

**Systems with Understanding Ability:  
Toward Advanced Agents**

Dr. Tuncer Ören

Emeritus Professor (University of Ottawa, Canada)

[oren@site.uottawa.ca](mailto:oren@site.uottawa.ca)

<http://www.site.uottawa.ca/~oren>

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding

# Motivations of the author 1/2

## Software agents

- are already used in many applications
- their usages are increasing
  - including agent-directed simulation
- they are becoming more advanced
  - such as agents with personality and emotions

## They can be even more advanced

- if they can have ability to “understand”
- ...

# Motivations of the author 2/2

- **Understanding** is  
an **interesting, important,** and **challenging** topic:
  - in philosophy
  - in artificial intelligence
- Ability to understand **has pragmatic values**  
in many application areas such as:
  - **program understanding**

# Presentations and Publications of Dr. Tuncer Ören on **Understanding**

[www.site.uottawa.ca/~oren/pubsList/understanding.htm](http://www.site.uottawa.ca/~oren/pubsList/understanding.htm)

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding

# A software agent:

- is an **autonomous** software module
  - with **perception** and **social ability**
- to perform **goal-directed knowledge processing** over time.
- can work
  - **on behalf of** humans or other software agents
  - **in** software or physical environments.

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding



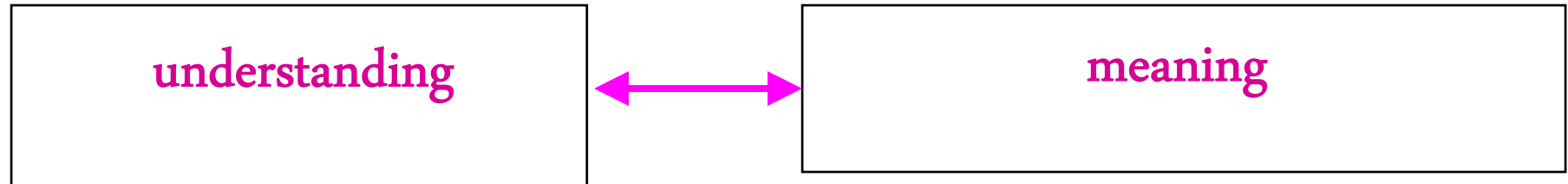
# Understanding - Dictionary definitions (Webster):

- to seize the meaning of  
<understand relativity>
- to grasp the reasonableness of  
<his behavior is hard to understand>
- to have thorough or technical acquaintance with or expertness  
in the practice of  
<understand finance>
- to be thoroughly familiar with the character  
<understand children>
- to accept as a fact or truth or regard as plausible without utter  
certainty  
<we understand that he is returning from abroad>

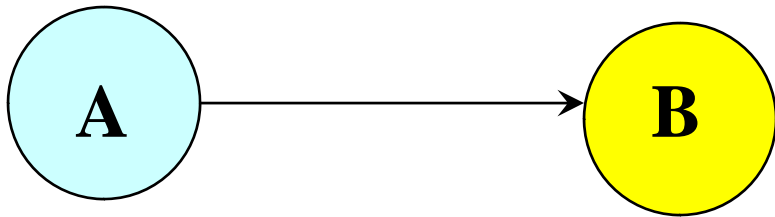
# Understanding - Dictionary definitions (Webster):

- to interpret in one of a number of possible ways
- to form a reasoned judgment concerning (something)
- to have the power of seizing meanings,  
forming reasoned judgments ...
- to believe or infer something to be the case  
<I understand she is not coming>
- to show a sympathetic or tolerant attitude toward  
<his manager understands him>
- to possess a passive knowledge of (a language)  
<he understands French>

# Relationship :



If A cannot **understand** B,  
B does not have a **meaning** for A.



J. Dewey (1910). How we think,  
D.C. Heath, Lexington, MA.

# Relationship

**understanding**



**knowledge**

“We say that a systems “**knows about**” a class of objects, or relations, if it has an internal ***knowledge about the*** relations for the class which enables it to operate on objects in this class and to communicate with others about such operations.”

(B.P. Zeigler, 1986)

“Thus, **if a system knows about X**, a class of objects or relations on objects,

it is able to use an (internal) representation of the class in at least the following ways:

- **receive** information about the class,
- **generate** elements in the class,
- **recognize members** of the class and **discriminate** them from other class members,
- **answer** questions about the class, and
- **take into account** information about changes in the class members.”

(B.P. Zeigler, 1986)

# A definition of understanding:

**Understanding** an entity

(a thing, a concept, an event, or a system)

**is** a mapping between

the **perceived *knowledge*** about the entity

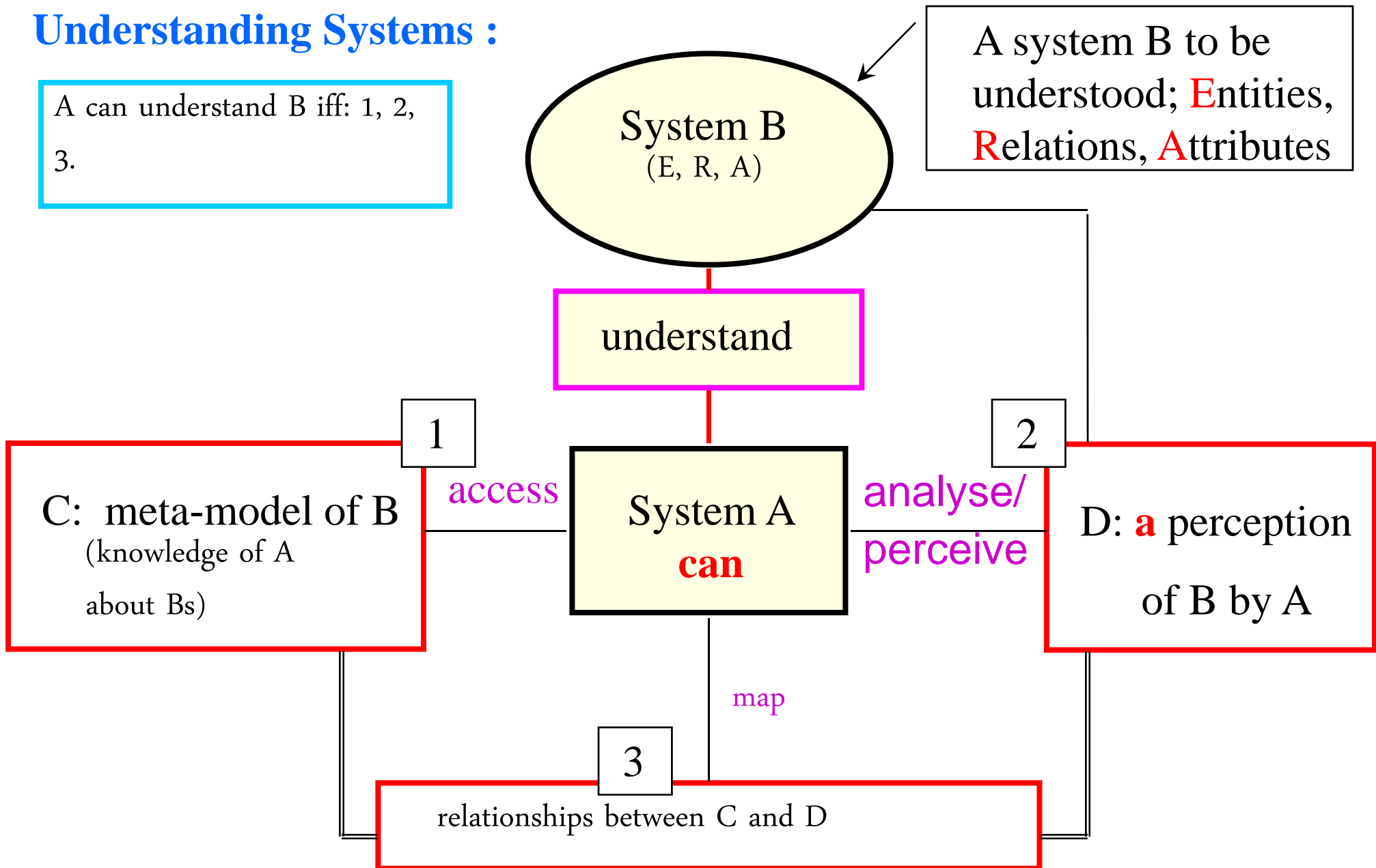
**and**

a ***meta-model*** (i.e., a more general knowledge) of the entity.

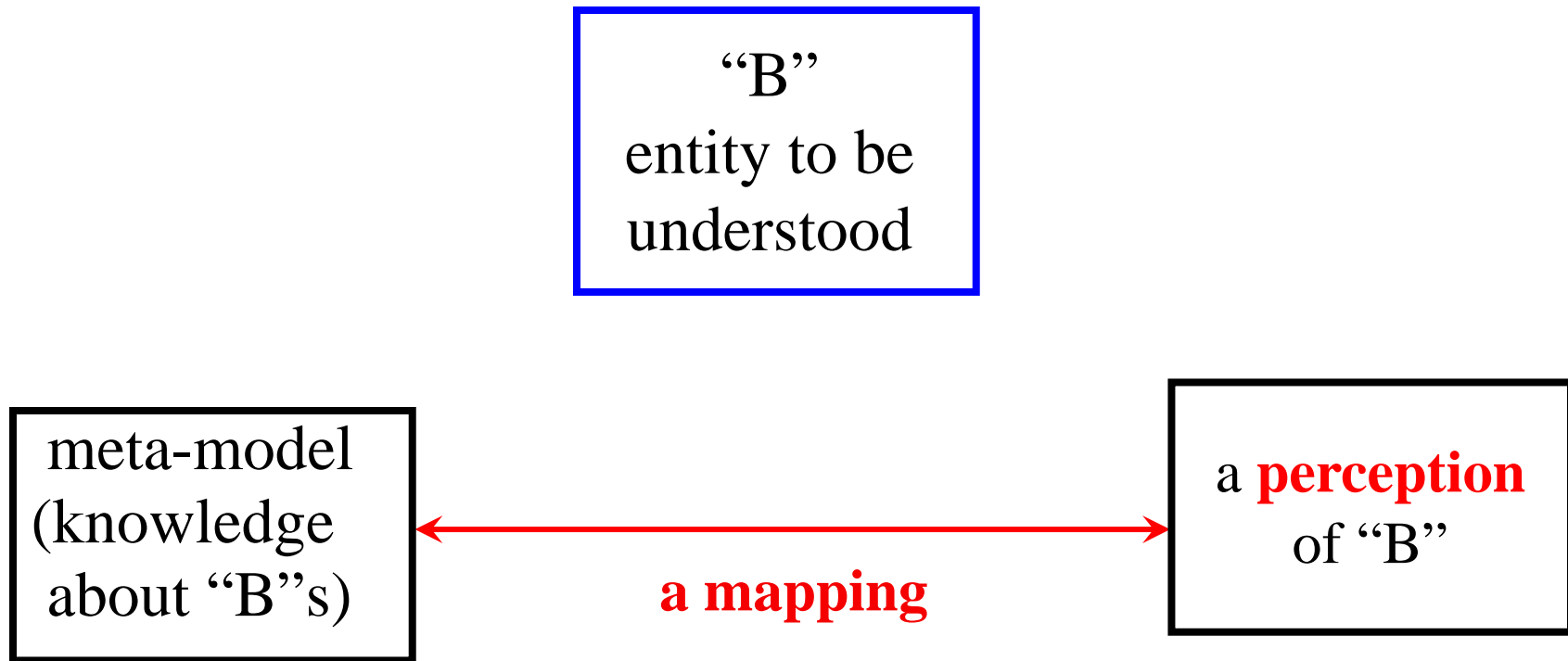
# Understanding Systems :

A can understand B iff: 1, 2, 3.

A system B to be understood; **E**ntities, **R**elations, **A**tttributes



# Understanding of an entity is a mapping (establishing a relationship):





# Ability to understand :

requires **intelligence and knowledge**



(advanced knowledge processing ability)

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding

A taxonomy requires identification of  
**elements** and **criteria**:

## Criteria:

- **product** of the understanding process
- understanding **process**
- **meta-model**
- characteristics of the understanding **system**

# Criterion - **Product** of the understanding process :

## **subcriteria:**

- domain
- nature
- scope
- granularity (level of detail)
- reliability

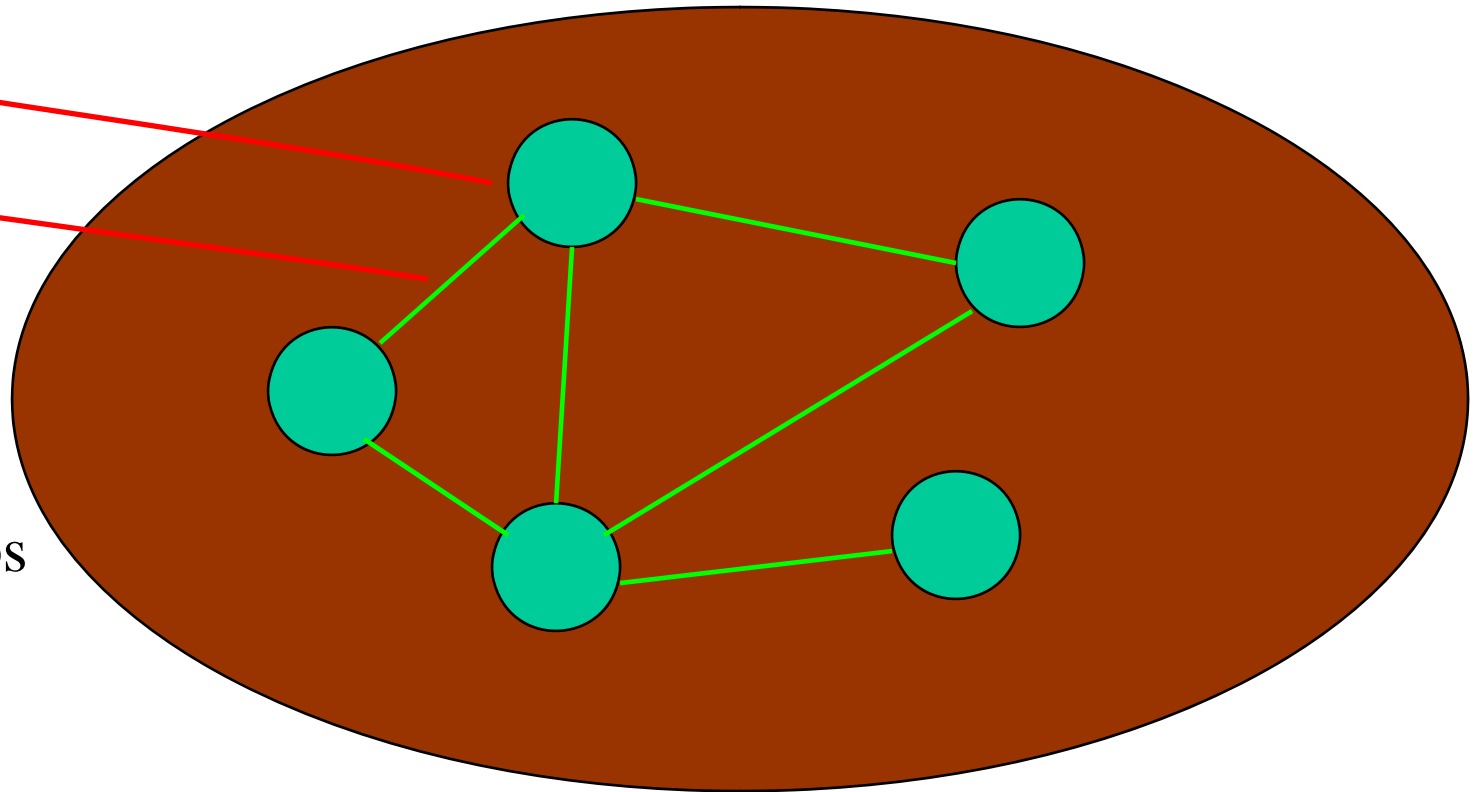
# **Product** of the understanding process

- **Domain** of understanding:

- **internal understanding**
- **external understanding**

## Internal understanding :

- elements
- relationships
- attributes
  - elements
  - relationships



## Internal understanding :

involves:

- **elements** of a system,
- **relationships** of the elements, and
- the **attributes** (time invariant / time varying) of the elements and the relationships.

In systems and software engineering terminology the system is considered as a **glass box**

## External understanding :

involves:

- **relationship** of a **system** and its **environment**.  
(time invariant / time varying)



In systems and software engineering terminology the system is considered as a **black box**



# **Product** of the understanding process

- **Nature** of understanding:

- **lexical understanding**
- **syntactic understanding**
- **morphological understanding**      **(structure)**
- **semantic understanding**      **(meaning)**
- **pragmatic understanding**      **(intention)**

# **Product** of the understanding process

- **Scope** of understanding:

- **focused understanding**

- **broad understanding**

**(understanding several or all aspects)**

# **Product** of the understanding process

- **Granularity** (level of detail) of understanding:

- **coarse understanding**
- **detailed understanding**  
(**in-depth understanding**)

# **Product** of the understanding process

- **Reliability** of understanding:

- **reliable understanding**
  - valid understanding
  - verified understanding
- **unreliable understanding**
  - invalid understanding
  - unverified understanding

# Criterion - understanding **process** :

## **subcriteria:**

- **direction** of the understanding process
- **directness** of the understanding process
- **accumulation of knowledge**

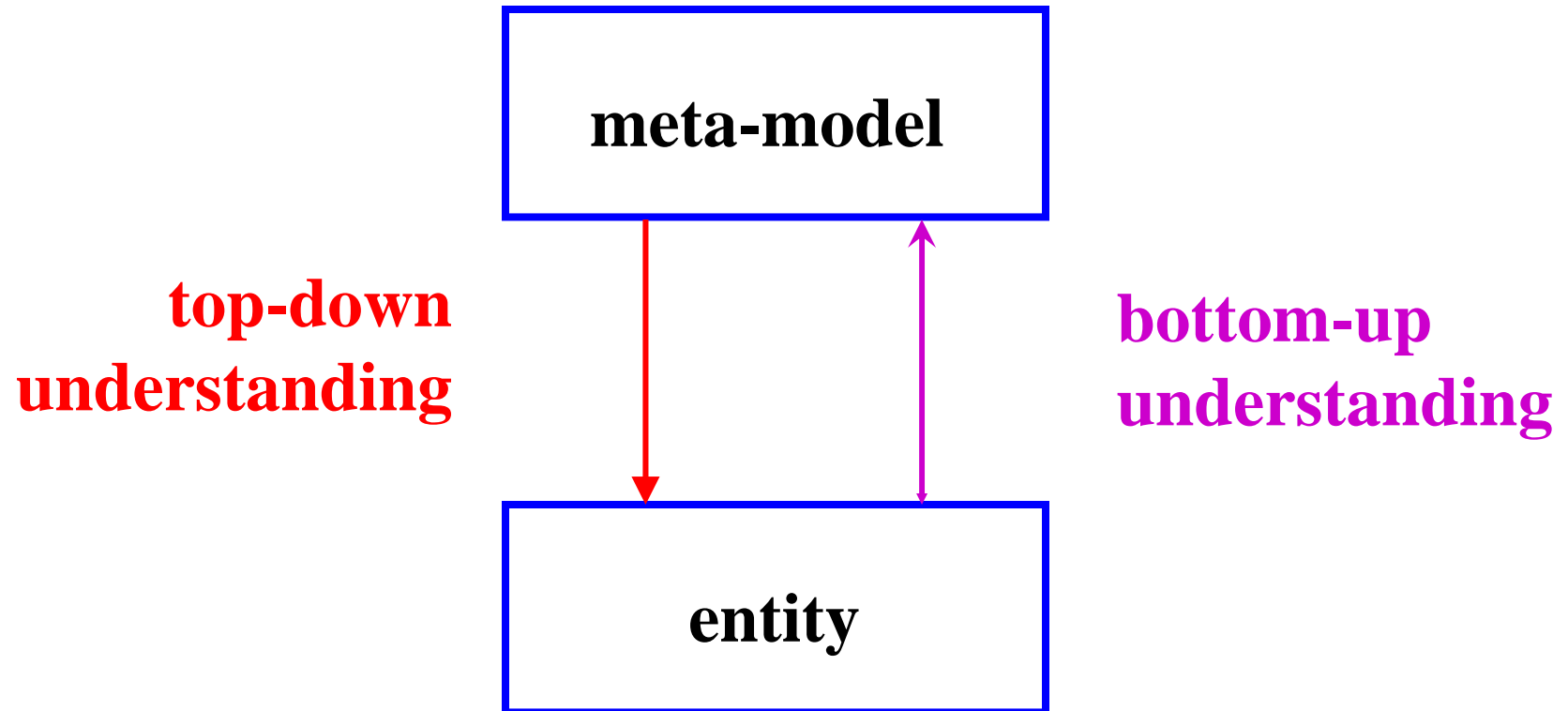
# Understanding **process**

- **Direction** of understanding:

- **top-down understanding**

- **bottom-up understanding**

# understanding process: direction



# Understanding **process**

- **Directness** of understanding:

- **direct understanding** - **apprehension**  
(intuitive understanding)
- **indirect understanding** - **comprehension**  
(logical understanding)



# Understanding **process**

- **Directness** of understanding:

- **direct understanding** - **apprehension**  
(intuitive understanding)

# Logical understanding:

- Access a *meta-model*
  - given (rote learning),
  - known and located,
  - searched and located  
(on the Internet, for example),
  - developed (!)
- For further details:
  - attempt to **refine / improve** the meta-model

# Criterion -

characteristics of **meta-model**:

## subcriteria:

- **fixed**
- **evolvable**
- **replaceable**

# Understanding **meta-model**

- **Fixed** meta-model:

- **single vision understanding**
  - dogmatic understanding

# Understanding **meta-model**

- **Evolvable** meta-model:
  
- **Learning understanding**

# Understanding **meta-model**

- **Replaceable** meta-model:

- **multivision understanding**
  - switchable understanding

# Criterion -

characteristics of understanding **system**:

## **subcriteria:**

- **initiative** of the understanding system
- **number** of understanding system
- **knowledge sharing** features of understanding system
- **mechanism to disseminate** the result of understanding process

# Understanding **system**

- **Initiative** of the understanding system:
  - **autonomous understanding**
  - **delegated understanding**
    - **remote understanding**  
(with mobile software agents, for example)



# Understanding **system**

- **Number** of the understanding system:

- **individual understanding**
- **group understanding**
  - collective understanding
  - distributed understanding

# Understanding **system**

- **knowledge sharing features**
- **repetitive understanding**
- **cooperative understanding**

# Understanding **system**

- **Dissemination mechanism**

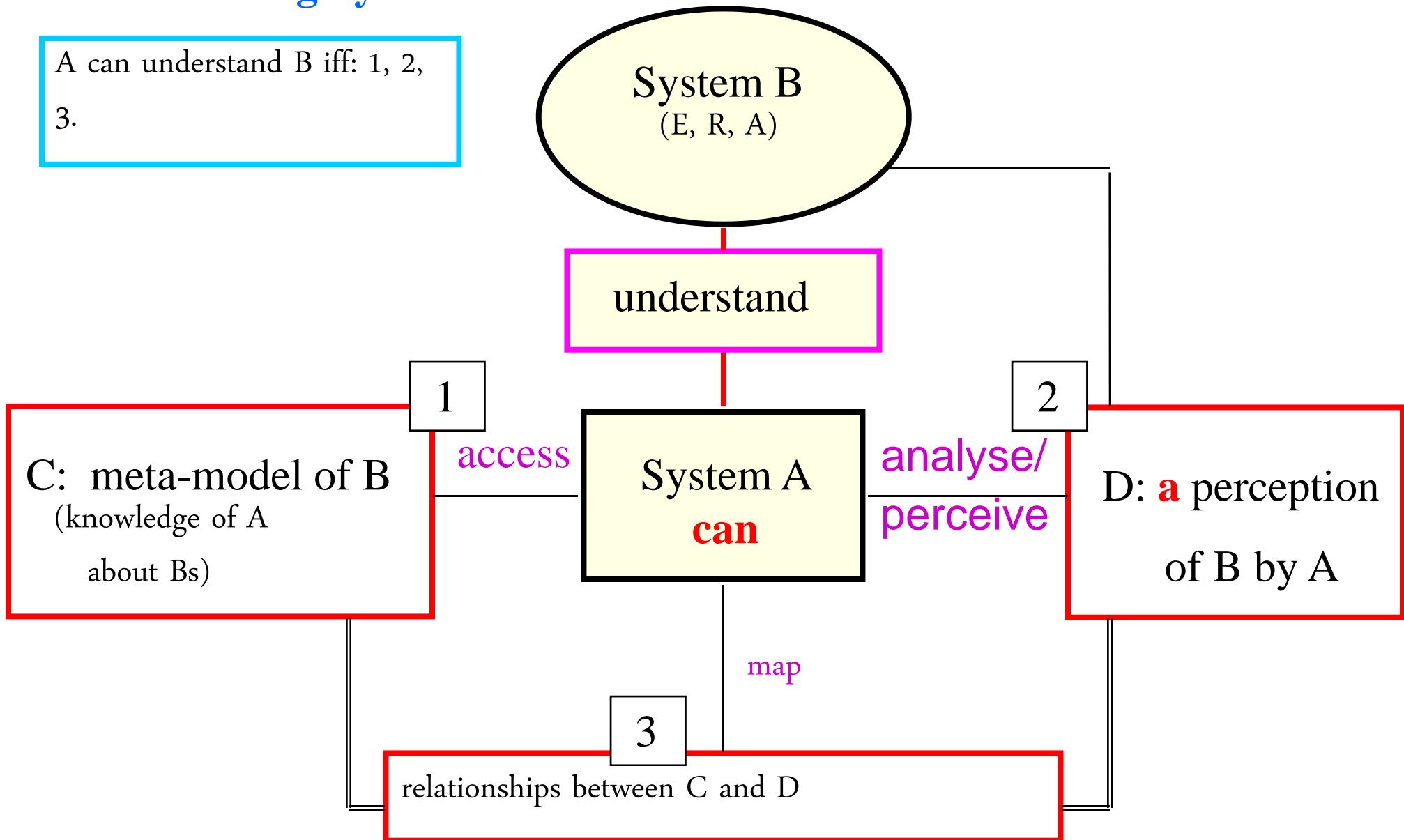
- **understanding per command**
- **understanding for subscribers**
- **broadcasted understanding**
- **blackboard understanding**
- **legacy understanding**

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding

# Understanding Systems :

A can understand B iff: 1, 2, 3.



# A's ability to understand B:

depends on

the restrictions of the three conditions; i.e.,

- **meta-model**
- **perception / analysis**
- **mapping**

For a system, another system can be

**intelligible** or **unintelligible**;

**i.e.**, intelligibility of a system often depends on the abilities of the perceiving system.

# Performance Factors

## 1. **Meta-model**: A's knowledge about Bs

- A system that does not have basic knowledge (meta-model) about another system cannot understand it.
- The scope of understanding is bound by the types of knowledge in the meta-model.
- Superiority of *understanding-based learning* over rote learning.



- **Access time** to the meta-model affects the *speed of understanding*.

- The granularity (the level of details) of the meta-model determines the **discrimination ability** of the understanding system.
- The **content and granularity** of the meta-model **depend on:**
  - the types of knowledge required from the understanding system.  
(**the types of questions it should be able to answer**)

# Performance Factors

## 2. Perception

“B” (the system to be understood)

can **be submitted** to “A”.

- In this case, “A” needs to analyse “B” with respect to “C” (a meta-model of “B”s)

“A” may need to **perceive** “B”

- In this case, A needs to keep track of some external / internal events, and recognize, classify, and filter them.

- The level of details (**granularity**) of the perception / analysis determines the **granularity of understanding**.

(**What is not perceived or discriminated in the analysis cannot be understood.**)

- The speed of perception / analysis affects the **speed of understanding**.

# Performance Factors

## 3. Relationship

- To understand a system “B,” a system “A” needs to perform a mapping between:
  - a meta-model “C” of “B”s, and
  - “D:” a perception of “B”  
or a result of the analysis of “B.”
- The types of relationships affect the **limits of understanding**.
- The speed of mapping affects the **speed of understanding**.

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding
  - 6.1 Types
  - 5.2 Elements
  - 5.3 Applications

# Types of program understanding systems:

## 1. **Static**

- analysis of the program without running it.

## 2. **Dynamic**

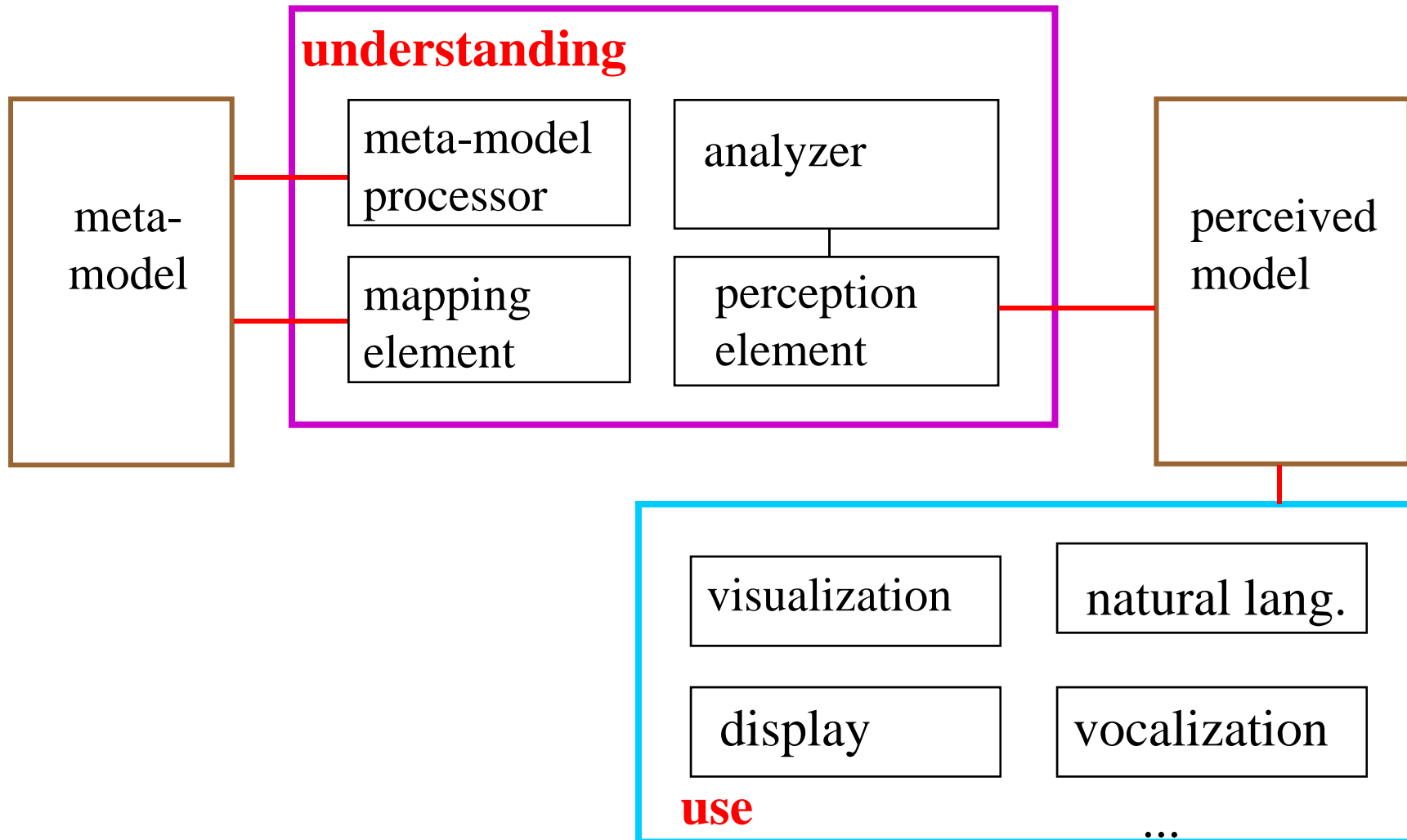
- observing the program while it is running.

# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding
  - 6.1 Types
  - 5.2 Elements
  - 5.3 Applications



# Elements of program understanding systems:



# Plan

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding
  - 6.1 Types
  - 5.2 Elements
  - 5.3 Applications

# Applications of program understanding systems:

## 1. **Automatic documentation**

of software:

- **display**

- **text**

- annotation
    - categorization
    - natural language doc.

- **visualization**

- data structures
    - control structures
    - object-oriented

- **sonorization**

- **speech** synthesis
  - **voice** annotation

## 2. **Answering questions** about a software (for maintenance)

This ability requires the following abilities:

- abstraction
- generalization
- filtering
- communication in natural language  
(both ways)

3. **Certification** of the existence or lack of some properties.
4. **Computer aided** verification and validation of software.

5. **Critique** of user programs  
(One uses the perceived image “D” and the meta-model ‘C.’)
  - explanations of the critique
  - recommendations to improve user software
6. **Explanations** of programs
7. Extraction of **re-usable components**
8. **Tools for automatic testing** of software

Criteria related with the characteristics of the		types of understanding
<b>product</b> of the understanding process	domain	<ul style="list-style-type: none"> <li>- internal understanding</li> <li>- external understanding</li> </ul>
	nature	<ul style="list-style-type: none"> <li>- lexical understanding</li> <li>- syntactic understanding</li> <li>- morphological understanding (understanding the structure)</li> <li>- semantic understanding (understanding the meaning)</li> <li>- pragmatic understanding (understanding the intention)</li> </ul>
	scope	<ul style="list-style-type: none"> <li>- focused understanding</li> <li>- broad understanding (understanding several or all characteristics)</li> <li>- multiaspect understanding</li> </ul>
	granularity (level of detail)	<ul style="list-style-type: none"> <li>- coarse understanding</li> <li>- in-depth understanding (detailed understanding)</li> </ul>
	reliability	<ul style="list-style-type: none"> <li>- reliable understanding</li> <li>- valid understanding, - verified understanding</li> </ul>

Criteria related with the characteristics of the		types of understanding
understanding <b>process</b>	direction	<ul style="list-style-type: none"> <li>- top-down understanding</li> <li>- bottom up understanding</li> </ul>
	directness	<ul style="list-style-type: none"> <li>- apprehension (direct understanding)</li> <li>- comprehension (indirect understanding, mediated understanding)</li> <li>- logical understanding)</li> </ul>
	accumulation of knowledge	<ul style="list-style-type: none"> <li>- re-initialized understanding (tabula rasa understanding)</li> <li>- cumulative understanding</li> </ul>



Criteria related with the characteristics of the		types of understanding
<b>meta-model</b> used	fixed	- single vision understanding - dogmatic understanding
	evolvable	- learning understanding
	replaceable	- multivision understanding (switchable understanding)
understanding <b>system</b>	<i>initiative</i> of the understanding system	- autonomous understanding delegated understanding - remote understanding
	<i>number of</i> understanding system	- individual understanding - group understanding - distributed understanding
	<i>knowledge sharing</i> features of understanding system	- repetitive understanding - cooperative understanding
	mechanisms to <i>disseminate</i> the result of understanding process	- understanding per command - understanding for subscribers - broadcasted understanding - blackboard understanding - legacy understanding -

# 7. Challenges

Designing architectures and realization of advanced agents with different types of understanding (for several types of applications)

# We Have Seen

1. Motivations of the speaker
2. Software agents
3. Understanding: Basic concepts and definitions
4. A taxonomy of understanding
5. Performance factors of understanding systems
6. An example: Program understanding
7. Challenges

**Thank you for your attention!**