Mathematical Logics
Introduction*

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

*Originally by Luciano Serafini and Chiara Ghidini
Modified by Fausto Giunchiglia and Mattia Fumagalli

Outline

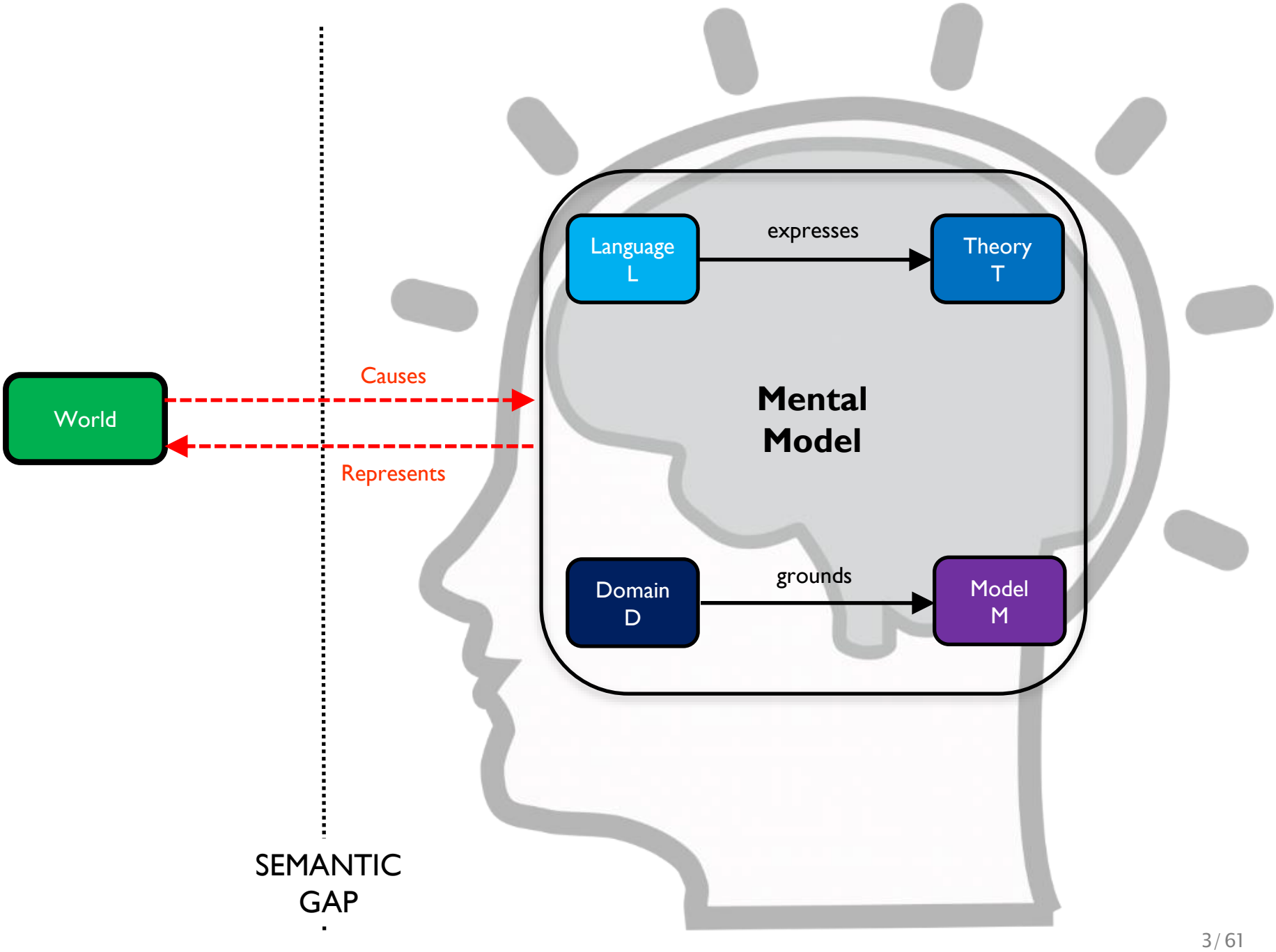
1. Mental, computational and logical models
2. Language
3. Logical modeling
4. Why Logic? Formal and informal languages/ models

Models and Conceptual Modeling

- A **(conceptual) model** is a meaningful representation of a portion of the world, described in a certain **language**

- **(Conceptual) modeling** is the activity which leads to the construction of (conceptual) models

Mental Model – how we represent the world



Mental Model – how we represent the world

- World: What we perceive
- Mental Model: A mental representation of world, decomposed in 4 constituents
- Semantic gap: The difference between world and mental model

- Language: ***Alphabet+syntax*** used to describe the world (for instance: "monk", "banana", "near", "and", "or")
- Theory: ***sentences*** describing what is ***true*** in the world, also called ***facts***, (for instance: {"monkey near banana", "banana on tree and banana is yellow"})

- Domain: Images / pictures which represent ***atomic elements*** used to describe what we see (for instance: objects like: monkey; facts like monkey near banana)
- Model: Images / pictures which represent the sets of facts (scenes) that we have seen (for instance: the scene described by {"monk near banana", "banana on tree and banana is yellow"})

The link from language to the images is in the mind of the person looking at the world – but not (!) in the mind of the others

Mental Model – how we represent the world

World

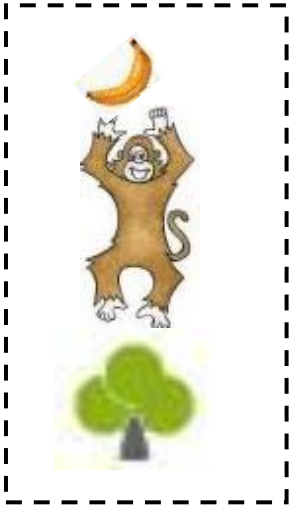
Mental Model

Language L

Domain D

Theory T

Model M



L: "*MonkeyclimbedTree, MonkeyNearBanana, MonkeyEatedBanana, and, or, therefore, …*"

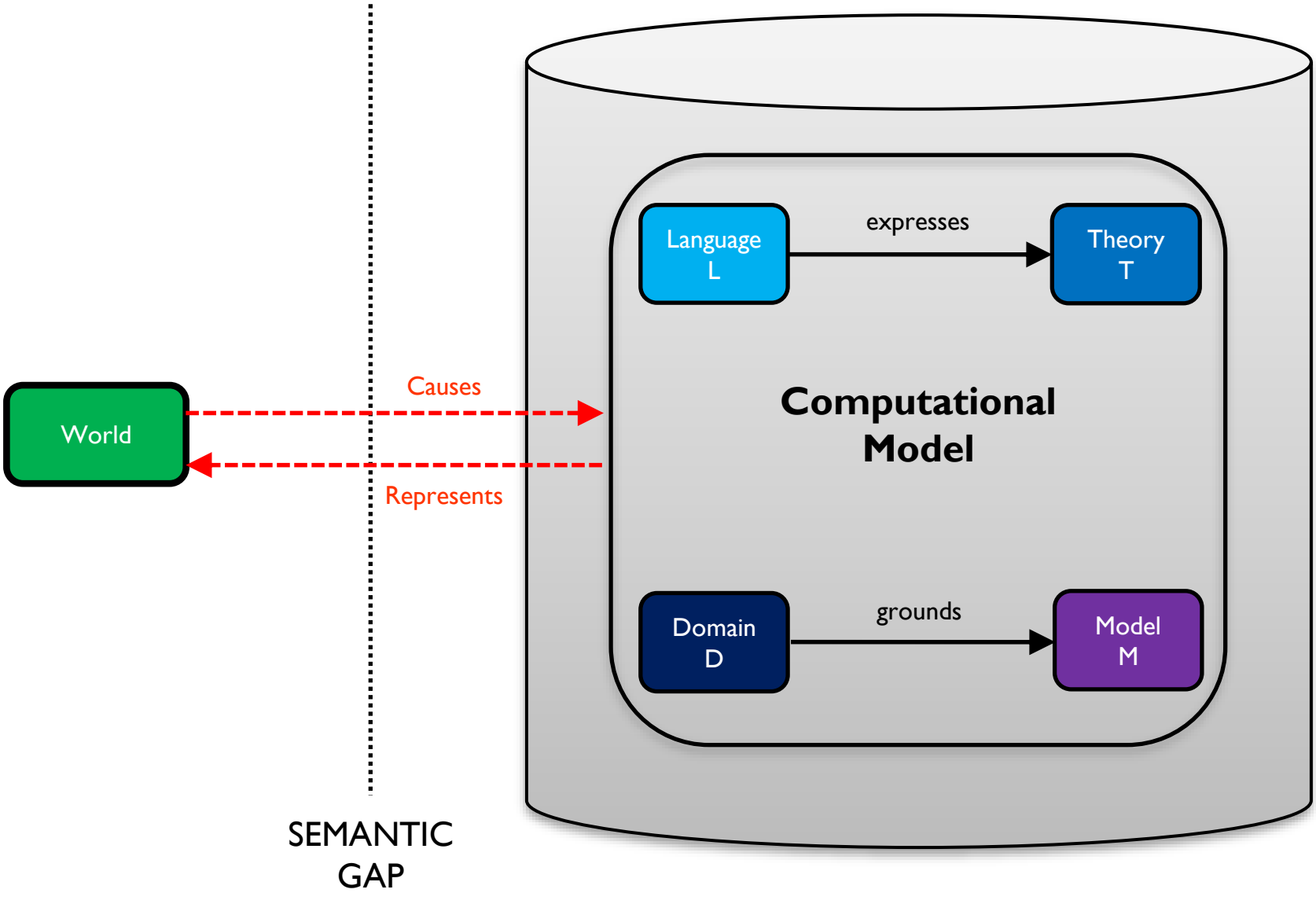
T: "*MonkeyclimbedTree, MonkeyNearBanana*"

D: {  }

M: "  "

SEMANTIC GAP

Computational Model – how we implement programs



Computational Model – how we implement programs

World

Computational Model

Language L | Domain D | Theory T | Model M

MONKEY

/	A	N	G
M	T #1	Ba #2	Ba #3
…	…	…	…

L: "*MCT, MAT, MBeBa, MNBa, MCR, MGBa, if-then-else, while, …*" (abstraction of language of the table)

T: "*MAT*, *MNBa*, *MGBa*" (actual table)

D: {#1, #2, #3} (possible memory locations)

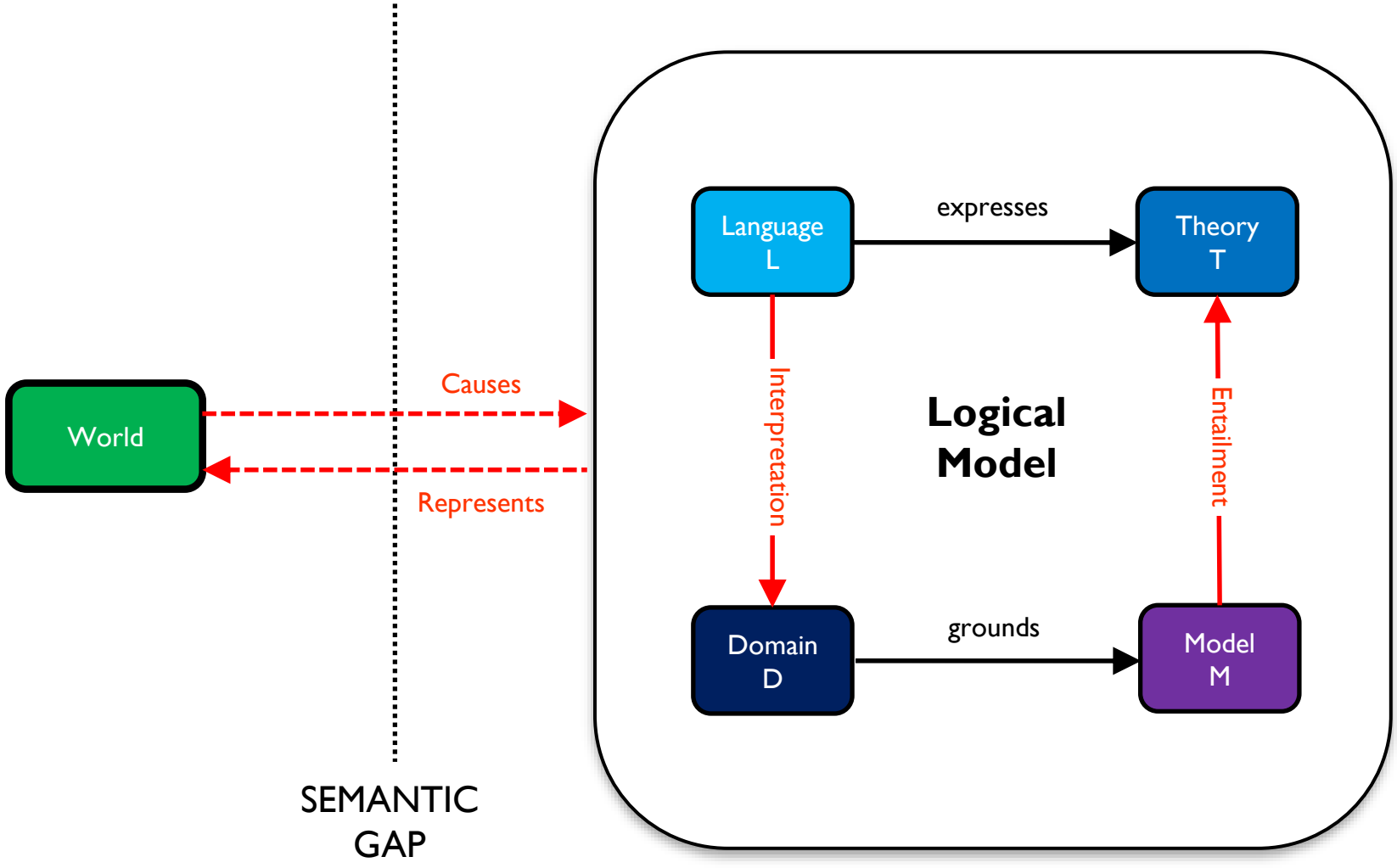
M: "*#1, #3*" (actually used memory locations)

SEMANTIC GAP

monkey(M), climb(C), tree(T), above(A), below(Be), near(N), banana(Ba), ock/R), get(G).

NO computer mental model. The mental model is only in the mind of developer

Logical Model – how we make explicit what we mean



Meaning of language made explicit. It can be used to describe both mental and computational models

Logical Model – how we make explicit what we mean

World

Logical Model

Language L

Domain D

Theory T

Model M

MONKEY			
/	A	N	G
M	T #1	Ba #2	Ba #3
…	…	…	…

L = "*MCT, MAT, MBeBa, MNBa, MCR, MGBa,* $\wedge$, $\vee$, $\neg$, $\rightarrow$, …"

T = "$MGBa \rightarrow (MAT \vee MNBa)$"

D: {#1, #2, #3}

I: "I(MAT) = #1, I(MNBa) = #2, …

M: "*#1, #2, #3*"

M $\models MAT$

M $\models MNBa$

M $\models MAT \vee MNBa$"

SEMANTIC GAP

Logical Model

- World: What we perceive
- ~~Mental~~ Logical Model: A ~~mental~~ ***Logical*** representation of world, decomposed in 4 constituents
- Semantic gap: The difference between world and ~~mental~~ ***Logical*** model
- Language: ***Logical*** ***Alphabet+syntax*** used to describe the world (for instance: "monk", "banana", "near", "and", "or")
- Theory: ***sentences*** describing what is ***true*** in the world, also called ***facts***, (for instance: {"monk near banana", "banana on tree and banana is yellow"})
- Domain: ~~Images / pictures~~ ***(Sets of) elements*** which represent ***atomic elements*** used to describe what we see (for instance: objects like: ~~monk~~ ***a,b,c, … monk*** ; facts like ~~monk near banana~~, ***A, B, C Monk_near_Banana***)
- Model: ~~Images / pictures~~ ***(Sets of) elements*** which represent the sets of facts (scenes) that we have seen (for instance: the scene described by {***A, Monk_near_Banana***})
- Interpretation: ***a function which associates each and any element of the language to one and only one element of the domain***
- Truth-relation / entailment /satisfiability (⊨): ***a relation which associates what is true in the model with a subset of the sentence of the language. A sentence can be an element in a theory if and only if its interpretation is true in the model***

NOTE: mental models always finite (L,T,D,M) while logical models often infinitary (L,T,D,M; (example of L: A and A and A and …; example of D: Natural Numbers)

Why Logical Models

Logical models make precise what we mean when we describe something

- Useful in the interaction developer - customer:
  - Customer: how am I sure that you are implementing the system which does what I want
  - Developer: how am I sure that you will not change the requirements later

- Mainly useful in high value applications (e.g., safety critical applications, security critical applications) because of its cost. *Largely solved. Lots of solutions in the market.*

Why Logical Models (cont'd)

Logical models make precise what we mean when we describe something

- Useful in the integration between two developers/ programs
  - How are we sure that a program understands the output of another program
    - Syntactic compliance. *Easy, via standards.*
    - Semantic/meaning compliance (e.g., meaning of word Java). *Very hard.*

- Useful in high value application (e.g., safety critical applications, security critical applications). *Largely solved*

- Useful anytime you need system interoperability (e.g., Web applications, web services). *Largely unsolved, because of open unpredictable open world, as it is the case, for instance, in the Web.*

Why Logical Models (cont'd)

Logical models make precise what we mean when we describe something

- Useful to build intelligent programs capable of autonomous reasoning (e.g., expert systems, decision support systems, Artificial Intelligence (AI) systems, intelligent Software agents)
  - Explicit semantics allow to provide a formal/ computational notion of (deductive) reasoning and to be guaranteed that the reasoning performed by programs is "correct". *Very hard*
- Useful in the next generation AI based computer systems. Goal is the integration of inductive reasoning (machine learning) and deductive reasoning (logical reasoning). *Largely unsolved.*

Mathematical Logics
Introduction*

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

*Originally by Luciano Serafini and Chiara Ghidini
Modified by Fausto Giunchiglia and Mattia Fumagalli