Abordagem Evolucionária; Karl R. Popper; Editora Itatiaia Ltda; 1975
35. [Quine70] Philosophy of Logic; W. V. Quine; Prentice Hall, Inc.; 1970
36. [Rumb94] Modelagem e Projetos Baseados em Objetos; James Rumbaugh, Michael Blaha, William Premierlani, Frederick Eddy, William Lorensen; Editora Campus Ltda.; 1994.
37. [Russ59] Meu Desenvolvimento Filosófico; Bertrand Russell; Zahar Editores S. A.; 1980
38. [Saus16] Curso de Lingüística Geral; Ferdinand de Saussure; Publicações Dom Quixote; 1978
39. [Sebrae99] Sistema Mercadológico do SEBRAE/ES – Projeto Conceitual; Orlando Alcantara Soares; brochura; 1999
40. [SwTa99] Ontologies; William Swartout, Austin Tate; in IEEE Intelligent Systems, Jan/Feb 1999
41. [Todor77] Teorias do Símbolo; Tzvetan Todorov; Papirus Editora, 1996
42. [UML97] UML - Notation Guide, version 1.0; Rational; 1997.
43. [UP99] The Unified Software Development Process; Ivar Jacobson, Grady Booch, James Rumbaugh; Addison Wesley; 1999

44. [Whith68] Algumas idéias científicas; Alfred North Whitehead; Editorial Gredos, S. A. ;
1919
45. [Wittg68] Tractatus logico-philosophicus; Ludwig Wittgenstein; Nacional; 1968

Índice Detalhado

Resumo.....	11
Abstract.....	13
Tema.....	15
1 Introdução	16
Parte I – O Problema	19
2 Dificuldades dos Formalismos de Representação	20
2.1 Inviabilidade da Modelagem Monoplana	20
2.2 A Pesquisa.....	21
2.3 Delimitação do Tema.....	23
2.4 A Proposta.....	25
Parte II – Epistemologia	27
3 Significado	28
4 Teoria do Signo de Peirce.....	31
4.1 Signo, Objeto e Interpretante	31
4.2 Conceito de Semiose.....	34
4.3 Aplicação do Conceito de Semiose	36
4.4 Implicações da Teoria do Signo.....	38
Parte III – Ontologia.....	40

5	A Estrutura.....	41
5.1	Conceito de Objeto	46
5.2	Representação de Objeto	50
5.2.1	Representação Textual	51
5.2.1.1	Associações	51
5.2.1.2	Operações.....	52
5.2.1.3	Classe Exemplo - CIDADAO	53
5.2.1.4	Observações sobre Classe	53
5.2.2	Representação Gráfica	54
6	O Modelo	58
6.1	O que é um Modelo	58
6.2	Modelo e Realidade	62
6.3	Crítica: Modelo, Ontologia e Base de Conhecimento.	64
	Parte IV – Fenomenologia.....	68
7	Predicado.....	69
7.1	Conceito Tradicional de Predicado.....	69
7.2	Conceito Lógico de Predicado.....	71
7.3	Conceito Semântico de Predicado	74
8	Interpretação de um Predicado.....	78
9	Exemplo de Aplicação de Predicado e Papéis Semânticos.....	84

9.1	Venda	86
9.1.1	Conceito	86
9.1.2	Exemplo de Aplicação	86
9.1.3	Classe	87
Parte V – Ética		88
Estado		89
10	Noção Tradicional de Estado	90
11	Conceito Formal de Estado	94
11.1	Base Intuitiva	94
11.2	Observações sobre o Conceito	96
11.3	Representação Gráfica	96
11.4	Aspectos Semânticos do Conceito de Estado	98
11.4.1	Semântica Fundamental de Estado	98
12	Definição Formal de Estado	102
12.1	Estado: Formalização em Eiffel	105
12.2	Formalização em Object Pascal	108
12.3	Exemplo de Aplicação do Conceito de Estado	112
Evento		113
13	Noção Tradicional de Evento	114
14	Definição Formal de Evento	116

14.1	Especificação Formal de Evento com Implementações	117
15	Consequências das Noções Tradicionais.....	119
16	Implicações dos Conceitos de Estados e Eventos	122
Parte VI – Pragmática		125
17	Aplicação de Evento – Análise de Estrutura Organizacional.....	126
17.1	Função_SEBRAE	128
17.1.1	Conceito	128
17.1.2	Exemplo de Aplicação	129
17.1.3	Classe	130
17.1.4	Diagrama.....	131
18	Aplicação de Estados e Eventos – Rede Neural	132
18.1	Topologia de Redes Neurais – Diagrama	133
18.2	Topologias de Redes Neurais – Classes	134
18.3	Arquiteturas de Redes Neurais	135
18.4	Perceptron	137
18.5	Neurônio	140
19	Desenvolvimento Futuro	144
20	Conclusão.....	148
21	Bibliografia.....	153