

# Técnicas Para Representação Computacional de Conhecimento

Por Sergio Navega  
[snavega@intelliwise.com](mailto:snavega@intelliwise.com)  
Intelliwise AI Research  
Infoimagem 2005

## Resumo

Neste artigo apresento algumas técnicas para representar conhecimento computacionalmente. Início com uma exposição teórica que procura categorizar o que são símbolos e de que forma eles são tradicionalmente utilizados para representar conhecimento explícito. Em seguida, observaremos os frames, as redes semânticas, os formalismos que usam a lógica e finalmente introduziremos as ontologias. A principal intenção do artigo não é descrever esses conceitos profundamente, mas apenas torná-los conhecidos pelo leitor.

## Introdução

Há muito tempo as empresas passaram a ser avaliadas por mais do que apenas seus ativos tangíveis. Muitas empresas atuais dispõem de notável valor comercial, mesmo sem possuir muitos imóveis ou maquinários próprios. Essas empresas têm boa parte de seu valor atribuído àquilo que conhecem. O conhecimento de uma empresa (o seu capital intelectual) representa atualmente um diferencial de importância vital para o seu sucesso. Neste artigo, apresento alguns conceitos relacionados às múltiplas formas de representação computacional de conhecimento. Dada a vastidão do assunto e a limitação de espaço, não é possível fazer uma exposição detalhada.

## Conhecimento e Informação

Em termos rigorosos, tudo o que veremos a seguir deveria se referir não a conhecimento, mas sim a informação. Conforme defendo em um outro artigo (veja Navega 2002), conhecimento requer a figura de um agente inteligente com potencial para ação. Livros, documentos e computadores não contêm conhecimento, contêm apenas informação. Esta se transforma em conhecimento a partir do momento em que entra na mente de um agente (nos dias de hoje, uma pessoa) que possa fazer uso dela.

## Lidando Com Dois Níveis

Além da distinção informação/conhecimento, é preciso mencionar que nem tudo o que as pessoas conhecem pode ser representado de forma simbólica. Em um outro artigo (veja Navega 2000) mostro que boa parte do conhecimento humano não tem como ser explicitado, constituindo-se de elementos que não podem ter expressão simbólica. Esse tipo de conhecimento é conhecido como “sub-simbólico” e tem relação com a forma como nossa cognição categoriza o mundo (veja mais detalhes em Harnad 1990).

## O Que São Símbolos?

Símbolos são representações que usam marcações em um suporte físico qualquer. Essas representações “substituem” (fazem-se passar) por um objeto ou processo externo. Os símbolos fazem referência a uma “entidade” externa, que pode ser um objeto, uma qualidade, um processo, evento, quantidade, etc.

Os símbolos (assim como informação, um conceito associado) precisam necessariamente de um suporte físico, mas não podem ser confundidos com esse suporte físico. Símbolos são construções abstratas, não tem massa nem cor, sua “existência” necessariamente requer a presença de um *observador*. É esse observador que faz a associação entre o símbolo e aquilo que ele representa. Essa associação é conhecida como designação ou denotação. Nesta nossa análise, há dois níveis de observadores: um observa o símbolo (ou “pensa” nele) e o associa ao objeto externo. Um segundo observador (nós) observa esse sistema (objeto + observador) e compreende a associação feita<sup>1</sup>.

Aquele primeiro observador realiza uma tarefa que é mais complexa do que se pode pensar à primeira vista. Visualizar um objeto externo requer desse observador uma capacidade de interpretação, algo que somente um sistema perceptual pode executar. Percepção (ou interpretação) é um dos pontos normalmente esquecidos em discussões de sistemas simbólicos<sup>2</sup>. A associação feita entre um símbolo e aquilo que ele denota é uma operação mental, só existente na mente do observador, e que requer deste uma sofisticada operação cognitiva.

### **Símbolos Arbitrários e Icônicos**

Há duas espécies básicas de símbolos: os arbitrários e os icônicos. Os símbolos arbitrários (ou amodais) são representações que não guardam nenhuma associação intrínseca com o objeto que representam. A palavra “LIVRO”, por exemplo, não tem nenhuma característica estrutural que a associe ao objeto correspondente. Já o símbolo /OO/, por exemplo, guarda alguma proximidade perceptual (similaridade estrutural) com o objeto “óculos”. Essa similaridade é suficiente para que esse símbolo possa ser usado em contextos onde a identificação visual faça o observador lembrar-se do objeto real. Praticamente todos os sistemas computacionais atuais só se utilizam de símbolos amodais, já que os símbolos icônicos requerem uma difícil etapa de interpretação perceptual, um dos grandes problemas enfrentados pelos pesquisadores que lidam com inteligência computacional. Mais informações sobre este tópico pode ser vista em Barsalou (1999).

### **O Acoplamento Causal**

Objetos e processos no meio ambiente estão ligados entre si pela causalidade que move o universo<sup>3</sup>. Essas seqüências de causa-efeito são suficientemente regulares para que possamos representá-las com descrições lingüísticas, equações ou expressões lógicas. Em outras palavras, representamos as seqüências causais do universo através de grupos associados de símbolos. Disso decorre uma interessante propriedade da relação entre alguns símbolos: eles mantêm entre si um tipo de ligação que “duplica” a correspondente conexão externa. Dizemos que esses símbolos estão causalmente ligados, ou seja, que dispõem de um *acoplamento causal*. Esse acoplamento está representado em nossa mente (ou no interior de um processador de informações, como um computador) por associações “paralelas” à correspondente externa. A solidez dessa conexão interna tem, como única garantia, apenas a regularidade com que descobrimos regras genéricas do universo<sup>4</sup>.

Um sistema informatizado de controle de temperatura, por exemplo, recebe informações de um sensor que são digitalizadas e transformadas em símbolos. São esses símbolos (por exemplo, 23°) que farão o sistema acionar o ar condicionado (ou seja, executar uma ação no mundo real). Há pouca chance de esse processo dar errado, por isso não há muita preocupação em haver descompasso entre a temperatura do ambiente (causalidade externa) e aquilo que o sistema tem representado (causalidade interna). Contudo, basta que alguém acenda um isqueiro logo abaixo do sensor para introduzir uma desvinculação dessas causalidades.

O histórico de como se obtém representações simbólicas que guardem proximidade causal com o ambiente externo é muito maior do que parece: é atividade que nós humanos desempenhamos há milênios. Hoje em dia, esse tipo de atividade tem um nome definido: é o método científico. Através do processo de investigação da ciência conseguimos determinar as ligações (modelos) entre conceitos relativos ao evento que analisamos. Isso se transforma em representações lingüísticas e equações matemáticas (portanto, símbolos) que procuram preservar essa relação de causa-efeito.

## Usos dos Símbolos

Vimos até agora o uso de símbolos levando em conta o efeito de representar objetos e propriedades do ambiente na mente de um agente inteligente. Isto permite que esse agente efetue operações sobre esses símbolos e que disto conclua algo. Computadores já fazem isso, só que eles ainda não aprendem esses processos por experiência própria (precisam ser meticulosamente programados).

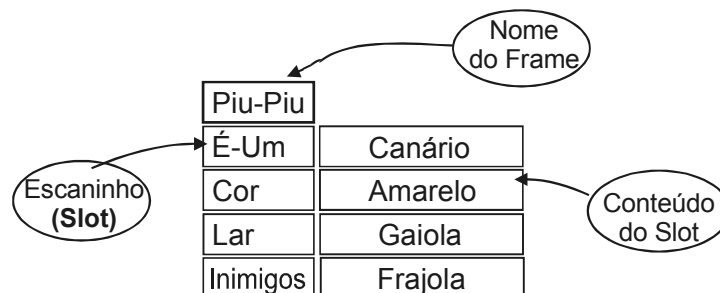
Entretanto, no uso mais corriqueiro dos símbolos, tem-se como finalidade principal a comunicação entre dois ou mais agentes. Para que isto ocorra a contento, é necessário que *ambos* tenham capacidade similar de associar os símbolos aos seus objetos respectivos (e que sejam os mesmos objetos). Portanto, para a finalidade de compartilhamento de informação, símbolos precisam ser objeto de convenção pública.

## Técnicas para Representação

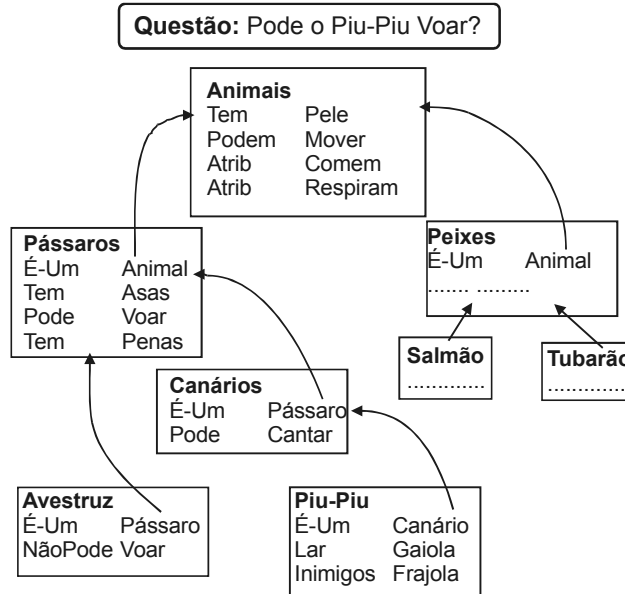
Agora que já vimos essa introdução sobre símbolos, ficará mais fácil entender as diversas características dos sistemas representacionais. Vale recordar que todos esses métodos que veremos (que, diga-se novamente, não é uma lista exaustiva) possuem como limitação a incapacidade de representar informações sub-simbólicas. Disso deve-se concluir que existe uma significativa parcela do conhecimento empresarial que não tem como ser explicitado e, portanto, não pode ser capturado por sistemas processadores de informação. A maior parte desse conhecimento sub-simbólico está “armazenado” nos cérebros das pessoas que compõe uma organização.

## Introduzindo os Frames

Introduzidos por Marvin Minsky (1975), os frames são estruturas compostas de “gavetas” (slots ou escaninhos) capazes de conter um valor. Um grupo associado de gavetas forma um frame. O conteúdo de um slot pode estar associado a instâncias hierarquicamente superiores. Através desse modelo, temos como implementar conceitos como herança, tão importante para as modernas linguagens orientadas a objeto. No exemplo abaixo temos um frame que descreve o “Piu-Piu”, aquele personagem de desenho animado:

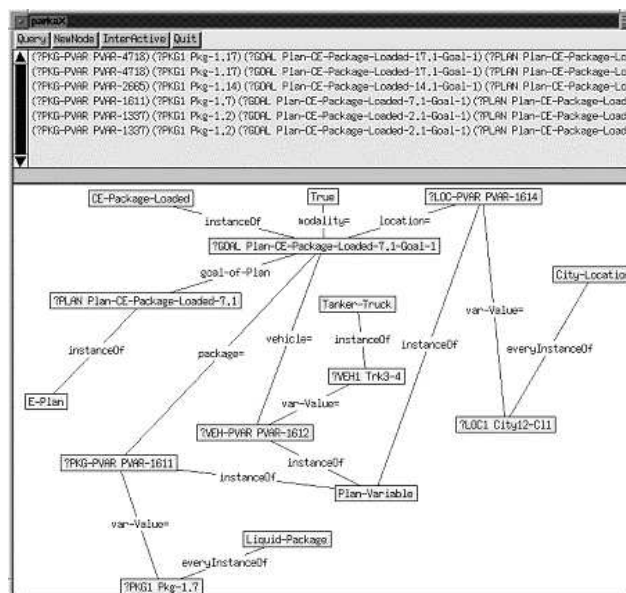


Estruturas de frames estão associadas montando complexas relações entre cada conceito. Isto permite que os sistemas baseados em frame possam responder a certas perguntas. No exemplo abaixo, perguntamos ao sistema se o Piu-Piu pode voar:



Começando com o frame inferior direito, observamos que Piu-Piu 'É-UM' canário. Isto nos leva ao frame que define o que é um canário. Sabemos que canários podem cantar e que canários são pássaros. Daqui, novamente subimos um nível até encontrar o frame Pássaros. Inspecionando esse frame determinamos que pássaros podem voar. Disso vem a resposta que Piu-Piu pode voar.

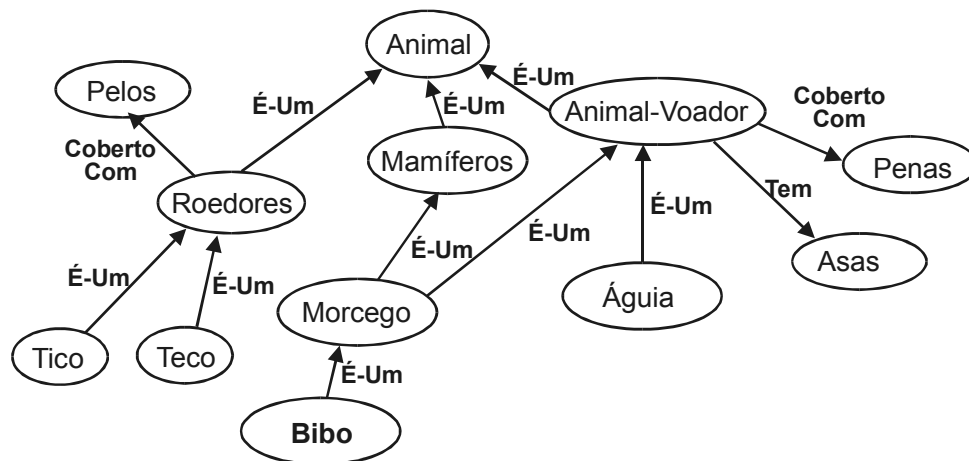
Frames podem ser usados em grande número, e com o uso de sistemas especificamente desenhados, pode-se obter tempos de processamento mínimos mesmo para consultas sofisticadas. O exemplo abaixo apresenta a interface de um sistema baseado em frames (sistema PARKA):



A implementação do PARKA em uma CM-5 (Connection Machine, computador paralelo de alto desempenho, que atualmente já está um pouco obsoleto) obteve tempos de processamento de poucas centenas de milissegundos para consultas envolvendo centenas de milhares de frames.

## Redes Semânticas

As Redes Semânticas (Semantic Networks) são construções que representam conhecimento através de *nós* conectados por *arcos*. As origens das redes semânticas são bastante antigas, tendo sido exploradas pela Filosofia, Psicologia e Linguística. Um exemplo típico segue:



As redes semânticas também podem “responder a perguntas”: Será que Bibo tem asas? Siga o diagrama, começando pela parte inferior, observando o nó chamado Bibo. Dele concluímos que Bibo É-UM morcego e prosseguindo, verificamos que morcego É-UM animal-voador. Logo em seguida verificamos que animal-voador Tem asas e assim concluímos que Bibo tem asas. Essa estrutura parece bastante similar à dos frames. Mas a similaridade termina quando extensões das redes semânticas entram em ação.

Uma dessas extensões são as chamadas redes semânticas proposicionais, onde os nós representam proposições. Os nós podem conter redes aninhadas que especificam mais informações sobre as proposições. Outras extensões são feitas quando os nós representam conceitos ou relações entre conceitos, em uma estrutura conhecida como Grafos Conceituais (Conceptual Graphs, cujo principal teórico é John Sowa<sup>5</sup>). A capacidade expressiva dessas construções é, em muitos casos, superior à conseguida pela lógica de primeira ordem (veja adiante).

O sistema SNePS (Semantic Network Processing System) é um exemplo de implementação de uma rede semântica proposicional, tendo sido desenvolvido por Stuart Shapiro e vários colegas desde a década de 1970.

## Regras de Produção

A febre dos Sistemas Especialistas ocorreu na década de 80. Muitos achavam que estávamos diante de sistemas realmente inteligentes. A principal estrutura representacional usada por esses sistemas foram as Regras de Produção. Nelas, procurava-se representar (explicitar) conhecimentos sobre um domínio. As regras dispunham de antecedentes (condições iniciais) e conseqüentes (condições finais).

O sistema funcionava adicionando em uma memória de trabalho os dados iniciais fornecidos pelo usuário (uma pergunta, por exemplo). Desses dados, o sistema buscava encontrar as regras que poderiam ser “disparadas”, ou seja, que tivessem suas condições iniciais satisfeitas por causa do conteúdo dessa memória de trabalho. Sempre que uma regra disparava, a memória de trabalho recebia a consequência correspondente, o que fazia o sistema entrar em um novo ciclo de busca por condições iniciais satisfeitas, até que uma resposta final emergisse. A seguir temos um exemplo de codificação de regras de produção:

Conhecimento Explícito	
<b>Se</b> Estamos no entardecer E o céu está avermelhado <b>Então</b> Tempo amanhã deve ser bom.	<b>Se</b> temperaturaForno > 800 E tempoOperacao < 15 <b>Então</b> FecharVálvula V1 E ApresentarAviso A1.

Um dos exemplos mais clássicos foi o sistema MYCIN (1976), especializado no diagnóstico de doenças infecciosas. Este é o aspecto geral de uma regra do Mycin:

### Regras

<pre> (&lt;predicado&gt; &lt;objeto&gt; &lt;atributo&gt; &lt;valor&gt;) (  membf      cntxt      site      sterilestites  )           </pre>
--

Abaixo temos um exemplo de codificação de uma regra:

```

((and
  (same cntxt infect primary-bacteremia)
  (membf cntxt site sterilestites)
  (same cntxt portal gi) )
 (conclude cntxt ident bacteroides tally 0.7))
  
```

Vastas seqüências dessas regras compõem a “base de conhecimento” do sistema, algo que, para ser criado, requer o esforço de diversos especialistas.

### Representando Conhecimento com Lógica

Uma das mais tradicionais formas de representar conhecimento tem origem no ramo da Matemática que investiga a Lógica. Um primeiro formato muito usado é chamado de Lógica Proposicional (ou Cálculo Proposicional). Nele, usam-se proposições e relações lógicas entre proposições. As proposições podem ser verdadeiras ou falsas, assim como as expressões montadas com proposições e relações lógicas.

- ∧ Conector E
- ∨ Conector OU
- ⇒ Implicação
- ↔ Equivalência
- ¬ Negação

A chamada semântica do Cálculo Proposicional define de que forma os valores de validade (verdadeiro ou falso) são obtidos a partir das diversas possibilidades de relação entre as proposições e os conectores lógicos:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \wedge \neg Q$
falso	falso	verdad.	falso	falso	falso
falso	verdad.	verdad.	falso	verdad.	falso
verdad.	falso	falso	falso	verdad.	verdad.
verdad.	verdad.	falso	verdad.	verdad.	falso

Contudo, o Cálculo Proposicional é menos expressivo do que a conhecida *Lógica de Primeira Ordem* (FOL - First-Order Logic, ou Predicate Calculus). Neste caso, lida-se com uma estrutura de representação mais rica, que inclui elementos como:

- Objetos**           Pessoas, veículos, empresas, épocas, etc.
- Relações**        Maior que, dentro de, irmão de, etc.
- Propriedades**   Azul, áspero, redondo, horizontal, etc.
- Funções**         Mais, menos, seguinte, terceiro, etc.
- Variáveis**       ?x, ?y, ?z, etc.

A partir desses elementos, a FOL pode representar sentenças como “O rei malvado John governou a Inglaterra em 1200”. Essa expressividade garante grande parte do benefício dessas estruturas. Além disso, há formas de especificar quantificadores, que delimitam (ou ampliam) o escopo das expressões lógicas. Dois dos principais quantificadores são o Existencial e o Universal:

$\forall$    Quantificador Universal

$$\forall x \text{ Gato}(x) \Rightarrow \text{Mamífero}(x)$$

A expressão acima deve ser lida assim: para qualquer X tal que X seja um gato, isto implica que esse X é um mamífero. Este é um exemplo do quantificador Existencial:

$\exists$    Quantificador Existencial

$$\exists x \text{ Governador}(x)$$

É assim que lemos a expressão acima: Existe um X tal que X é Governador. Isto nos permite montar expressões que signifiquem frases completas:

“Existe alguém que é amado por todo mundo”

$$\exists y \forall x \text{ Ama}(x, y)$$

“Todo mundo ama alguma pessoa”

$$\forall x \exists y \text{ Ama}(x, y)$$

Mas é no concernente às inferências que a FOL mostra seu poder. O exemplo abaixo mostra como isso acontece:

$\forall x \text{ advogado}(x) \Rightarrow \text{rico}(x)$   
 Todo Advogado é Rico

$\forall x \text{ temCasa}(x) \wedge \text{rico}(x) \Rightarrow \text{casaGrande}(x)$   
 Todo aquele que é rico e tem casa,  
 tem casa grande

$\forall x \text{ casaGrande}(x) \Rightarrow \text{custaManter}(x)$   
 Quem tem casa grande, gasta  
 muito para mantê-la

As expressões acima (com suas “traduções” em Português logo abaixo) fazem parte da base de conhecimentos de um sistema. A partir de alguns dados entrados pelo usuário (por exemplo, João é um advogado e João tem casa), o sistema rapidamente concluirá:

$\text{advogado}(\text{João})$ $\text{temCasa}(\text{João})$	$\Rightarrow$	João gasta muito com sua casa
---	---------------	----------------------------------

Este exemplo pode parecer até mesmo trivial, mas imagine o poder desta estratégia quando se dispõe de milhares de expressões lógicas coerentes e encadeadas.

## Compartilhando Conhecimento

O processamento automático de conhecimento (via inferência lógica, por exemplo) tem sido considerado um grande avanço, mas mesmo assim existem diversos problemas teóricos ainda sem solução definitiva<sup>6</sup>. Contudo, além de inferência, representar conhecimento para compartilhá-lo tem sido um sonho antigo. Algumas formas já foram tentadas, como o KIF:

```

KIF Knowledge Interchange Format

(Define-Ontology Example (Slot-Constraint-Sugar topLevelOntology))
(Define-Class Age (?X) :Def (Property ?X))
(Define-Relation Cousin(?s ?p) "Relation type Cousin"
:Def (And (Person ?s) (Person ?p)))

(Exists ((?j Person)
(And (Name ?j John)
(Believer ?j '(Exists ((?m Person) (?p Person) (?a Age))
(And (Name ?m Mary) (Chrc ?m ?a)
(Cousin ?m ?p) (Chrc ?p ?a)
)))
)))
  
```

Mais informações sobre essa metodologia podem ser vistas em <http://logic.stanford.edu/kif/dpans.html>. Outro conhecido estratagemas é o KQML:



## KQML Knowledge Query and Manipulation Language

```
tell(can-ask("O Estado de S. Paulo", 'city(x, "São Paulo)'))
ask-if(city(cid2384, "Campinas"))
ask-one(x, smoking(x, "Allowed"))
ask-all(x, city(x, "Ribeirão Preto") and pets(x, "Allowed"))
```

Mais informações em <http://www.cs.umbc.edu/kqml/>. Contudo, nada teve o impacto no compartilhamento de conhecimento similar àquilo que é conhecido como Ontologias.

## Introduzindo as Ontologias

Ontologias são formas de conceituar um domínio usando termos que disponham de uma hierarquia e que possam ser facilmente compartilhados. Tom Gruber define ontologias como sendo “especificações de uma conceitualização”. Em geral, ontologias preocupam-se em definir as entidades que ocupam a mente de um agente inteligente (um indivíduo, uma empresa, uma nação, etc.) e as relações que existem entre essas entidades.

Ontologias são, portanto, espécies de taxonomias que procuram descrever as entidades (conceitos, atributos, eventos, categorias, etc.) que constituem um domínio do conhecimento. Em alguns casos, esses domínios incluem noções abrangentes, mas na maioria das vezes a principal preocupação é definir e especificar domínios específicos, pertencentes ao ramo de atuação de quem está interessado na ontologia.

Como um dos principais objetivos das ontologias é servir de “base estável” para a troca de conhecimentos, muitos teóricos dizem que as ontologias devem se preocupar com aqueles detalhes de uma base de conhecimentos que não mudam durante a resolução de um problema. Essa base fixa é a principal força da ontologia para permitir a troca de informação entre agentes inteligentes. Ontologias só fazem sentido se estiverem sendo propostas para comunidades de usuários, sendo portanto um “acordo de cavalheiros”. O diagrama a seguir mostra que tipo de informação pode ser explicitada em uma ontologia:

- ▶ **Categorias**  
Laranja -> Frutas -> Vegetais -> Vida Orgânica
- ▶ **Medidas**  
Massa, Idade, Preço, Comprimento
- ▶ **Objetos Compostos**  
Carros, Prédios, Partida de Futebol
- ▶ **Tempo, Espaço, Movimento (Mudança)**  
Século XX, Cidade de São Paulo, Viagem à Bahia
- ▶ **Eventos e Processos**  
[aquisição laranjas], [espremer laranjas]
- ▶ **Objetos Físicos**  
Elefante, Telefone, Pneu
- ▶ **Substâncias**  
Suco de laranja, Manteiga, Cimento
- ▶ **Objetos Mentais e Crenças**  
[desejo de comprar laranja],  
[Louis Lane acredita que Super-Homem sabe voar]

## Por Que Precisamos de Ontologias?

Uma das principais razões, como já vimos, é permitir o compartilhamento de entidades entre usuários. Mas existe uma outra forte razão: não é possível representar o universo com níveis arbitrários de detalhes. Sempre que precisamos discorrer sobre algum domínio (por exemplo, os conceitos associados a uma instalação industrial), devemos restringir essa representação a partes específicas do universo (aquilo que estamos chamando de domínios), com detalhes suficientes para permitir a resolução de problemas relativos a um objetivo ou tarefa específicos.

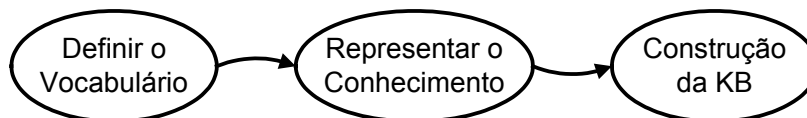
Isto envolve elaborar uma Conceitualização, ou seja, montar um conjunto abstrato de objetos e outras entidades que são assumidas como existentes nesse domínio específico, além de relações e funções sobre esses elementos. A escolha de um conjunto como esse envolve um compromisso ontológico (*ontological commitment*), que deve ser aceito por todos os usuários.

## Comparando Ontologias com Banco de Dados

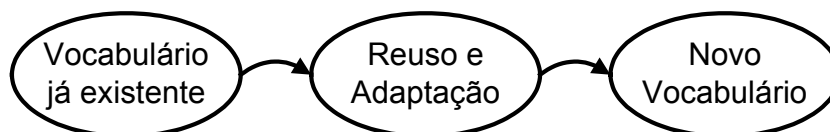
Pode parecer, à primeira vista, que Ontologias são desnecessárias, visto termos hoje em dia uma grande variedade de bancos de dados. Contudo, há importantes diferenças entre esses dois esquemas. Nos banco de dados, as informações estão normalmente estruturadas em tabelas e registros. Portanto, esse é um formato que *mistura níveis distintos*, pois em um mesmo registro podemos encontrar informações sobre pessoas, salários, cargos, departamentos, idade, etc. Todas essas informações pertencem a níveis muito diferentes. Em um repositório que usa Ontologias, as informações estão agrupadas dentro de uma hierarquia, o que permite estabelecer uma espécie de categoria para cada item. Assim, haveria um item “pessoa” e, abaixo deste, itens sobre “salário”, “idade”, “tempo de serviço”, etc. Acima do item pessoa encontraríamos departamento, chefe, unidade, projetos de que participa, etc. Observe, contudo, que toda essa descrição representa aspectos comuns do dia-a-dia, justamente onde uma ontologia não tem tanto valor. Imagine a vantagem de usar essa hierarquia quando temos noções técnicas complexas ou conceitos muito específicos, cujos nomes e relações precisam ser compartilhados entre muitas pessoas.

## Como Montar Ontologias?

Estes são os passos típicos para providenciar a elaboração de uma ontologia:



Em muitos casos, a ontologia pode iniciar a partir de um modelo já existente (um modelo antigo próprio ou de outra empresa). Isto envolve esta seqüência de passos:



De acordo com Guarino, estes são alguns dos critérios que devem ser usados para montar ontologias:

### **Particionamento**

- O que é considerado como parte de uma entidade?
- Quais as propriedades das partes?
- Quais os tipos de partes?

### **Integridade**

- O que é considerado como um Inteiro?
- Em que sentido suas partes estão conectadas?

### **Identidade**

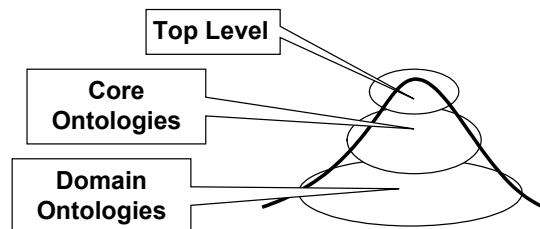
- Como a entidade pode mudar mantendo a identidade?
- Quais são suas propriedades essenciais?
- Em que circunstâncias a entidade perde a identidade?
- Mudança de partes/ponto de vista alteram a identidade?

### **Dependência**

- Entidade pode existir sozinha?
- Depende de outras entidades?

### **Níveis em Ontologias**

Há três níveis em ontologias elaboradas, conforme mostra o diagrama a seguir:

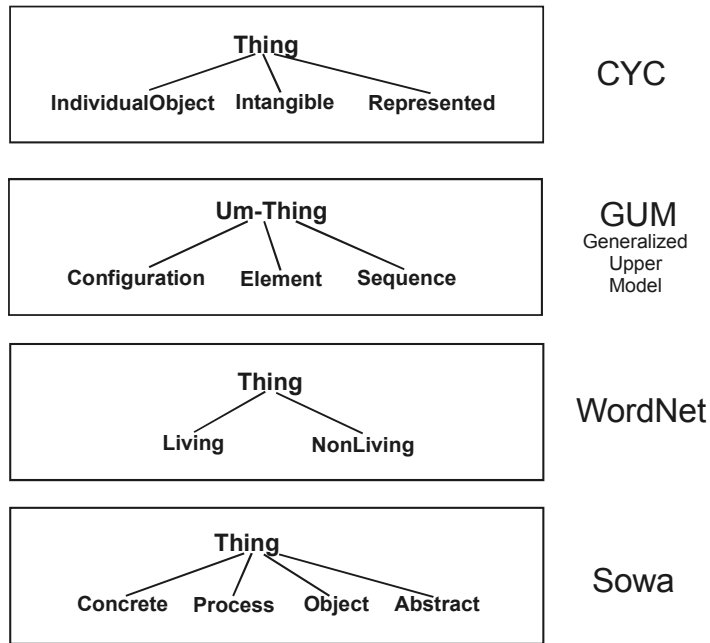


O nível mais superior (Top Level) contém entidades e relações que fazem parte do “conhecimento de senso comum”. São usualmente noções que estão nas mentes das pessoas, não importando se especialistas ou leigos. Algumas dessas relações e conceitos podem ser do conhecimento de crianças. Há vários modelos de ontologias disponíveis atualmente (um exemplo é o CYC) que disponibilizam esse top level.

O segundo nível (Core Ontologies) já reflete um tipo de conhecimento mais abstrato e elaborado. Em geral, esse nível lida com conceituações pertencentes a uma educação formal, onde são estabelecidas relações mais complexas entre entidades.

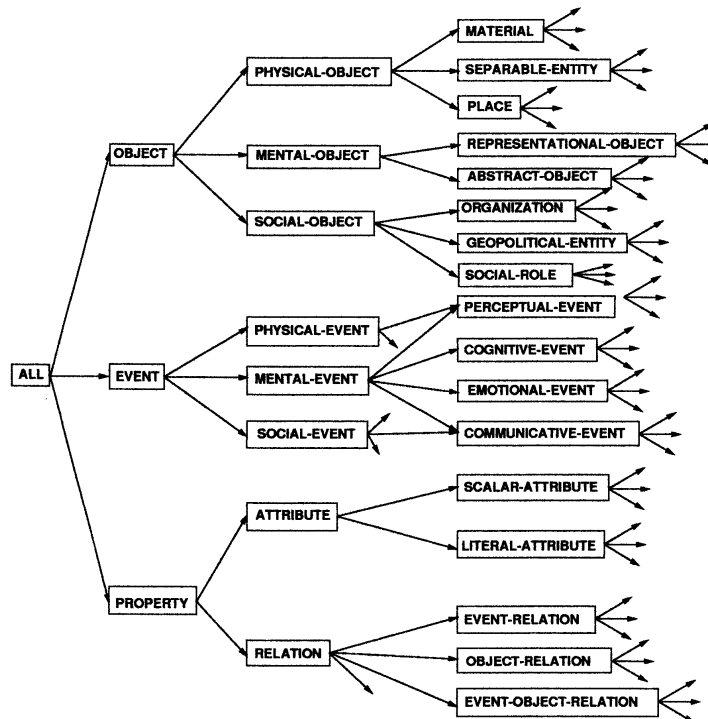
O último nível é onde a atividade principal dos engenheiros do conhecimento reside: no nível de domínio são explicitados os conceitos e relações que podem ser específicos da empresa. Este é o nível que maior utilidade tem, não só em relação à lógica de representação, mas principalmente no efeito de compartilhamento que permite.

O diagrama a seguir mostra apenas o primeiro nível do “top level” de quatro ontologias típicas:

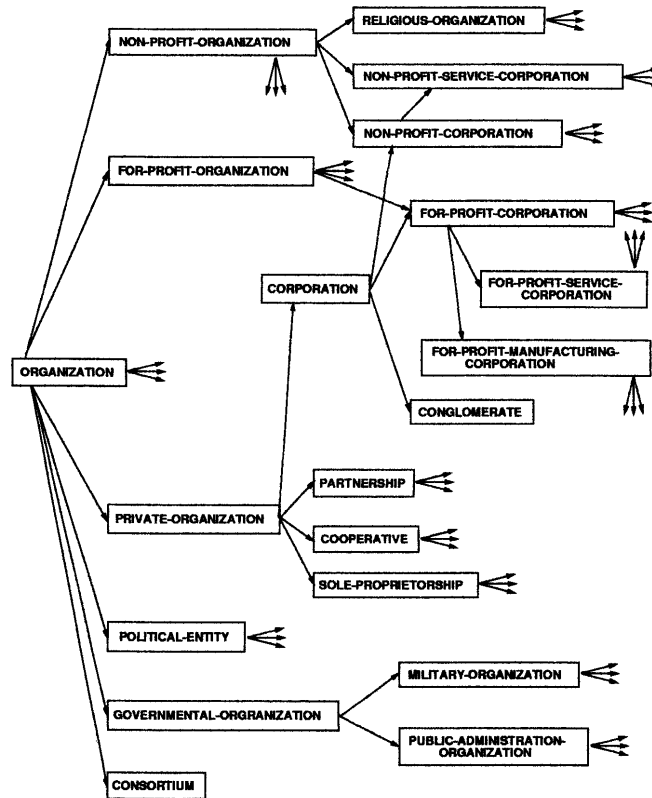


### Um Exemplo de Ontologia: Mikrokosmos

O projeto Mikrokosmos (Computing Research Laboratory, New Mexico State University) preocupou-se em montar uma ontologia que suportasse sistemas automatizados de tradução. Uma das finalidades básicas da Mikrokosmos foi ocupar-se em processar textos sobre fusões e aquisições de empresas. Esses textos vêm, via de regra, em formatos não padronizados. Além disso, as informações abrangem um universo muito amplo, o que requer a presença, na ontologia, de um vasto número de conceitos associados. Abaixo é apresentado o primeiro nível do Top Level da Mikrokosmos:



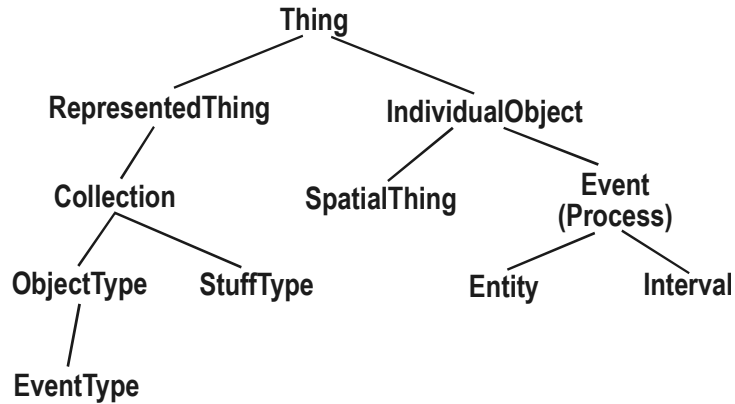
A partir no nó “ALL” temos as divisões fundamentais: “object”, “event” e “property”. Vamos seguir o nó “object” e depois o nó “social object”. Encontramos o nó “organization”, aberto abaixo:



A Mikrokosmos usa a representação de frames para cada nó. Os frames têm “slots” e valores associados, tal qual a definição de frames que já vimos. Através dos slots há a herança de propriedades de outros níveis da ontologia, através do uso de conexões chamadas de “relações” e também através de valores específicos (os “atributos”). Com o uso dessa formalização, a Mikrokosmos consegue ser eficaz na seleção de sentidos preferenciais das palavras, o que auxilia na resolução de ambigüidades. Isto também permite que sejam feitas inferências, através do mesmo processo que vimos quando falamos dos frames. Até mesmo a resolução de metáforas pode ser auxiliada por este processo, pois através da verificação do menor caminho entre dois conceitos consegue-se reduzir as possibilidades de confusão.

## Ontologia do CYC

O projeto CYC ([www.cyc.com](http://www.cyc.com)) é uma das mais ambiciosas tentativas de formalização do senso comum (common sense) em desenvolvimento. Em um outro artigo (veja Navega 2002) elaboro mais detalhes sobre o projeto. O primeiro nível da ontologia do CYC começa assim:



Os elementos da ontologia são representados através de clausulas da linguagem CycL. Eis alguns exemplos:

```

;;; #AnimalSound
(#$isa #AnimalSound #TemporalStuffType)
(#$genls #AnimalSound #AudibleSound)
(#$genls #AnimalSound #InformationBearingWavePropagation)
(#$comment #AnimalSound "A collection of information...")

;;; #AnimalWalkingProcess
(#$isa #AnimalWalkingProcess #ObjectType)
(#$isa #AnimalWalkingProcess #TemporalStuffType)
(#$isa #AnimalWalkingProcess #DefaultDisjointScriptType)
(#$genls #AnimalWalkingProcess #LocomotionProcess)
(#$genls #AnimalWalkingProcess #SimpleWholeBodyMovement)
(#$comment #AnimalWalkingProcess "The collection of individual...")

;;; #AquaticOrganism
(#$isa #AquaticOrganism #OrganismTypeByHabitat)
(#$genls #AquaticOrganism #Organism-Whole)
(#$comment #AquaticOrganism "The collection...")

;;; #Arm
(#$isa #Arm #SymmetricAnatomicalPartType)
(#$genls #Arm #Appendage-AnimalBodyPart)
(#$comment #Arm "The collection of all animal arms...")

;;; #ArmMovement
(#$isa #ArmMovement #TemporalStuffType)
(#$isa #ArmMovement #DefaultDisjointScriptType)
(#$genls #ArmMovement #BodyMovementEvent)
(#$comment #ArmMovement "The collection of any movements...")

```

O top level da ontologia do CYC está disponível em vários formatos. Em um deles, houve uma conversão para o formato MDB usado pelo MS Access:

<http://www.ontotext.com/downloads/CycMDB.html>

id	name	comment_short
16297	#\$AbandoningSomething	The collection of events in which some #Agent deliberately gives up possession of something, without giving it
16567	#\$Ablation	A collection of events. In an #Ablation, a (usually relatively thin) layer of material is removed from the surface
15259	#\$abnormal	Every default rule in our system $P(x_1, \dots, x_n) \Rightarrow Q(x_1, \dots, x_n)$ is implicitly treated as $(\text{not}(\text{abnormal}(x_1, \dots, x_n)))$ and
16478	#\$above-Directly	(#\$above-Directly ABOVE BELOW) means either (1) the volumetric center of ABOVE is directly above some pc
16731	#\$above-Generally	(#\$above-Higher OBJ1 OBJ2) means that OBJ1 is more or less above OBJ2. To be more precise: if OBJ1 would
14746	#\$AboveGroundLevelInAConstruction	This refers to first floor and up. We created this to make the distinction between basements and non-basement:
16404	#\$above-Higher	(#\$above-Higher ?OBJ-A ?OBJ-B) means that ?OBJ-A is "higher up" than ?OBJ-B. Since most contexts are te
15643	#\$above-Overhead	(#\$above-Overhead ABOVE BELOW) means that ABOVE is directly above BELOW, all points of ABOVE are h
14837	#\$above-Touching	(#\$above-Touching ABOVE BELOW) means that ABOVE is located over BELOW and they are touching. More
17146	#\$AbradingSomething	A collection of events. In an #AbradingSomething event, the surface of some object is gradually worn away by
15064	#\$AbsoluteValueFn	#\$AbsoluteValueFn is the unary mathematical function that returns the absolute value of its argument, e.g., (#\$
15076	#\$AbstractInformation	The collection of all instances of abstract information stored or transmitted in some manner, and representing (t
16132	#\$AbstractProgrammingLanguage	The collection of languages invented for use by computers. This includes both command languages and others
14927	#\$AbstractShape	A collection of objects; a subset of #GeometricThing. #AbstractShape is the collection of all abstract physic
15830	#\$AbstractShapeType	A collection of collections. Each element of #AbstractShapeType is a collection of things which are subsets c
16072	#\$AcademicProfessional	A collection of persons; a subset of #Professional. Each element of #AcademicProfessional is a person who
16546	#\$AcademicQuarter	Each instance of this collection is a #TimeInterval defined by some educational institution: one quarter of their
16106	#\$AcademicSemester	Each instance of this collection is a #TimeInterval defined by some educational institution: one half of their #
15816	#\$AcademicTrimester	Each instance of this collection is a #TimeInterval defined by some educational institution: one third of their #
14812	#\$AcademicYear	Each instance of this collection is an annually recurring #TimeInterval defined by an educational institution. Si
16668	#\$Acceleration	Acceleration is the change in speed of an object per unit time. It is a measurable physical quantity, measured
15943	#\$AccessingAnIBT	A collection of information transfer events. Each element of #AccessingAnIBT is an action by which an agent
15313	#\$Account	A collection of objects; a subset of #LegalAgreement. Each element of #Account is a recorded obligation (of
16294	#\$AccountAdministrator	The predicate #accountAdministrator identifies the agent who administers a particular account. (#\$accountAd
18018	#\$AccountBalance	The predicate #accountBalance is used to indicate the balance of a particular account. (#\$accountBalance A
15240	#\$AccountHolder	The predicate #accountHolder identifies the holder of a particular account. (#\$accountHolder ACCT AGENT) n
17299	#\$AccountStatus	The predicate #accountStatus indicates whether a particular account is paid up, overdue, delinquent, etc. (#\$
15359	#\$AccountStatusAttribute	A collection of attributes. Each element of #AccountStatusAttribute is an attribute that describes the obligat
16295	#\$AccountType	A collection of collections. Each element of #AccountType is a collection of financial accounts of some type.
17073	#\$AcknowledgeAnAct	#\$AcknowledgeAnAct ?ACKACT ?COMFACT means that the #CommunicationAct:Sinola ?COMFACT is acknow

O lugar ideal para se experimentar com o sistema CYC e sua ontologia é o site da versão para experimentação: <http://www.opencyc.org>

## Conclusões

Em que pese todas as possíveis restrições aos sistemas simbólicos (principalmente no que concerne a sistemas inteligentes capazes de aprender de forma autônoma), é preciso criar e trabalhar com representações simbólicas. Não se pode esquecer que o ser humano sempre tem que fazer parte desse circuito, pois atualmente somente ele poderá aplicar essas informações como conhecimento prático. As Ontologias representam um capítulo à parte dos sistemas simbólicos, com várias demonstrações práticas e múltiplos potenciais usos. Chegaremos ao dia em que nossos computadores estarão também capacitados a operar com conhecimento implícito, o que os fará potencialmente muito mais úteis do que são hoje.

## Referências

Barr, Avron; Feigenbaum, Edward A. (1982) The Handbook of Artificial Intelligence, Vol. II. Addison-Wesley Publishing, Inc. Massachusetts.

Barr, Avron; Cohen, Paul R.; Feigenbaum, Edward A. (1989) The Handbook of Artificial Intelligence, Vol. IV. Addison-Wesley Publishing, Inc. Massachusetts.

Barsalou, Lawrence W. (1999). Perceptual symbol systems. Behavioral and Brain Sciences, 22, 577-609.

Breuker, Joost; Valente, Andre; Winkels, Radboud (1997) Legal Ontologies: A Functional View. In LEGONT 97, Proceedings of the First International Workshop on Legal Ontologies, Melbourne, Australia

Davis, Ernest (1990) Representations of Commonsense Knowledge. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Ginsberg, Matt (1993) Essentials of Artificial Intelligence. Morgan Kaufmann Publishers, Inc.

- Guarino, Nicola (1994) The Ontological Level. In R. Casati, B. Smith and G. White (eds.), *Philosophy and the Cognitive Science*. Holder-Pivhler-Tempsky, Vienna, pp. 443-456
- Guarino, Nicola (1995) Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars (ed.) *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. IOS Press, Amsterdam: pp. 25-32
- Harnad, Stevan (1990) The Symbol Grounding Problem. *Physica D* 42: 335-346.
- Kettler, Brian; Andersen, William; Hendler, James; Luke, Sean (1995) Using the Parka Parallel Knowledge Representation System. University of Maryland technical report CS-TR-3485.
- Lenat, Douglas B.; Guha, R. V. (1990) *Building Large Knowledge-Based Systems*. Addison-Wesley Publishing Company, Inc., New York.
- Luger, George F.; Stubblefield, William A. (1993) *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*. Benjamin/Cummings Publishing Company, Inc., California.
- Mahesh, Kavi; Nirenburg, S.; Cowie, J.; Farwell, D. (1996) An Assessment of CYC for Natural Language Processing. Computing Research Laboratory, New Mexico State University
- Mahesh, Kavi; Nirenburg, Sergei (1995) A Situated Ontology for Practical NLP. In: *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal 1995.
- Markman, Arthur B. (1999) *Knowledge Representation*. Lawrence Erlbaum Associates, Inc., New Jersey.
- Mars, N. J. I. (editor) (1995) *Towards Very Large Knowledge Bases*. IOS Press
- Minsky, Marvin (1975) A Framework for Representing Knowledge. In: Winston, Patrick H. (ed) *The Psychology of Computer Vision*. McGraw-Hill, New York.
- Motta, Enrico; Reusable Components for Knowledge Modelling. Knowledge Media Institute Technical Report KMI-TR-54
- Navega, Sergio (2000) Inteligência Artificial, Educação de Crianças e o Cérebro Humano. *Leopoldianum, Revista de Estudos de Comunicações da Universidade de Santos* (Ano 25, No. 72, Fev. 2000, pp 87-102).
- Navega, Sergio (2002) Projeto CYC: Confundindo Inteligência com Conhecimento. *KMBrazil 2002*, 3o. Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento
- Navega, Sergio (2005) *Pensamento Crítico e Argumentação Sólida*. Publicações Inteliwise.
- Newell, Allen; Simon, Herbert (1976) *Computer Science as Empirical Enquiry: Symbols and Search*. In Haugeland, John (ed.) (1997) *Mind Design II*, MIT Press, Massachusetts.
- Newell, Allen (1982) The Knowledge Level. In Luger, George F. (1995) *Computation and Intelligence, Collected Readings*. AAAI Press
- Nilsson, Nils J. (1998) *Artificial Intelligence, A New Synthesis*. Morgan Kauffmann Publishers, Inc
- O'Hara, Tom; Mahesh, Kavi; Nirenburg, Sergei (1998) Lexical Acquisition with WordNet and the Mikrokosmos Ontology. In: *Content Visualization and Intermedia Representations (CVIR'98)*



Reed, Stephen L.; Lenat, Douglas (2003) Mapping Ontologies into Cyc. Cycorp, Inc., technical report

Riddle, Patricia (et. al) (1992) CYC Evaluation. Boeing Computer Services Technical Report.

Russell, Stuart, Norvig, Peter (1995) Artificial Intelligence, A Modern Approach, Prentice-Hall, Inc.

Shanahan, Murray (1997) Solving the Frame Problem. MIT Press, Massachusetts.

Shum, Simon B. (1997) Representing Hard-to-Formalize, Contextualized, Multidisciplinary, Organizational Knowledge, in AIKM Spring Symposium, Stanford University 1997

Sowa, John F. (ed.) (1991) Principles of Semantic Networks. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Stefik, Mark (1995) Introduction to Knowledge Systems. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

van Heijst, Gerardus A. C. M. (1995) The Role of Ontologies in Knowledge Engineering. Ph.D Thesis, Universiteit Van Amsterdam

## Notas

---

<sup>1</sup> Na realidade, pode-se imaginar um terceiro observador que contemple os sistemas que falamos, e esse processo vai sucessivamente ao infinito.

<sup>2</sup> Boa parte do fracasso de várias correntes da Inteligência Artificial deve-se ao “esquecimento” dessa questão perceptual.

<sup>3</sup> Este é um ponto filosoficamente um pouco mais complexo. Causalidade pode ser uma “atribuição” que fazemos do universo para que possamos desenvolver teorias e modelos com poder preditivo. A exploração deste assunto foge do escopo deste artigo.

<sup>4</sup> Este assunto é bastante vasto. Parte do tópico está explorado no livro Navega (2005).

<sup>5</sup> Os grafos conceituais guardam relação com os Grafos Existenciais (Existential Graphs) de Charles S. Peirce, um filósofo americano do início do século XX, também conhecido por ser o fundador do Pragmatismo, além do principal introdutor da Semiótica.

<sup>6</sup> Um desses problemas é conhecido como “Frame Problem”. A idéia básica é que mesmo em representações lógicas triviais do mundo há um número muito grande de interdependências óbvias que precisam ser manipuladas. Imagine, por exemplo, que representamos uma mesa com um livro sobre ela, uma mosca voando e um candelabro pendurado do teto. Para responder à pergunta “Se eu mover a mesa, o que também se moverá?”, um sistema lógico precisará consultar toda a sua base de conhecimentos para determinar quais as categorias de objetos que iriam se movimentar nesse caso. Sabemos que o livro, por estar sobre a mesa, irá também se mover, mas o mesmo não ocorrerá com a mosca nem com o candelabro. Embora simples, este exemplo mostra que qualquer item adicionado a essa situação irá aumentar catastróficamente o número de expressões lógicas que precisarão ser investigadas para determinar a “verdade” ou “falsidade” da situação. Com poucos milhares de expressões lógicas, mesmo sistemas sofisticados podem demorar muito tempo para responder mesmo a questões triviais. Este problema está explicado em detalhes em Shanahan (1997), que apresenta algumas possíveis soluções.