Why Representation?

To establish correspondence between the WORLD & AGENT’s head

```
Initial facts                              Final facts
A  B  C  D  E                           A  B  C  E  D

Actions

Reasoning

Representation-1                      Representation-2
```
Theory of knowledge

Agent ——> Environment

Intelligent agents anticipate environment & results of their actions
Intelligent behaviors

Assume intelligent agents have “knowledge” about their environments

Theory of knowledge

- What are the types of knowledge? ——> heuristic knowledge
- How can we represent it? ——> heuristic function
- How do agents use it? ——> search
- How can it be acquired? ——> BUT THERE ARE MORE!!!

From Data to Knowledge

Hierarchy of Knowledge

Meta-knowledge
Knowledge
Information
Data
Noise

knowledge about picture
picture
enhanced image
image
pixel
Types of Knowledge

- Procedural
  - "How to" knowledge
  - Implicit
  - Task-oriented

- Declarative
  - "What is" knowledge
  - Explicit
  - Classification-oriented

- Tacit
  - "Unconscious" knowledge
  - Unexpressed by language

Knowledge representation (KR)

Process of representing knowledge formally

Declarative KR
- Expressed in terms of declarative statements
  - Can be changed, retrieved and extended easily
- Does not specify the order of how to use it
  - Can be reused for different purposes
  - Inefficient for "how to"

Procedural KR
- Expressed in ad-hoc programs crafted for special purpose
  - Implicit and hard to modify
  - Less flexible
- Specify the order of how to use the knowledge
  - Efficient
**Declarative vs. Procedural KR**

Which is better?

- Given a fridge full of groceries
  - Goal: To cook a dish for dinner
    Better to have a list of what’s in the fridge

- Goal: To cook tuna casserole
  Better to have a recipe for the dish

**Objective of KR**

To find ways to capture knowledge in a computer-tractable form to be used for problem-solving

- *Representation* goes hand in hand with *Reasoning*
**Good Representations**

Properties of good representation

- **Representational adequacy** → expressiveness
- **Inferential adequacy** → retrieval/access
- **Inferential efficiency** → computational performance
- **Acquisitional efficiency** → storage/organization/updates

**Good representations**

- **Make important objects and relations “explicit”**
  You can see what’s going on at a glance
- **Expose “natural constraints”**
  You express the way one object/relation influences another
- **Bring objects and relation together**
  You can see all you need at one time
- **Surpress irrelevant details:** can retrieve details later
- **Transparant**
  You can understand what is being represented
- **Complete and Concise**
  represent all that is needed efficiently
- **Computable and Fast**
  You can create them with existing procedure, and can store and retrieve information fast
**Example of a simple KR scheme**

A spreadsheet of class with student names, scores, grades, etc.

- **Representational adequacy**
  - Can I represent "student A didn’t turn in HW1"?
  - Can I represent class evaluation policy?

- **Inferential adequacy**
  - Can I find a student with the highest grade on Test1?

- **Inferential efficiency**
  - How fast can I find a student with the highest grade on Test1?

- **Acquisitional efficiency**
  - Is it easy to add new student records?

**KR process**

- **Conceptualization**: identify concepts, functions and relations (universe of discourse)

  \[
  \begin{array}{ccc}
  \text{A} & \text{B} & \text{C} \\
  \text{D} & \text{E}
  \end{array}
  \]

\[
\text{({A, B, C, D, E, \{support\}, \{on, above, clear, on\table\})} \\
\text{Concepts Function Relations}
\]

\[
\text{({A, B, C, D, E, \table\}, \{support\}, \{on, above, clear\})}
\]

- **Encoding**: encode the findings by an appropriate encoding scheme (KR form)
KR Forms

- Procedural Forms
  - Code (e.g. LISP code)
- Declarative Forms
  - Logic
  - Rules
  - Objects

Declarative KRs and languages

Declarative KR

Syntactic Systems
- Non-monotonic System
- Predicate Logic
  - PROLOG
  - A-PROLOG

Semantic Systems
- Production Rules
  - CLIPS
  - OPS5
- Slot-and-Filler
  - Strong
  - Weak
  - Frame
  - Semantic Nets
  - Conceptual Graph
  - Conceptual Dependency

- Strong
- Script
- Conceptual Dependency
- Frame
- Semantic Nets