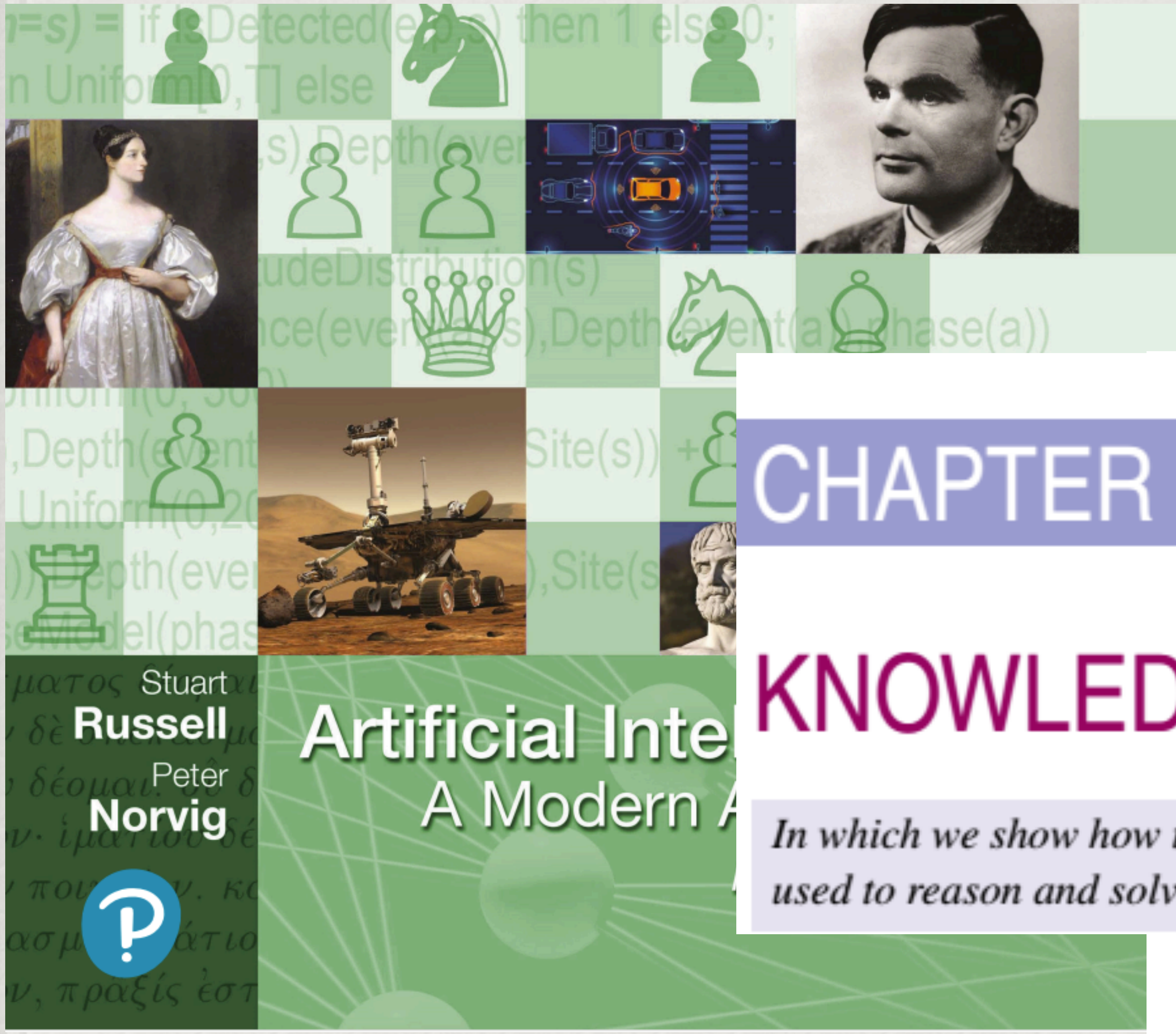
PMR-3510 Inteligência Artificial
Aula 10 - Representação de conhecimento e os sistemas especialistas

Prof. José Reinaldo Silva
reinaldo@usp.br

d-l@b



CHAPTER 10

KNOWLEDGE REPRESENTATION

In which we show how to represent diverse facts about the real world in a form that can be used to reason and solve problems.

will use first-order logic as the representation language, but later chapters will introduce different representation formalisms such as hierarchical task networks for reasoning about plans (Chapter 11), Bayesian networks for reasoning with uncertainty (Chapter 13), Markov models for reasoning over time (Chapter 16), and deep neural networks for reasoning about images, sounds, and other data (Chapter 22). But no matter what representation you use, the facts about the world still need to be handled, and this chapter gives you a feeling for the issues.

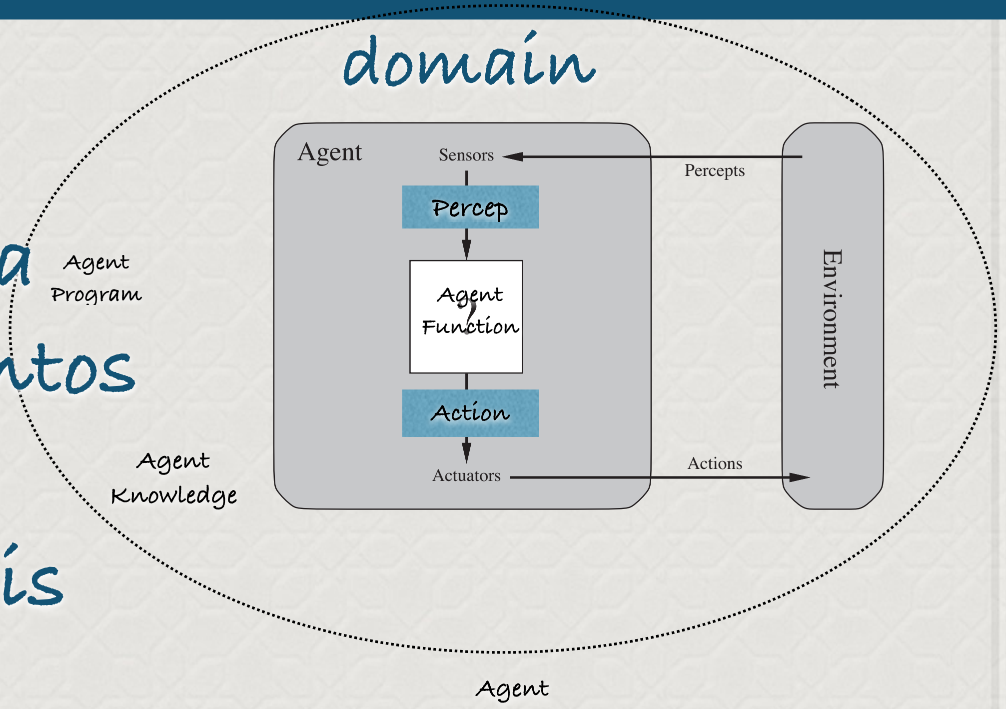
Section 10.1 introduces the idea of a general ontology, which organizes everything in the world into a hierarchy of categories. Section 10.2 covers the basic categories of objects, substances, and measures; Section 10.3 covers events; and Section 10.4 discusses knowledge about beliefs. We then return to consider the technology for reasoning with this content: Section 10.5 discusses reasoning systems designed for efficient inference with categories, and Section 10.6 discusses reasoning with default information.

10.1 Ontological Engineering

In "toy" domains, the choice of representation is not that important; many choices will work. Complex domains such as shopping on the Internet or driving a car in traffic require more

Complex domains

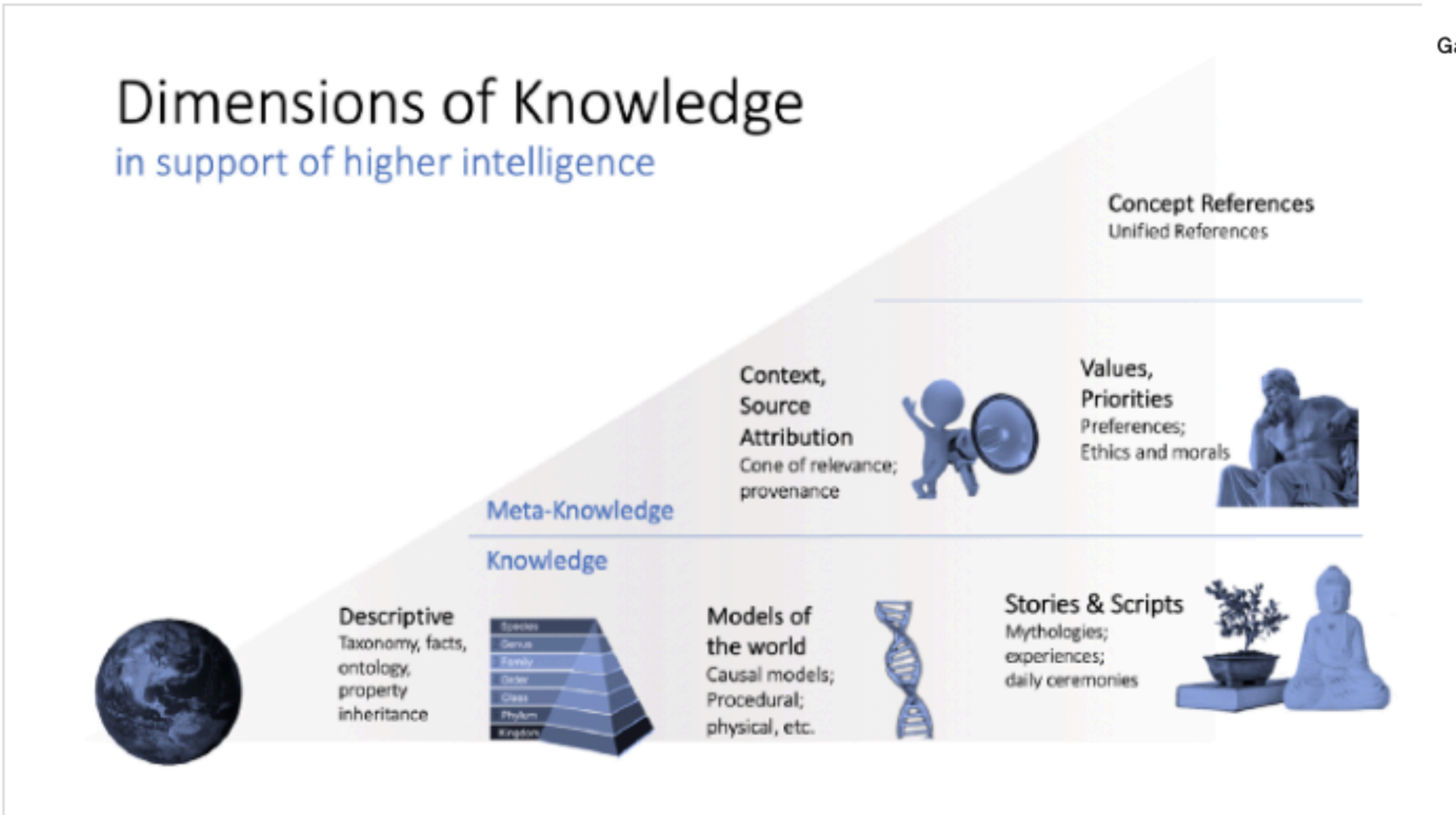
Domínios complexos estão associados a um environment aberto (novos elementos ou agentes podem aparecer ou desaparecer), à geração de percepts mais elaborados e a bases de conhecimento baseadas em " conhecimento profundo" (deep knowledge).



Domotics

https://towardsdatascience.com/understanding-of-and-by-deep-knowledge-aac5ede75169

Gadi Singer



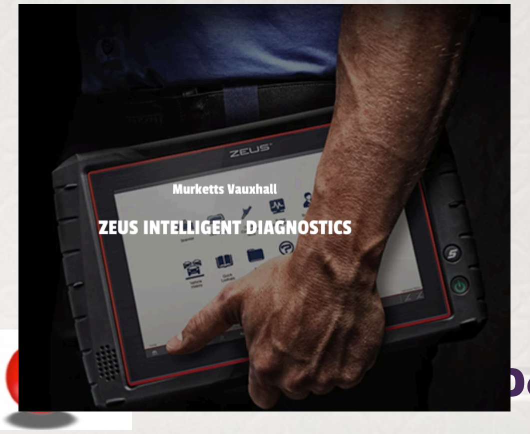
Dimensions of Knowledge in Support of Higher Intelligence. Credit: Gadi Singer/Intel Labs.

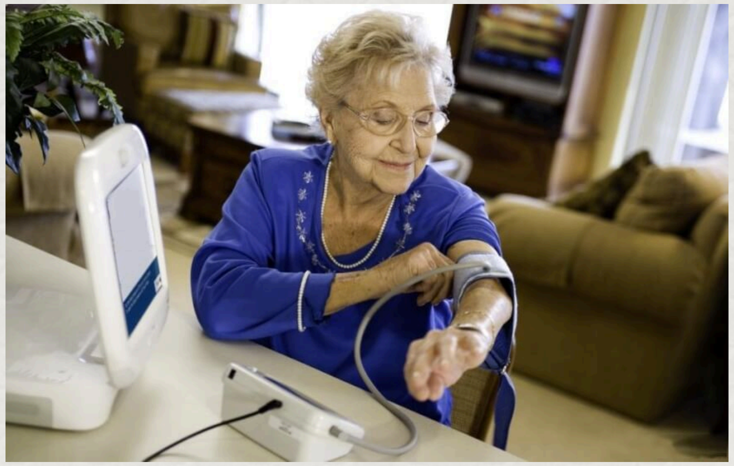
Exemplos

Telemedicina

Babylon

Internet Superhighway





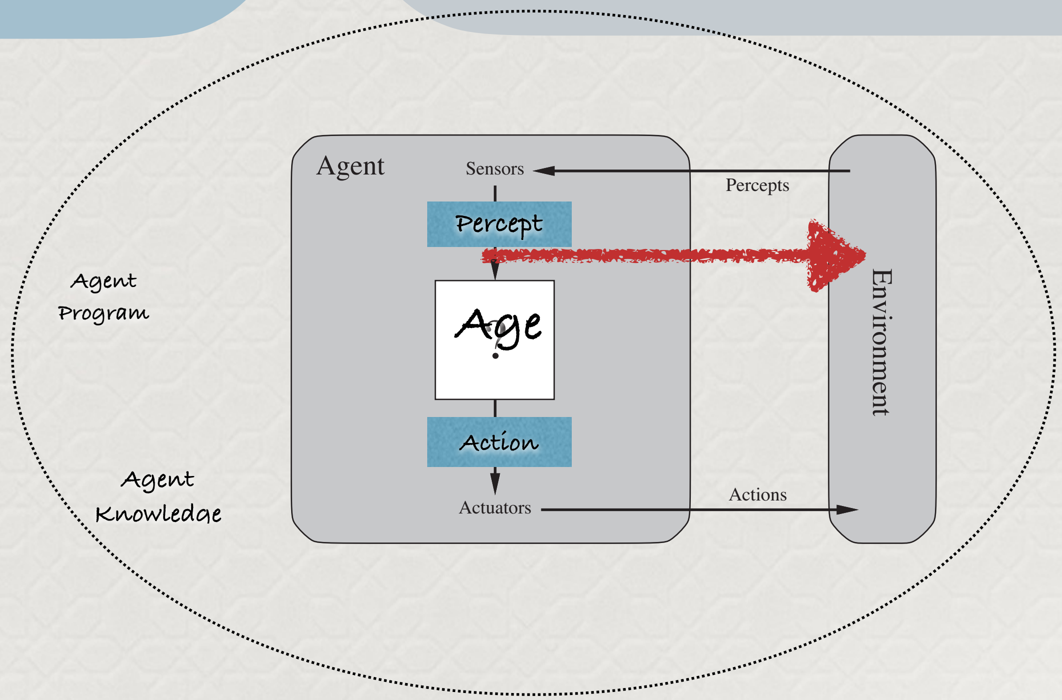
Furnas desenvolve um centro de excelência em IA para transformar processos já desenvolvidos (e novos) em um sistema RPA.

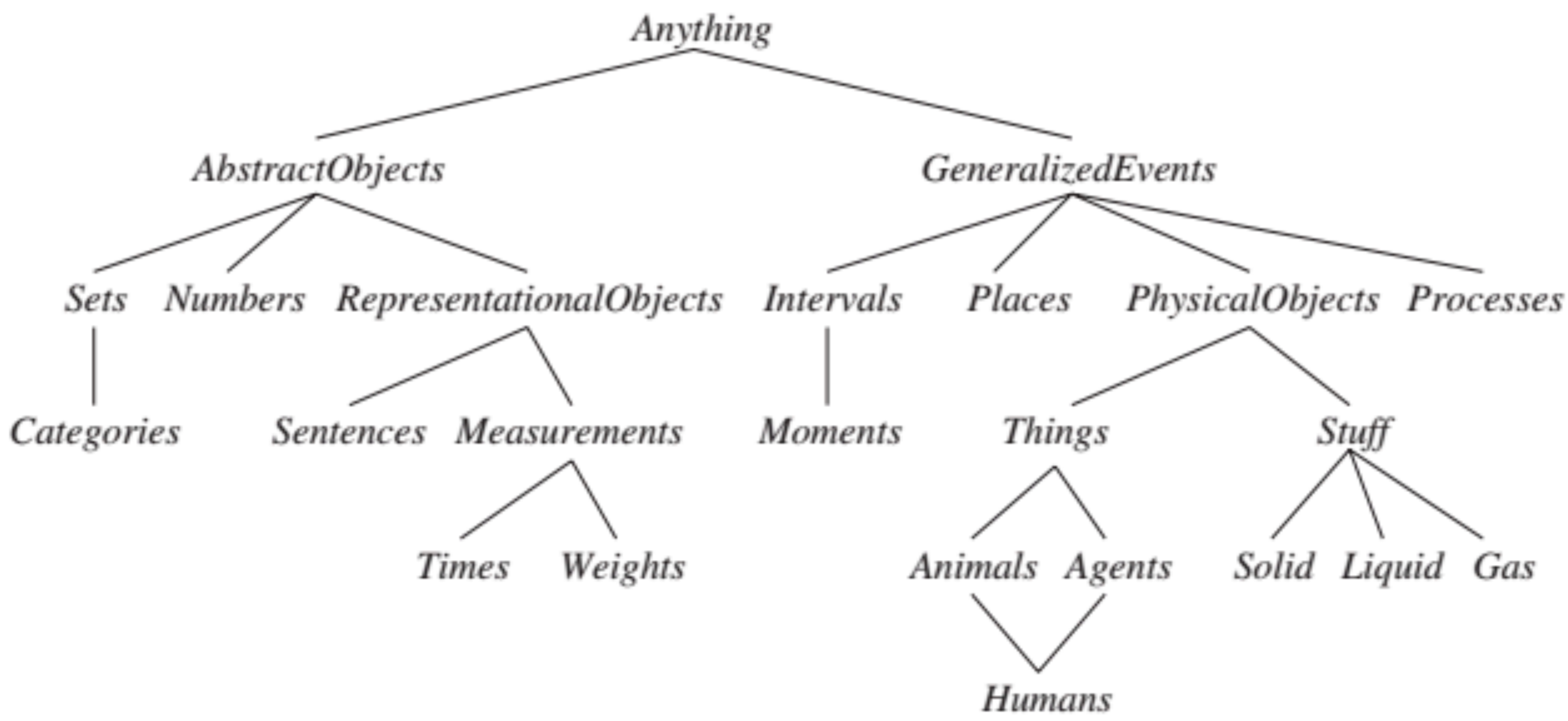
Aplicações Tradicionais da IA Clássica:

Jogos simples;
Sistemas de diagnóstico;
Sistemas instrucionais;
Sistemas de segurança;
Sistemas logísticos...

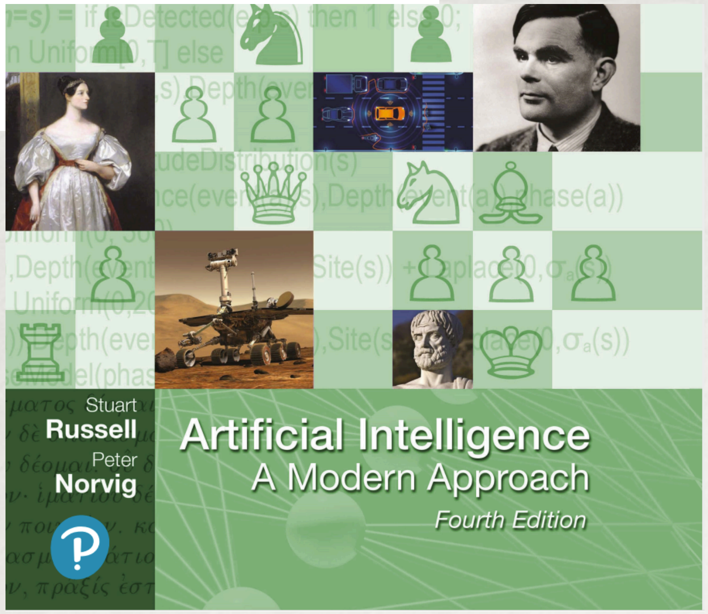
Aplicações Recentes:

"Serious games";
Sistemas de diagnóstico com interação NLP;
Sistemas assistivos (healthcare);
RPA...




Anything
AbstractObjects
GeneralizedEvents
Sets
Numbers
RepresentationalObjects
Intervals
Places
PhysicalObjects
Processes
Categories
Sentences
Measurements
Moments
Things
Stuff
Times
Weights
Animals
Agents
Solid
Liquid
Gas
Humans

- An object is a member of a category.
  $BB_9 \in Basketballs$
- A category is a subclass of another category.
  $Basketballs \subset Balls$
- All members of a category have some properties.
  $(x \in Basketballs) \Rightarrow Spherical(x)$
- Members of a category can be recognized by some properties.
  $Orange(x) \land Round(x) \land Diameter(x) = 9.5'' \land x \in Balls \Rightarrow x \in Basketballs$
- A category as a whole has some properties.
  $Dogs \in DomesticatedSpecies$



Ontologies

An **ontology** is a formal, explicit, shared specification of a conceptualization of a domain (Gruber, 1993).

Conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.

The term ontology is borrowed from Philosophy, where ontology is a systematic account of existence (what things exist, how they can be differentiated from each other etc.).

Today the word **ontology is a synonym for (shared!) knowledge base**.

Formal Languages for Ontologies

Ontologies are typically expressed in some **formal, logic-based language** e.g., FOL.

The literature also offers us special formalisms for defining ontologies that contain mainly taxonomic knowledge:

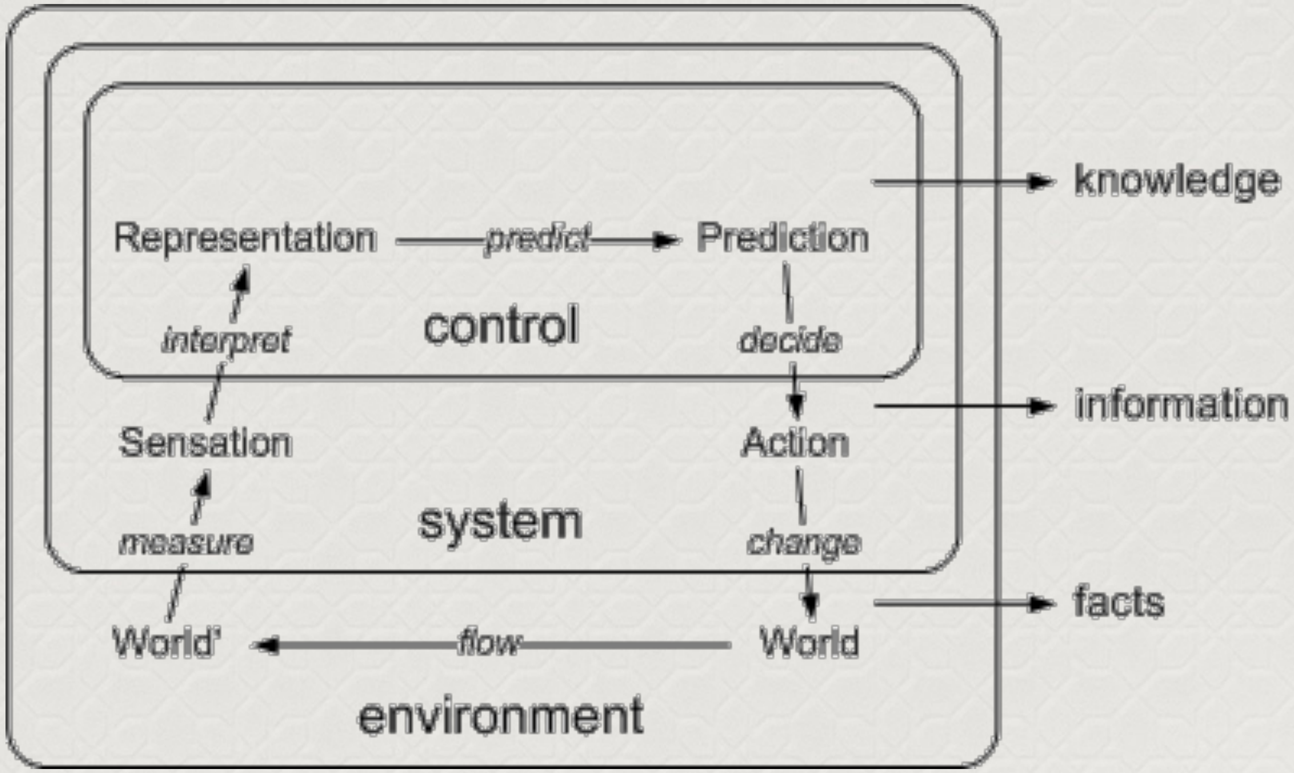
- Semantic networks
- Frames
- Description logics
- RDF, RDF(S) and OWL 2 for ontologies in the **Semantic Web**.

https://www.earley.com/insights/role-ontology-and-information-architecture-ai

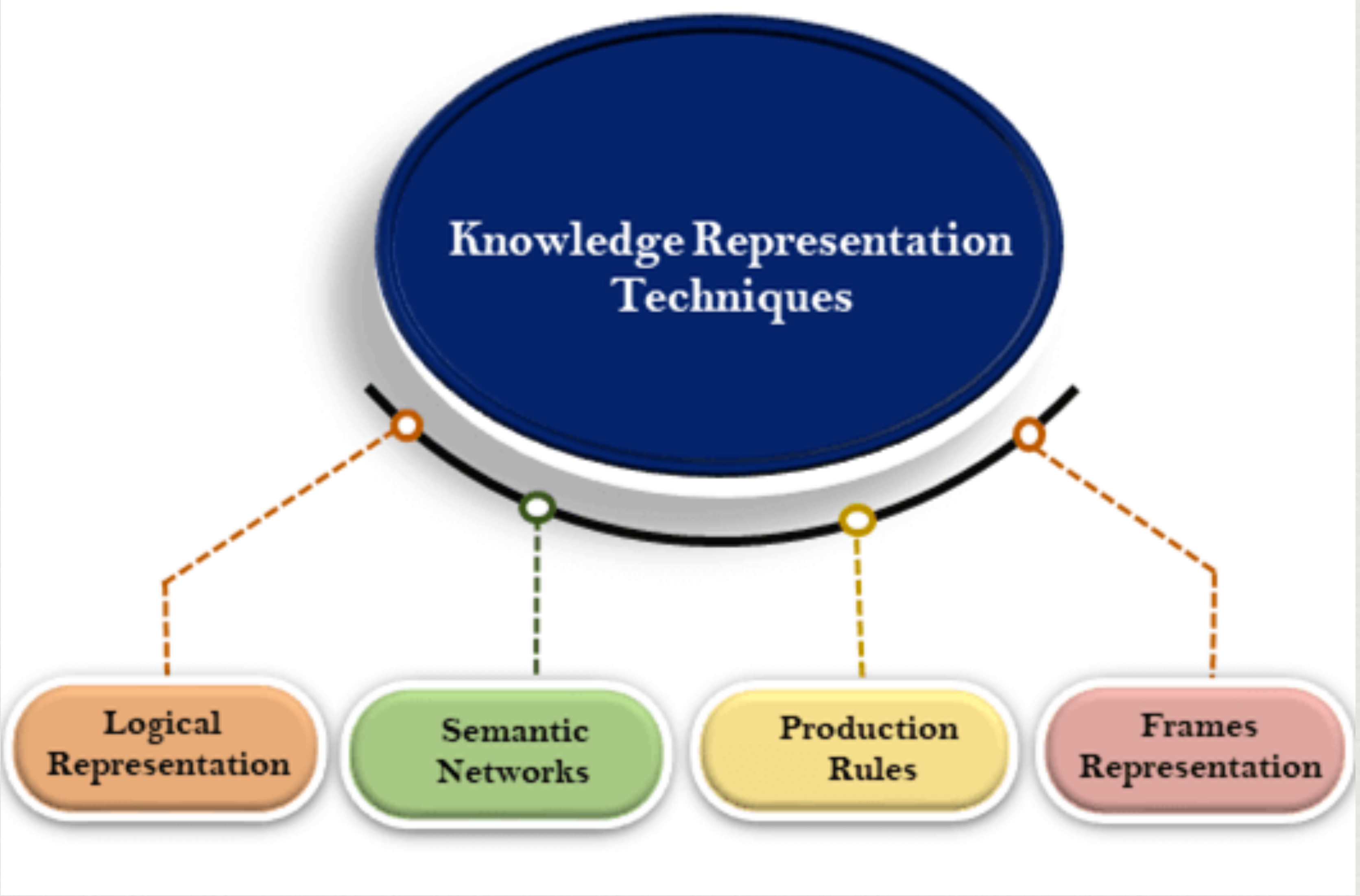
(Russell)

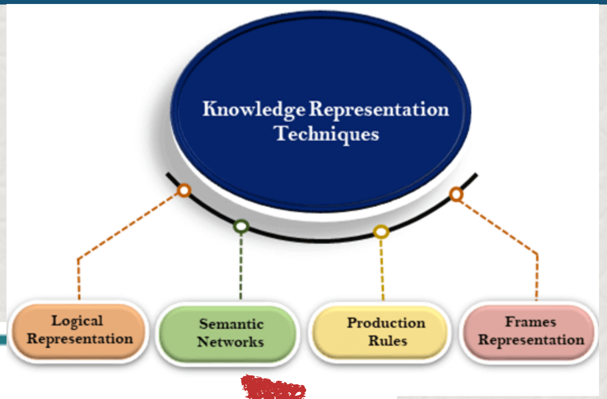
$$R \in R \Leftrightarrow R \in \{x \mid x \notin x\}$$
$$\Leftrightarrow R \notin R.$$

Nunca misture os níveis de abstração



Nos "serious games" que usam storyboard existem pelo menos dois níveis: o que regula as opções e o que executa. Se o jogo se adapta a usuários com diferentes habilidades pode existir um terceiro nível.


Knowledge Representation Techniques
Logical Representation
Semantic Networks
Production Rules
Frames Representation



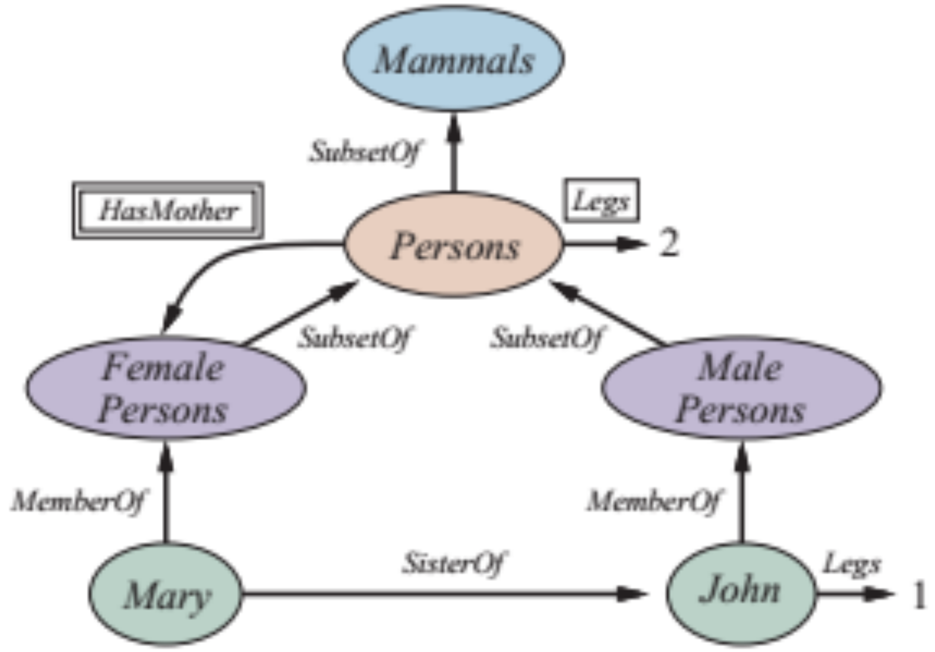


Figure 10.4 A semantic network with four objects (John, Mary, 1, and 2) and four categories. Relations are denoted by labeled links.

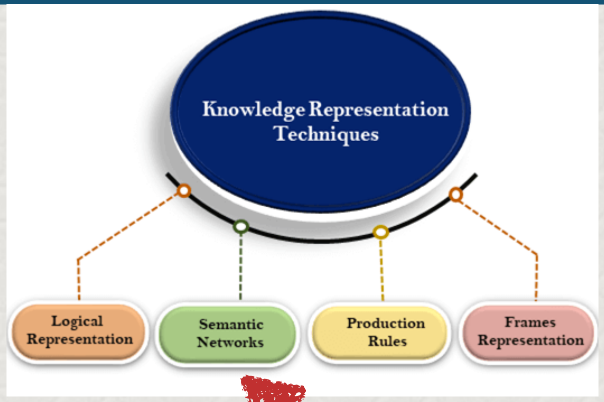
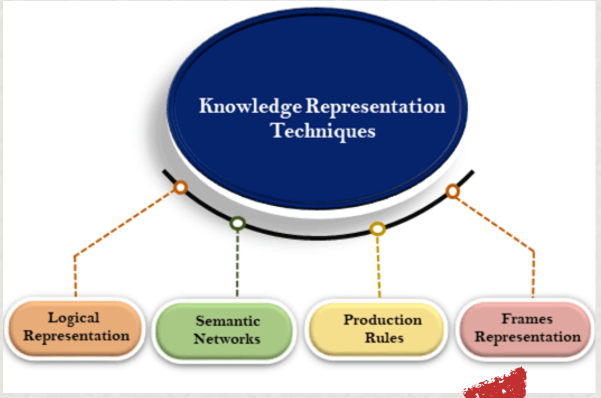
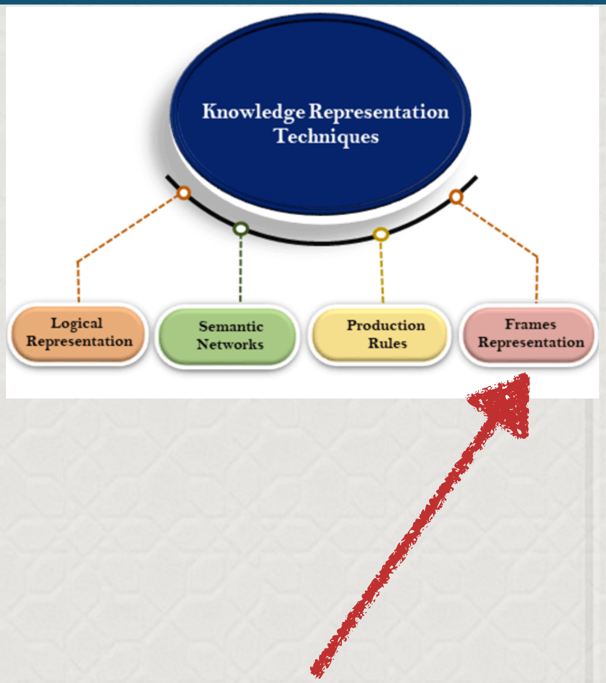


Figure 10.5 A fragment of a semantic network showing the representation of the logical assertion *Fly*(*Shankar*, *NewYork*, *NewDelhi*, *Yesterday*).



An influential paper by Marvin Minsky (1975) presented a version of semantic networks called **frames**; a frame was a representation of an object or category, with attributes and relations to other objects or categories.



A Framework for Representing Knowledge

Marvin Minsky

MIT-AI Laboratory Memo 306, June, 1974.

Here is the essence of the theory: When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a *Frame*. This is a remembered framework to be adapted to fit reality by changing details as necessary.

A *frame* is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

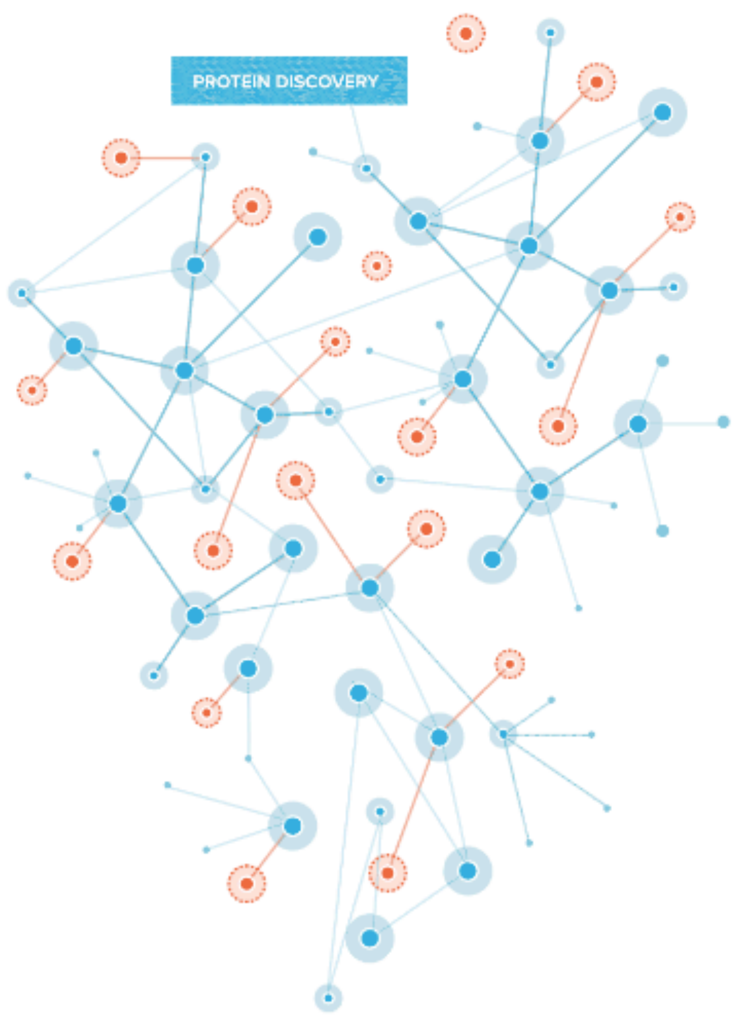
STARDOG Product Solutions Resources Company Learn Stardog Start for Free

Enterprise Knowledge Graph platform

Create a flexible, reusable data layer for answering complex queries across data silos. Stardog unifies data based on its meaning, creating a connected network of knowledge to power your business.

https://www.stardog.com

Why Knowledge Graphs?



A Knowledge Graph turns your *data* into machine-understandable *knowledge*. But what separates data from knowledge? Knowing this answer is key to understanding the definition of knowledge graphs.

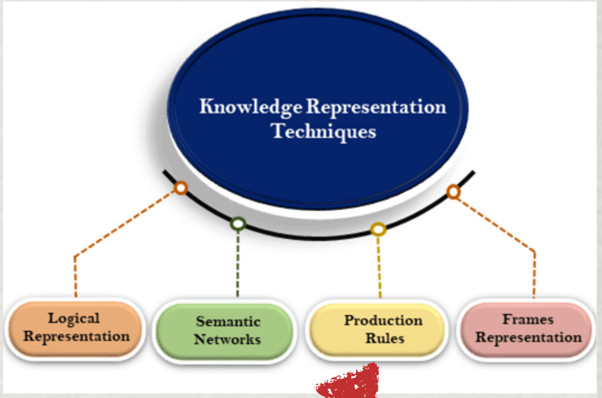
Real-world knowledge is:

- **situational**: meaning alters depending on circumstances
- **layered**: associations between concepts allow for nuanced understanding
- **changing**: new discoveries instantly change meanings

These facets of knowledge represent the *context* that is often missing from data. When traditional data management systems attempt to capture context, they fail. These failures generate gridlock over mastering data, delay timelines when adding new data sources or properties, and cause missing datasets from analyses that lead to mistrust.

Knowledge Graphs, on the other hand, are purpose-built for the fluctuating nature of knowledge. They offer a more flexible foundation for digital operations by easily accepting new data, new definitions, and new requirements.

Regras de produção



1. If Weather is Sunny

AND distance < = 20 miles

THEN transportation is bicycle.

2. IF transportation is bicycle

THEN no passenger insurance is considered.

3. IF no passenger insurance is considered.

THEN transportation insurance cost = 0.

Sentenças não-atômicas podem ser derivadas de um conjunto de sentenças por algum mecanismo de dedução. Por exemplo, uma sentença β é derivada de um conjunto de sentenças $\{s_i\}$, dizemos que

$$\{s_i\} \models \beta$$

Seja uma cláusula definida,

$$\neg p_1 \vee \neg p_2 \vee \ldots \neg p_n \vee q$$

composta por um conjunto de sentenças negadas p_i e somente uma sentença positiva q. Esta cláusula é equivalente a,

$$\overline{(p_1 \wedge p_2 \wedge \ldots p_n)} \vee q$$

$$p_1 \wedge p_2 \wedge \ldots p_n \rightarrow q$$

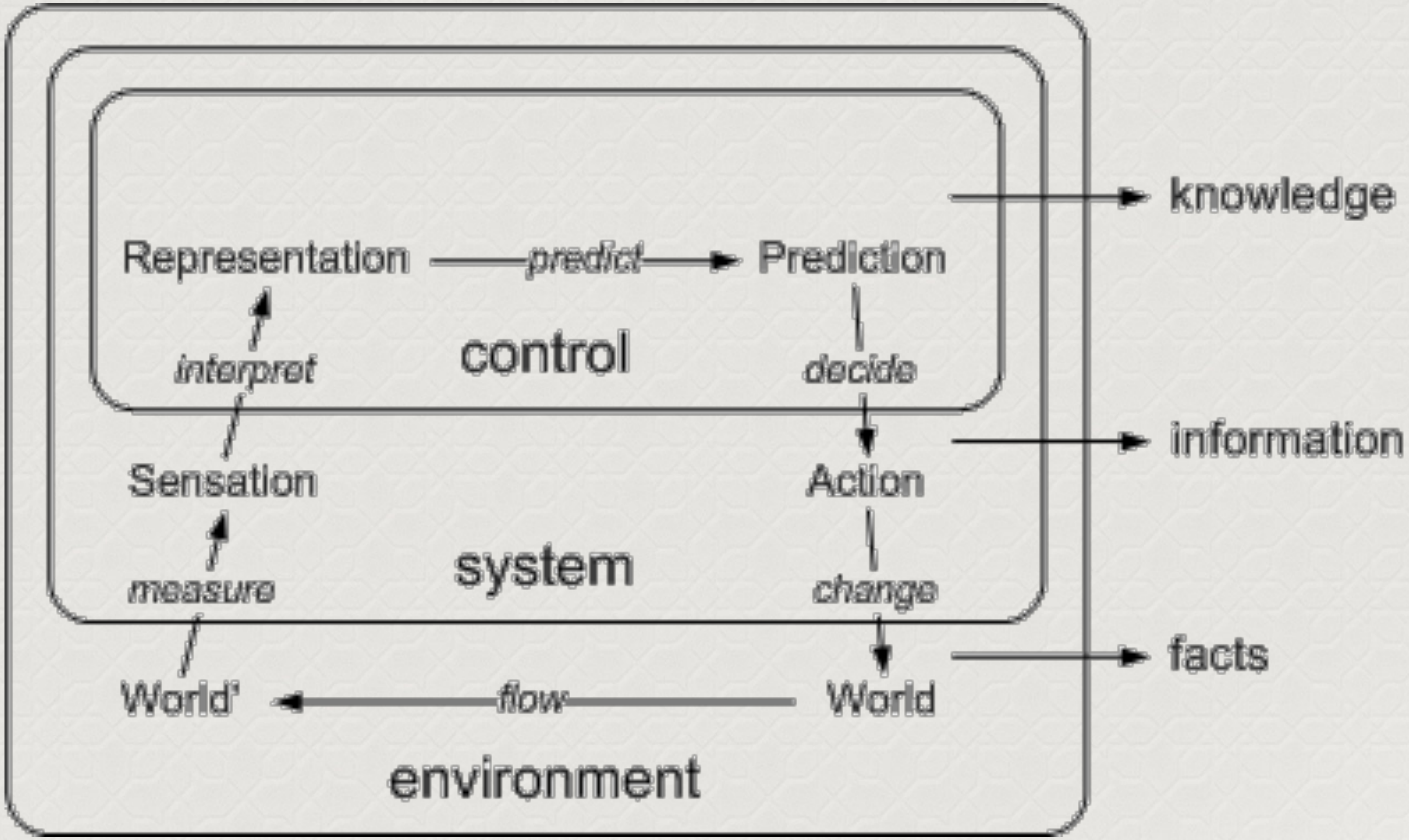
$$p_1 \land p_2 \land \ldots p_n \rightarrow q$$

$$q \leftarrow p_1 \land p_2 \land \ldots p_n$$

$$q :- p_1, p_2, \ldots p_n$$

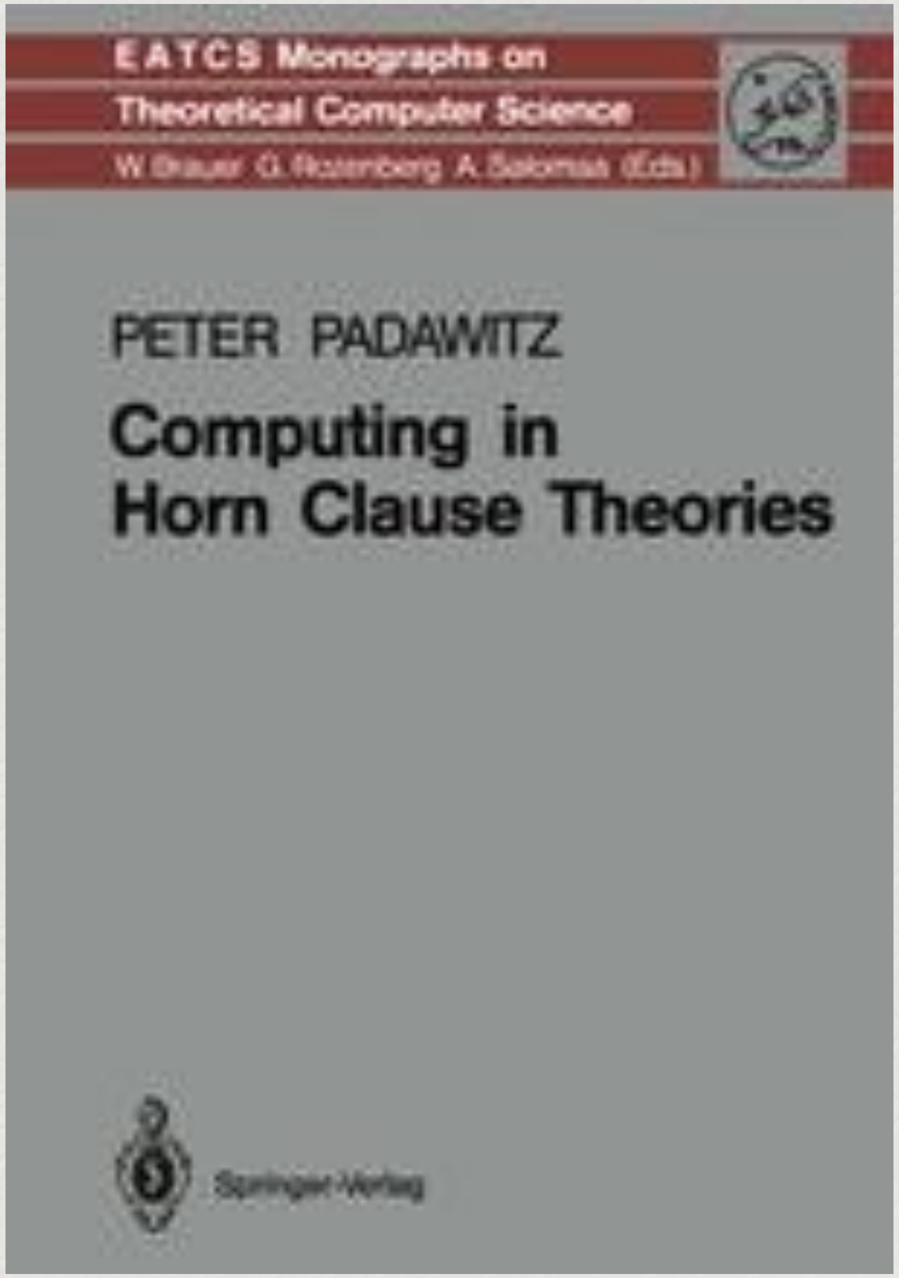
Portanto a representação de regras de produção em cláusulas if-then (else) é equivalente à representação em cláusulas de Horn e ambas podem ser expressas por uma forma normal (disjuntiva).

Supervisory



A representação de conhecimento baseada em regras de produção e cláusulas de Horn é usada em sistemas industriais, por exemplo nos supervisores.

Usando cláusulas de Horn



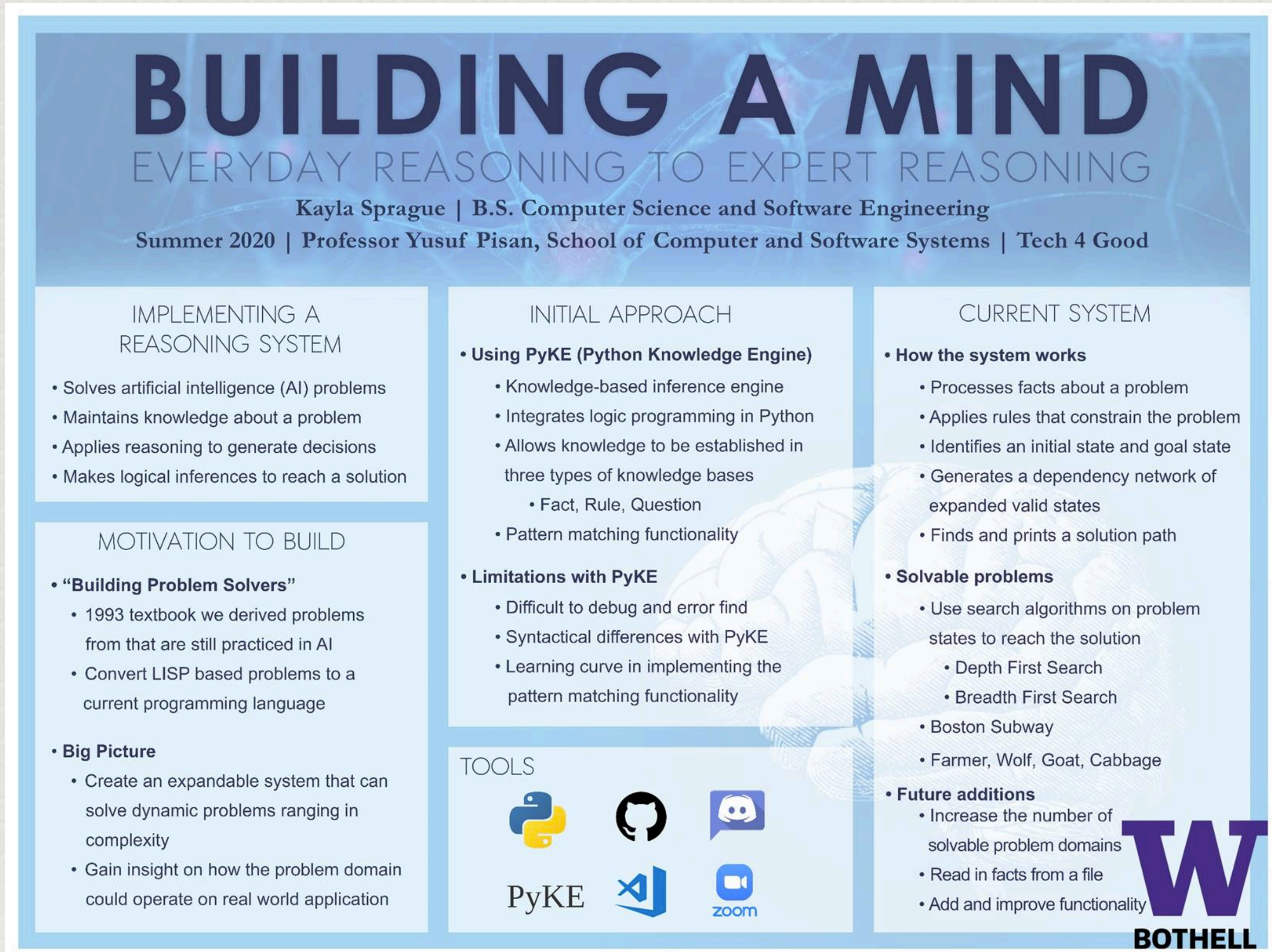
Usando regras de produção expressas em Cláusulas de Horn para construir sistemas especialistas.

https://www.metalevel.at/prolog/videos/predicate_logic

TABLE1 THE COMPARISON BETWEEN MATLAB, SCILAB, PROLOG AND LISP

Parameters / languages	License	Cost	Plug-Ins	User interface	Memory Requirement	Computation speed	Software updates	Functions and toolbox	Tutorials and demos
MATLAB	Proprietary	Very high	Available	User friendly	Very High	Very High	Available	Many	Easily available
Scilab	Open source	Open source	Open source	Open source	Open source	Open source	Open source	Open source	Open source
Prolog	Open source	Open source	Open source	Open source	Open source	Open source	Open source	Open source	Open source
LISP (LISt Processing)	Open source	Open source	Open source	Open source	Open source	Open source	Open source	Open source	Open source

PyKE python knowledge engine



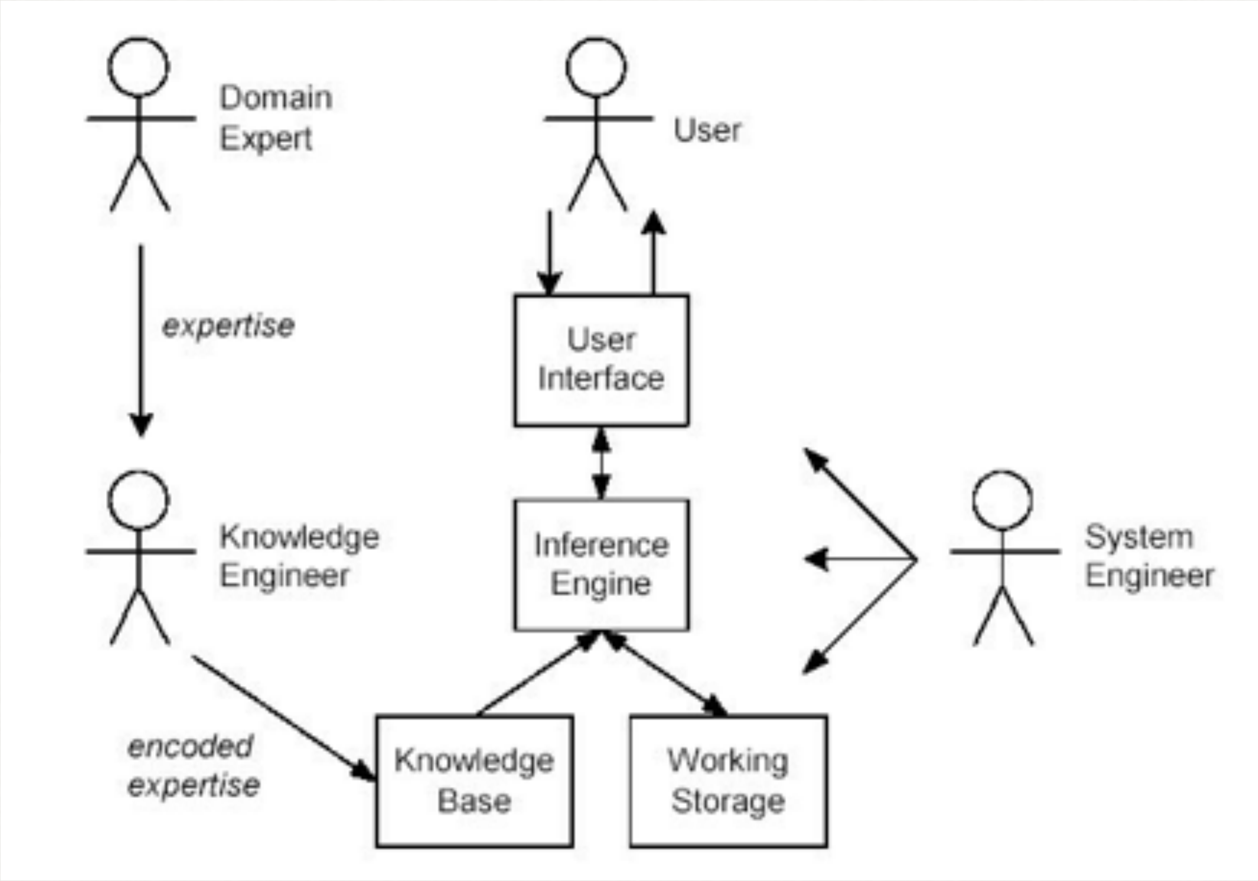
O uso comercial da linguagem Prolog

Existem vários desenvolvedores de ambientes de programação Prolog no mercado, como Quintus Prolog, SICStus Prolog, Visual Prolog, etc. Segundo o site da SICStus cerca de 30% dos processos de emissão de passagem aérea são feitos hoje em Prolog.

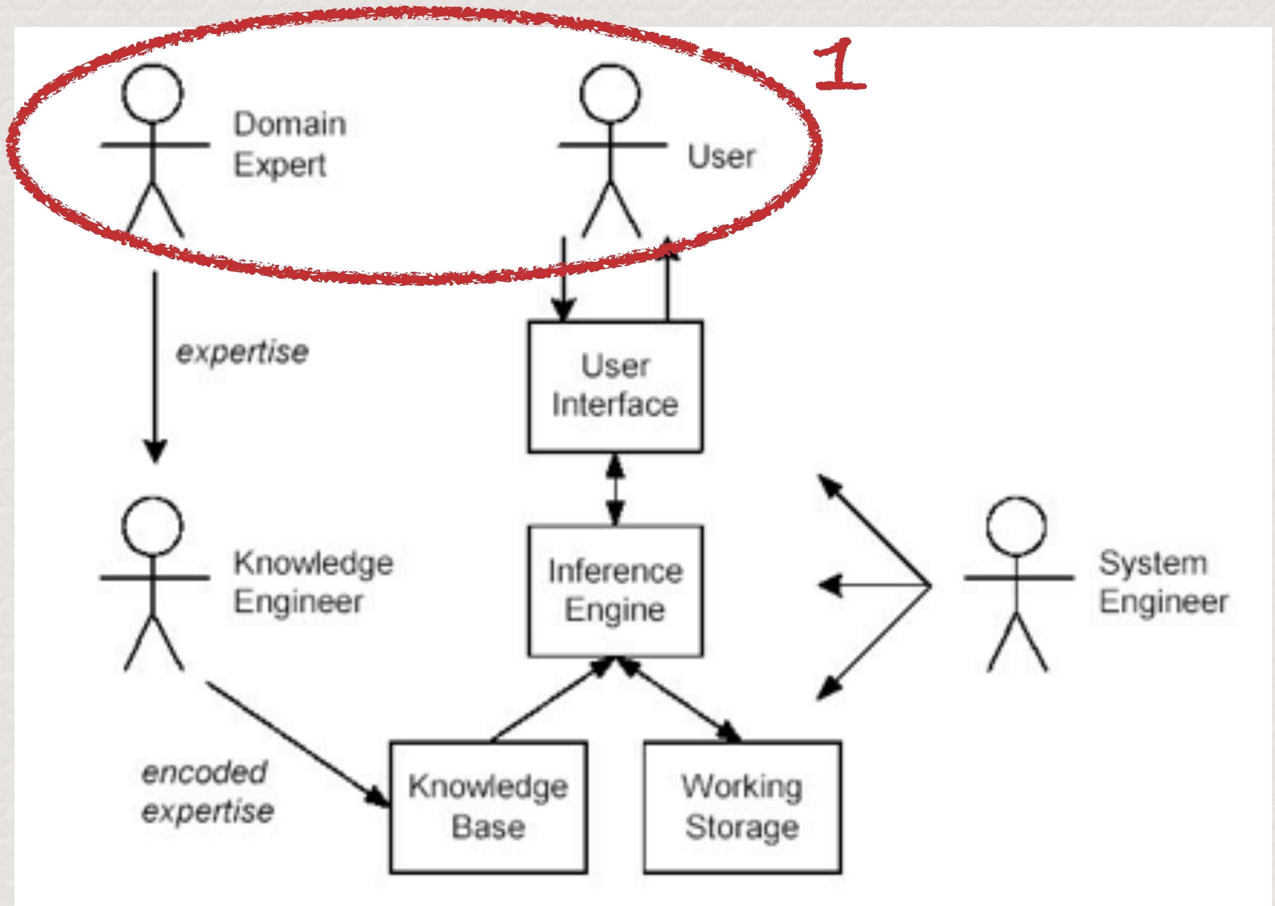
30.	Verilog	41.7
31.	D	41.1
32.	Julia	35.7
33.	Prolog	34.1
34.	Lisp	34.1
35.	LabView	33.3

Como se desenvolve um sistema especialista usando um KBE (Knowledge Based Engine) ou diretamente em alguma linguagem de programação lógica (Prolog, LISP, etc.)?

Expert System Architecture

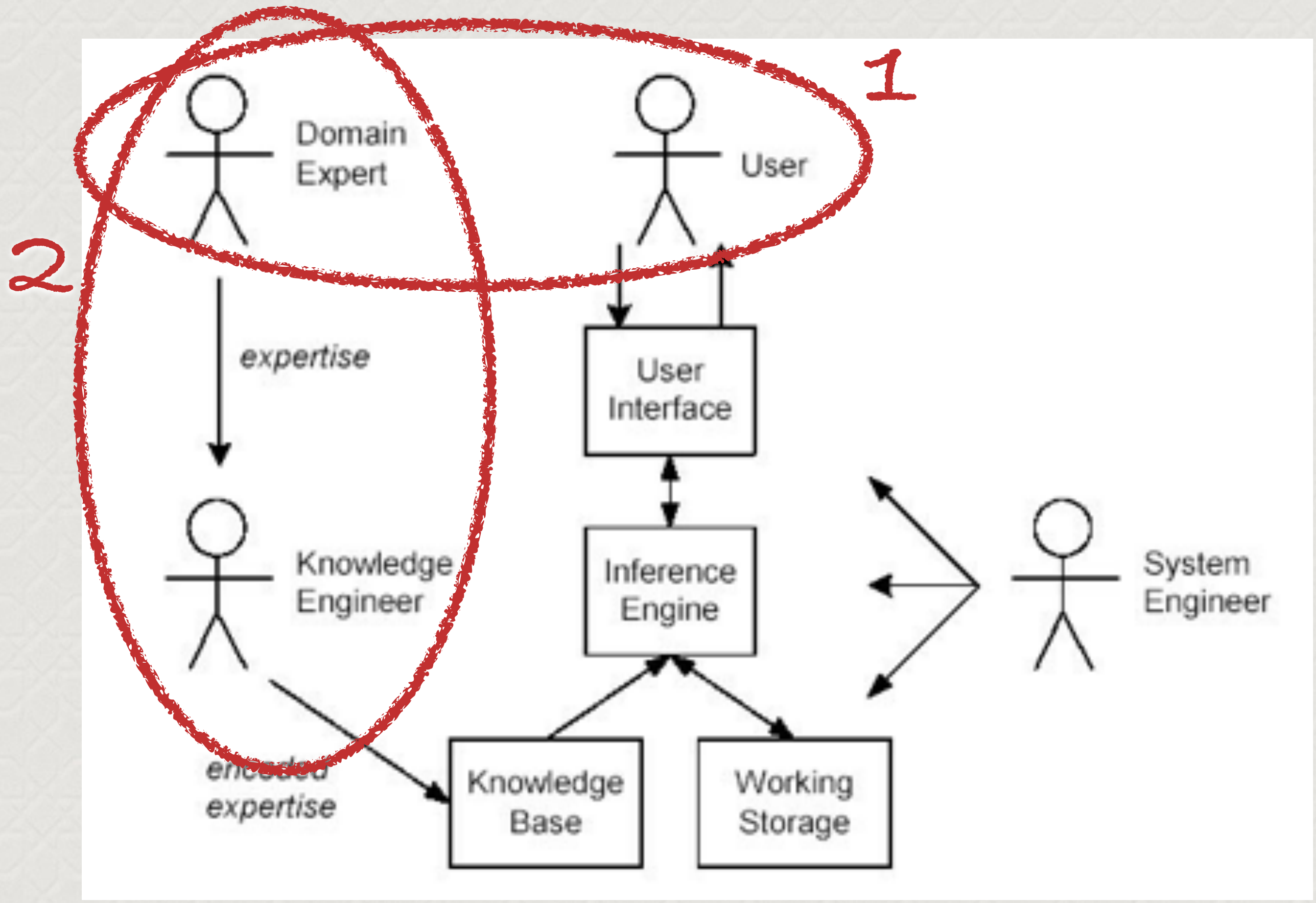


Expert System Design Cycle



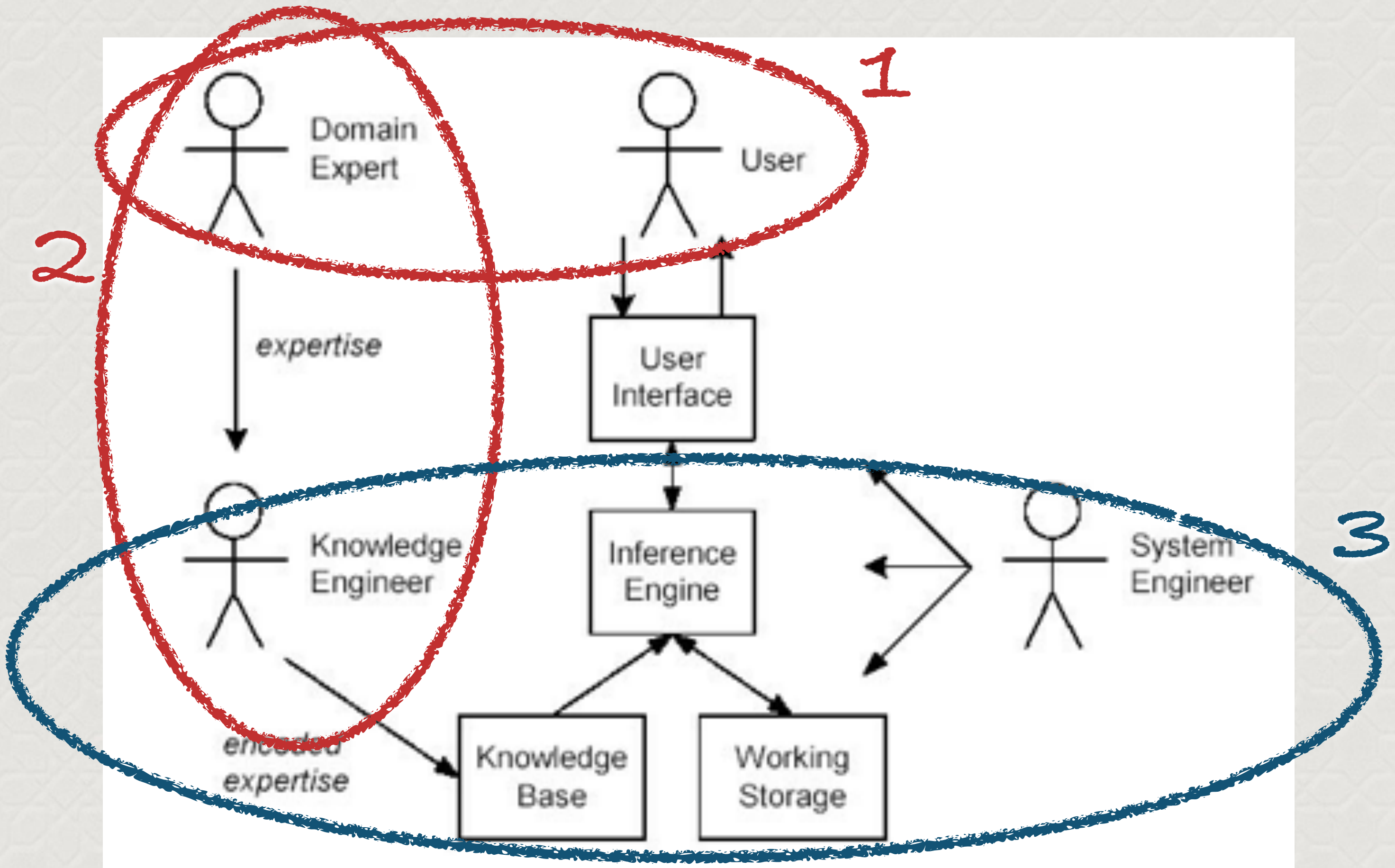
The Expert System Design cycle starts with the definition of the Domain Knowledge and how it could be used.

Expert System Design Cycle



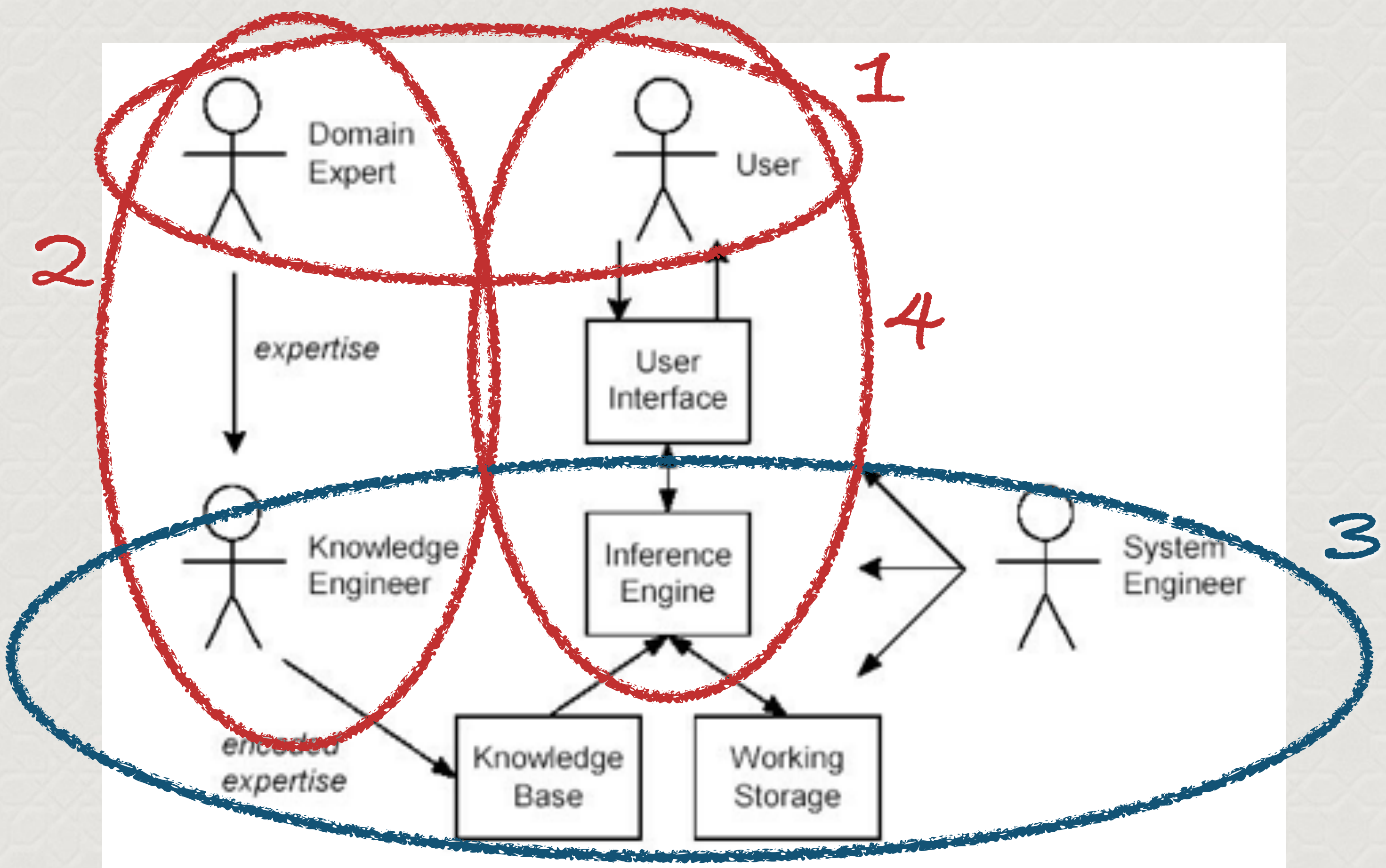
Next, we have to elicit knowledge from the expert in the chosen knowledge representation.

Expert System Design Cycle



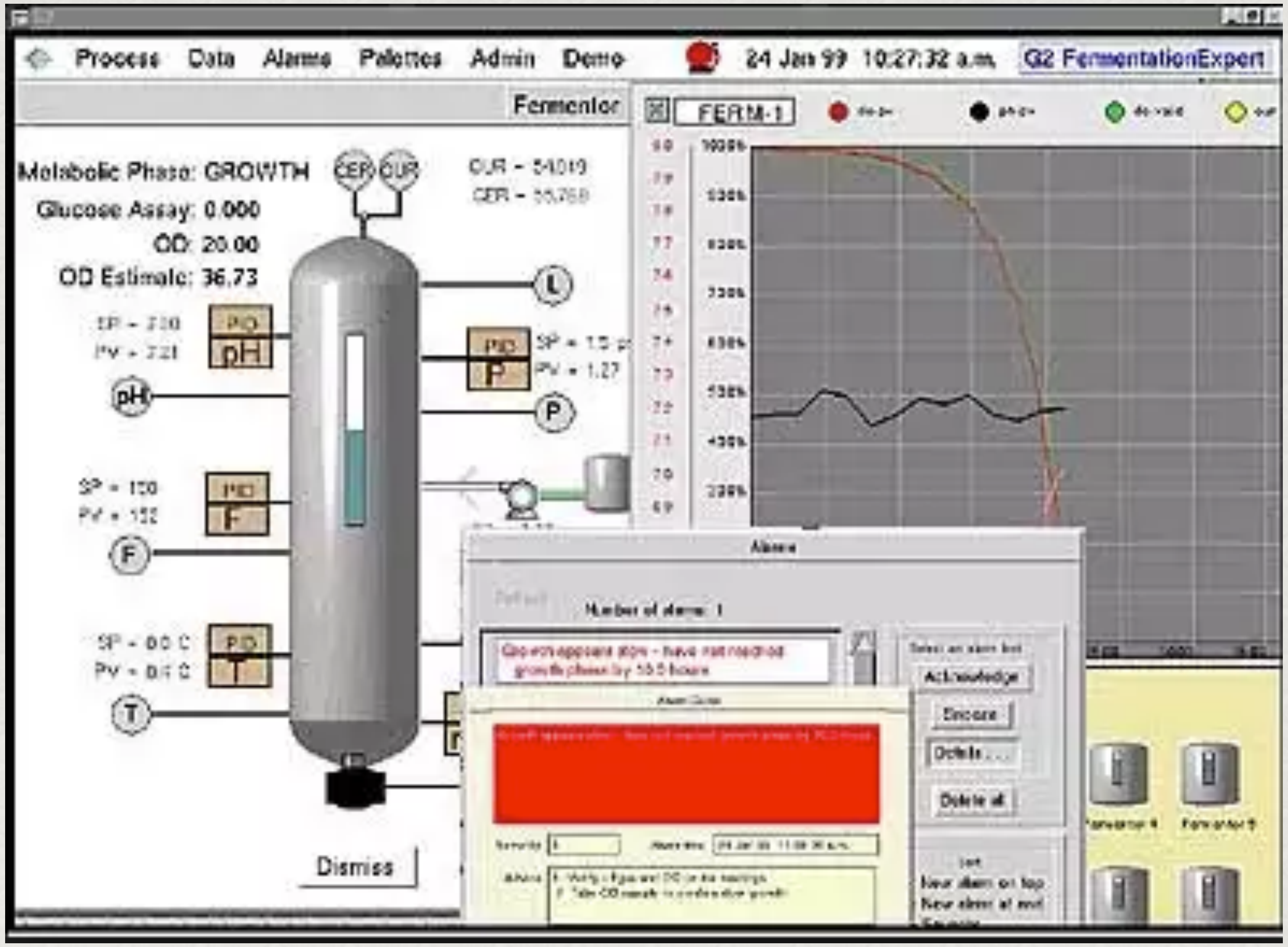
Then we reach the knowledge (and system) engineering process which consist of formalizing the KB with data and other processes associated with the inference engine.

Expert System Design Cycle



Finally, the integration between the inference behavior must be synchronized with the user journey and implemented in the interface.

Aplicação na indústria química



Na aula que vem discutiremos o funcionamento básico da máquina de inferência e como se pode interagir de forma sistêmica com esta máquina para obter o comportamento desejado do sistema especialista.

Perguntas?