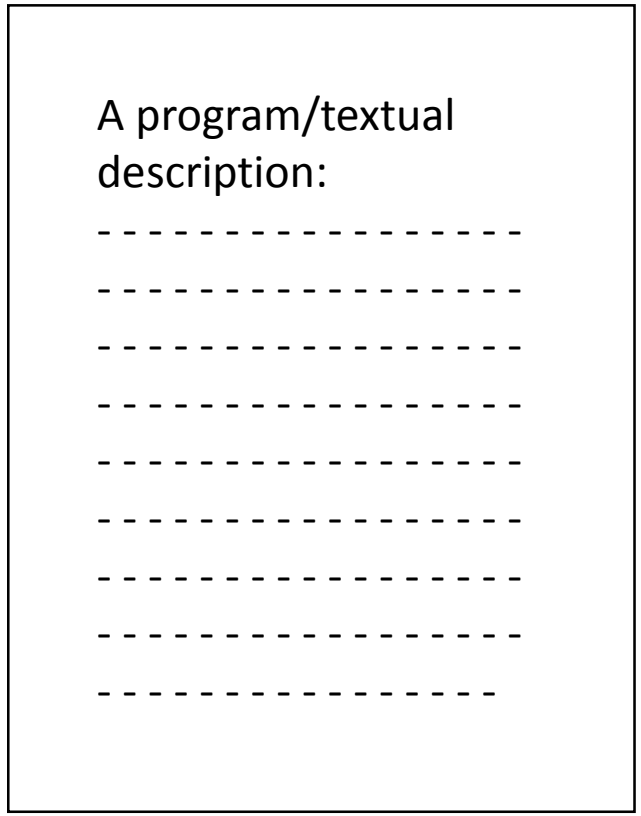
The Modeling of Programming Languages

Daniel Einarson
Kristianstad University
Sweden

ASU Conference 2017

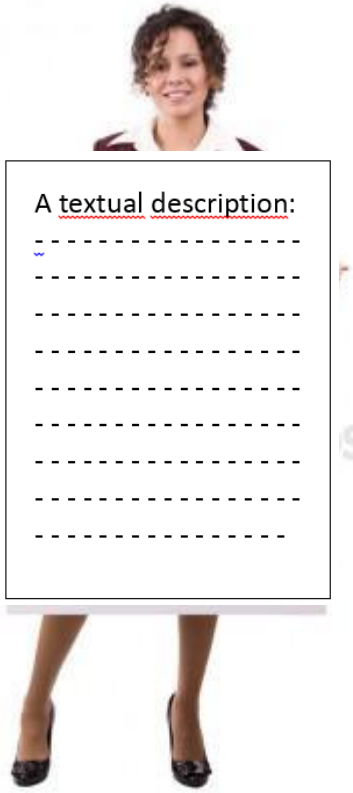
On programming

- As stated by Kristen Nygaard
  - ”To program is to describe”

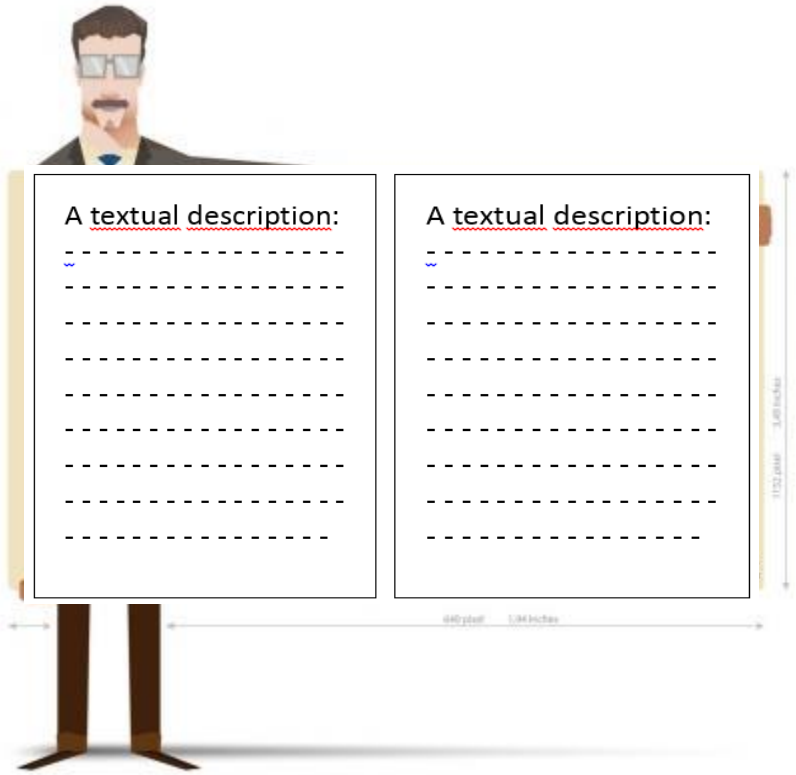


On descriptions, don’t we all describe?

A physicist

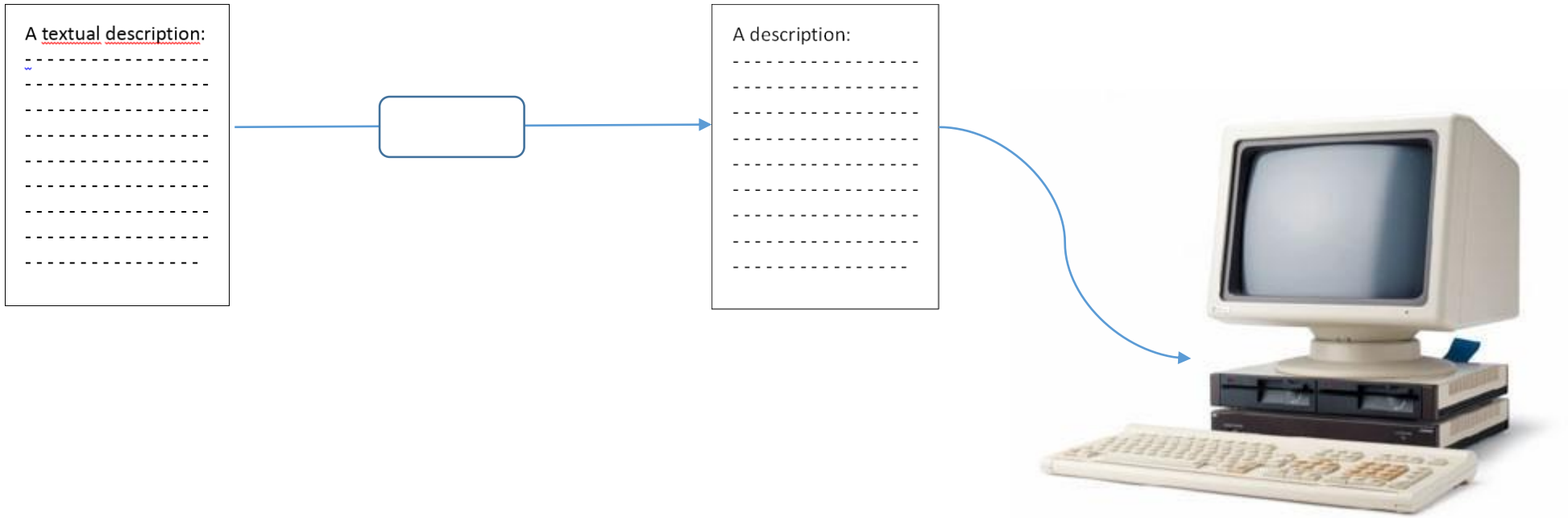


A chemist

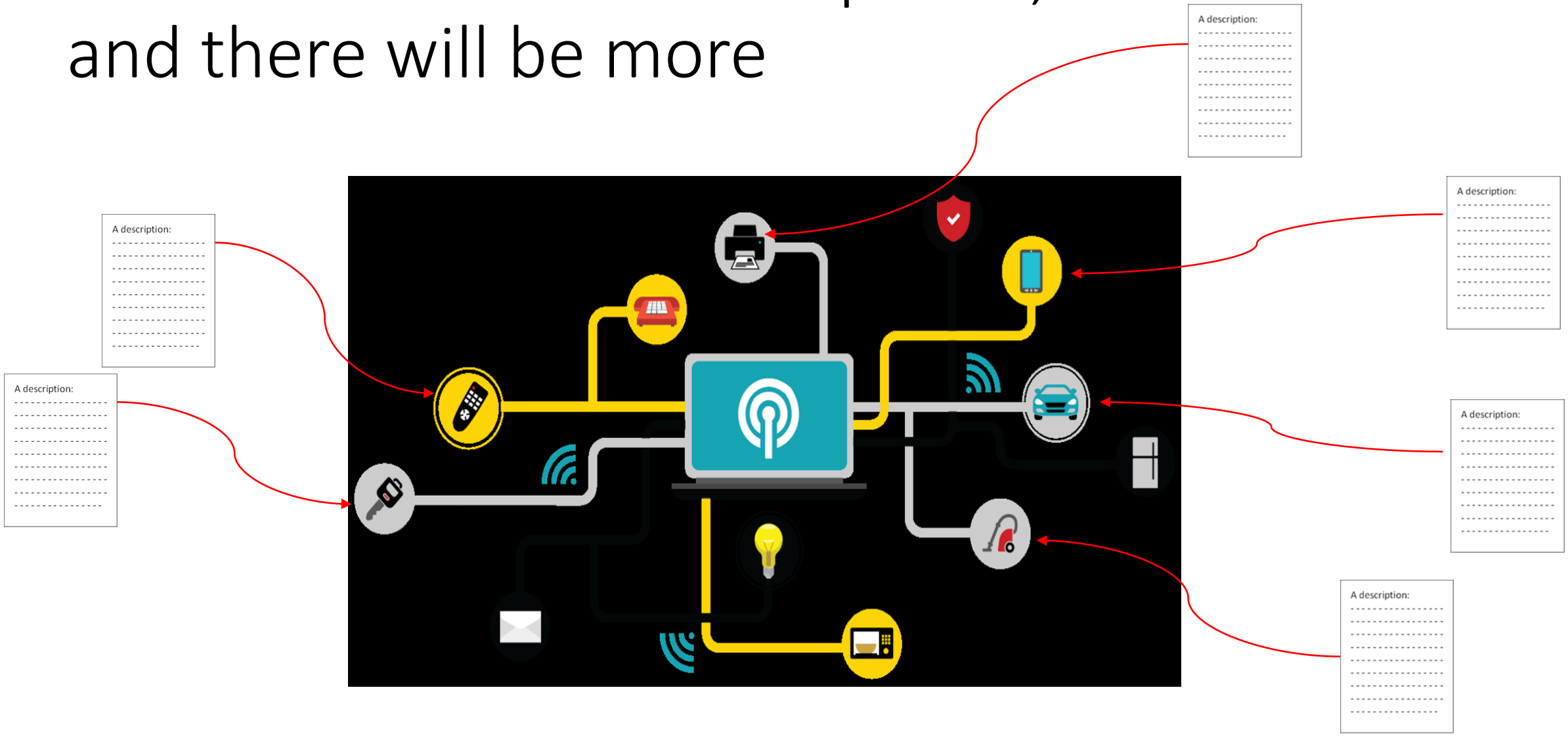


A social scientist

The Computer Science contribution

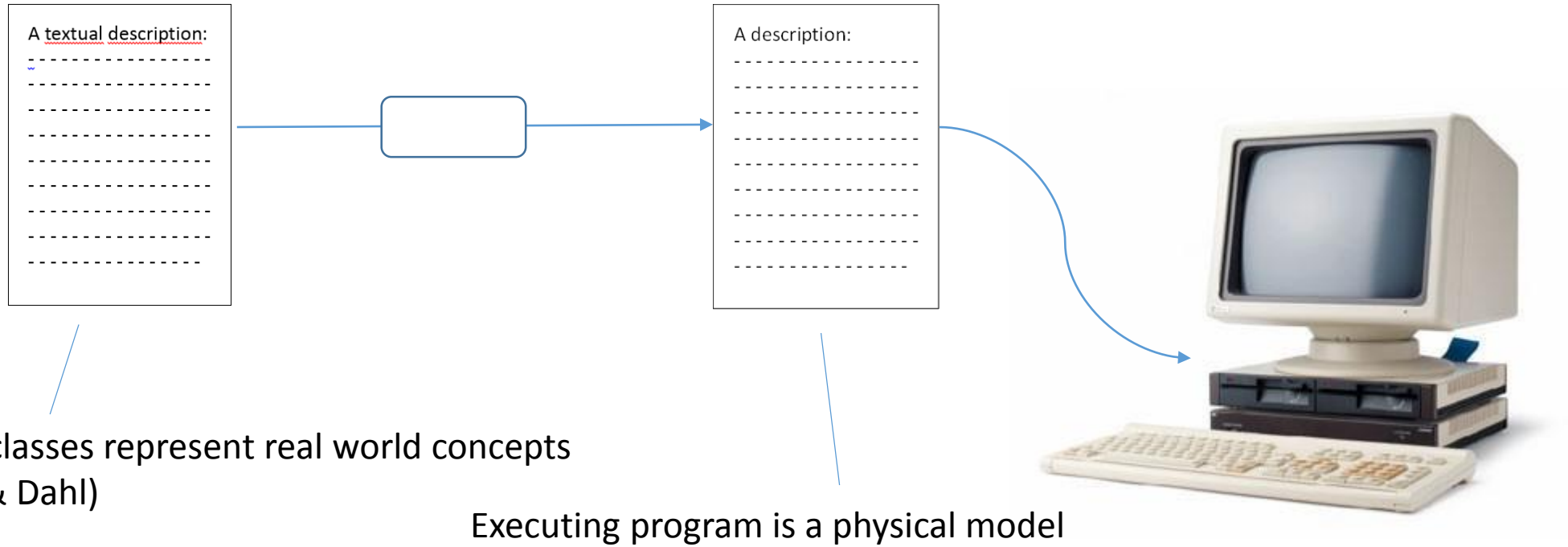


The world is full of descriptions, and there will be more



The Scandinavian School – On Modeling

”has the real world in mind”



Objects, classes represent real world concepts
(Nygard & Dahl)

Executing program is a physical model
simulating the real world behavior
(Madsen & Moller-Pedersen)

Languages for modeling

- Scandinavian School Modeling Languages
  - Simula
  - Beta

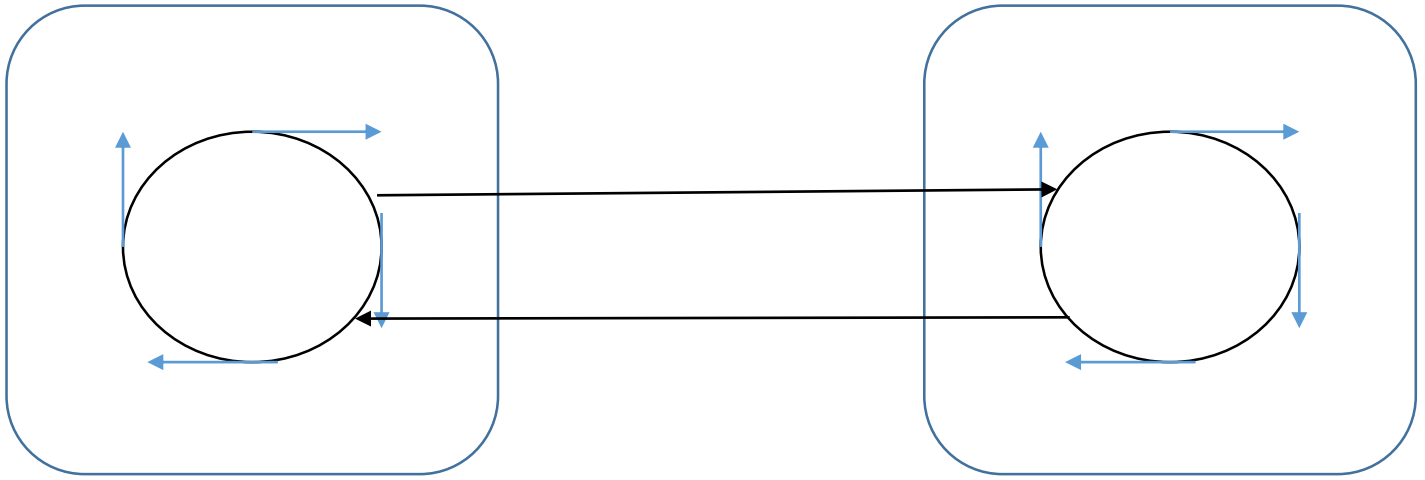
- Modeling language support, with meta-concepts of language to support real world modeling
  - Classes & Objects
  - Generalized vs. Specialized statements
  - Coroutines to simulate parallell actions
  - Nested classes (models in models)

A further elaboration on meta-concepts of modeling languages

- The model-oriented approach, discussed in late 1990 and early 2000
- Refining concepts of autonomous self-contained components
  - The active entity, in contrast to classes/objects that represent passive entities
  - The activity is central - Communication between the active entities are asynchronous (no method/procedure invocations)
  - Abstracted to represent both simulations and real-time
  - Hierarchical modeling, models in models, e.g., simulation models within real-time models

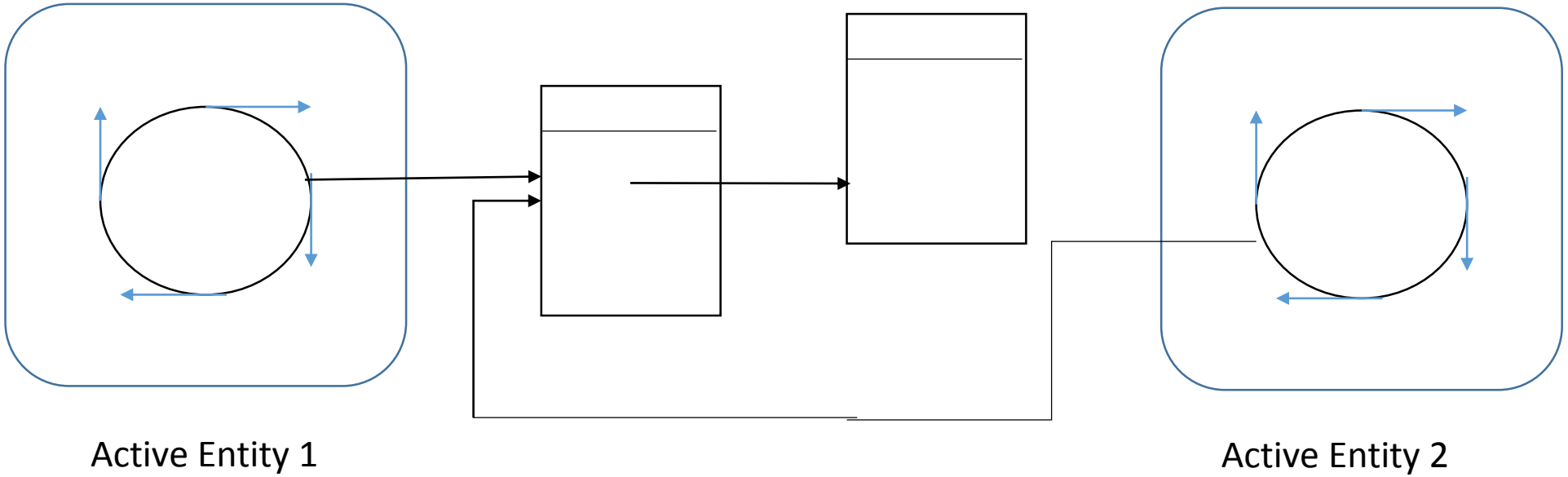
The Active Entity

- The activity runs much like the Simula coroutine



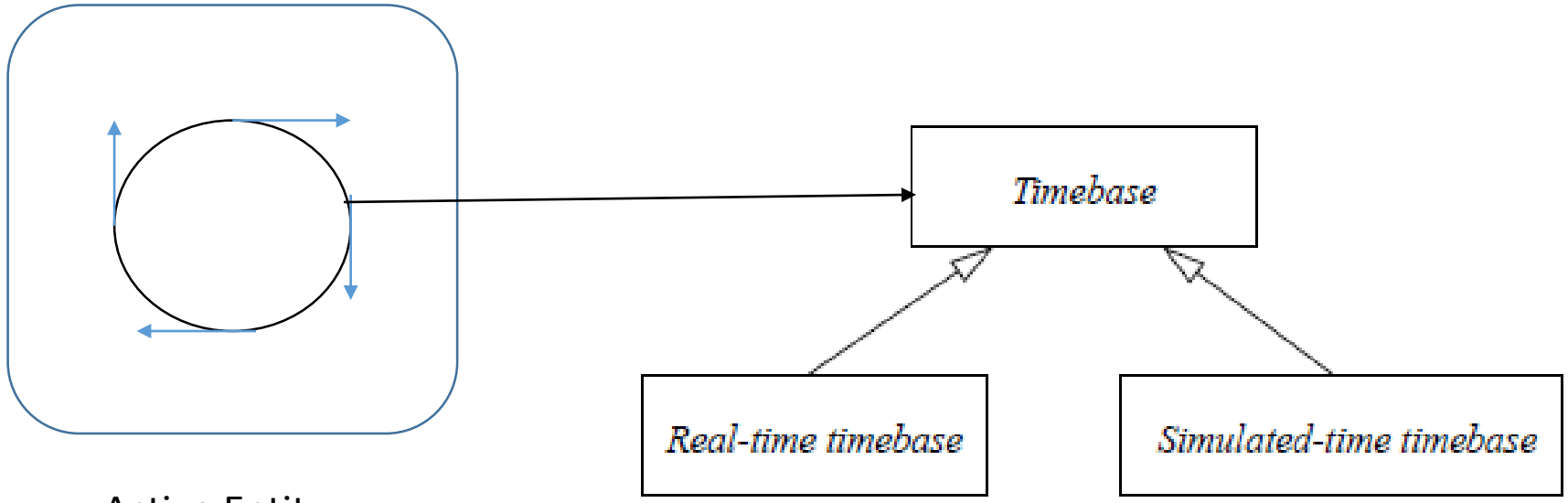
What about the methods then?

- Methods are practicle modeling tools to structure behavior of active entities
- Passive entities/objects, do not do things be themeselves, their methods are, at the end, always invoked be an active entity. Active entities may share similar behavior.



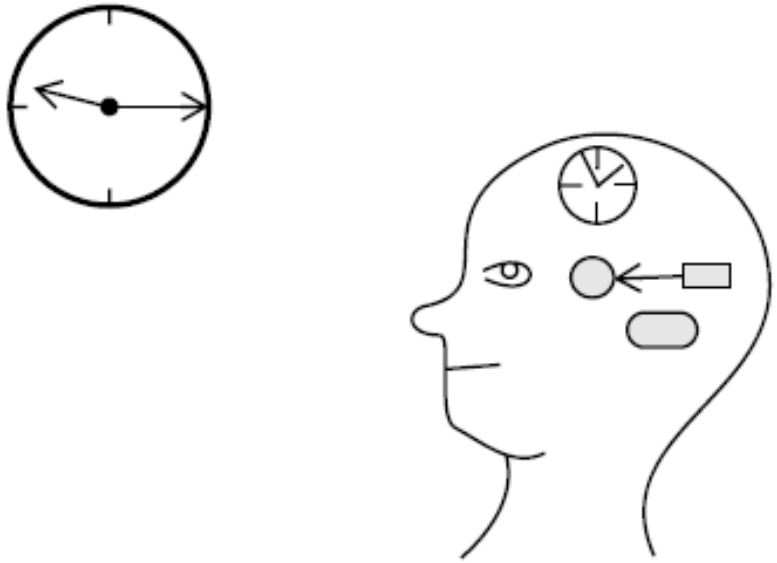
The abstraction of time

- The active entity may act on a real-time or simulated time base
- Timebase is a general concept, specialized into real-time or simulated time timebase



The abstraction of time – real world modeling

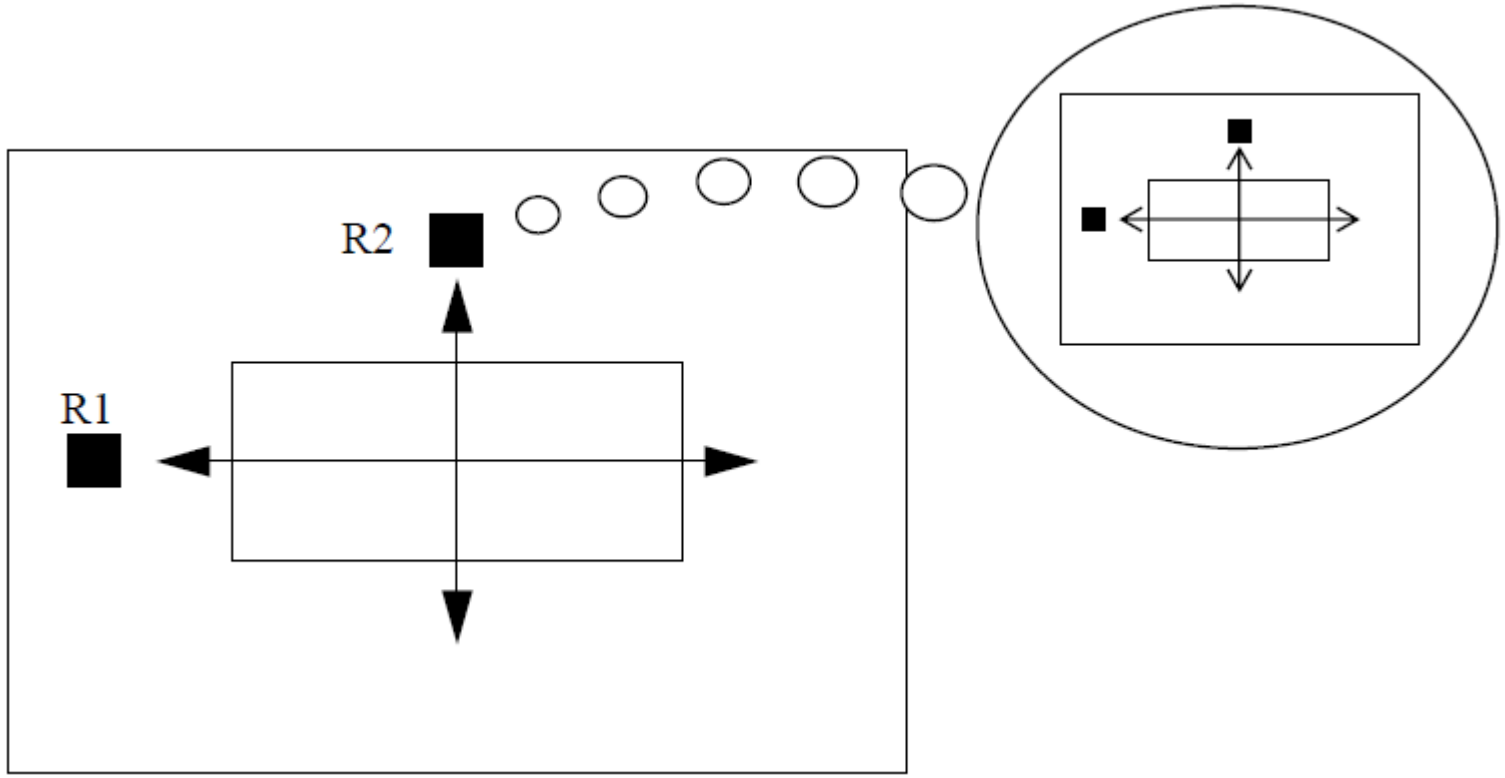
- Clocks at different levels



- is a basis for hierarchical modeling with different time bases

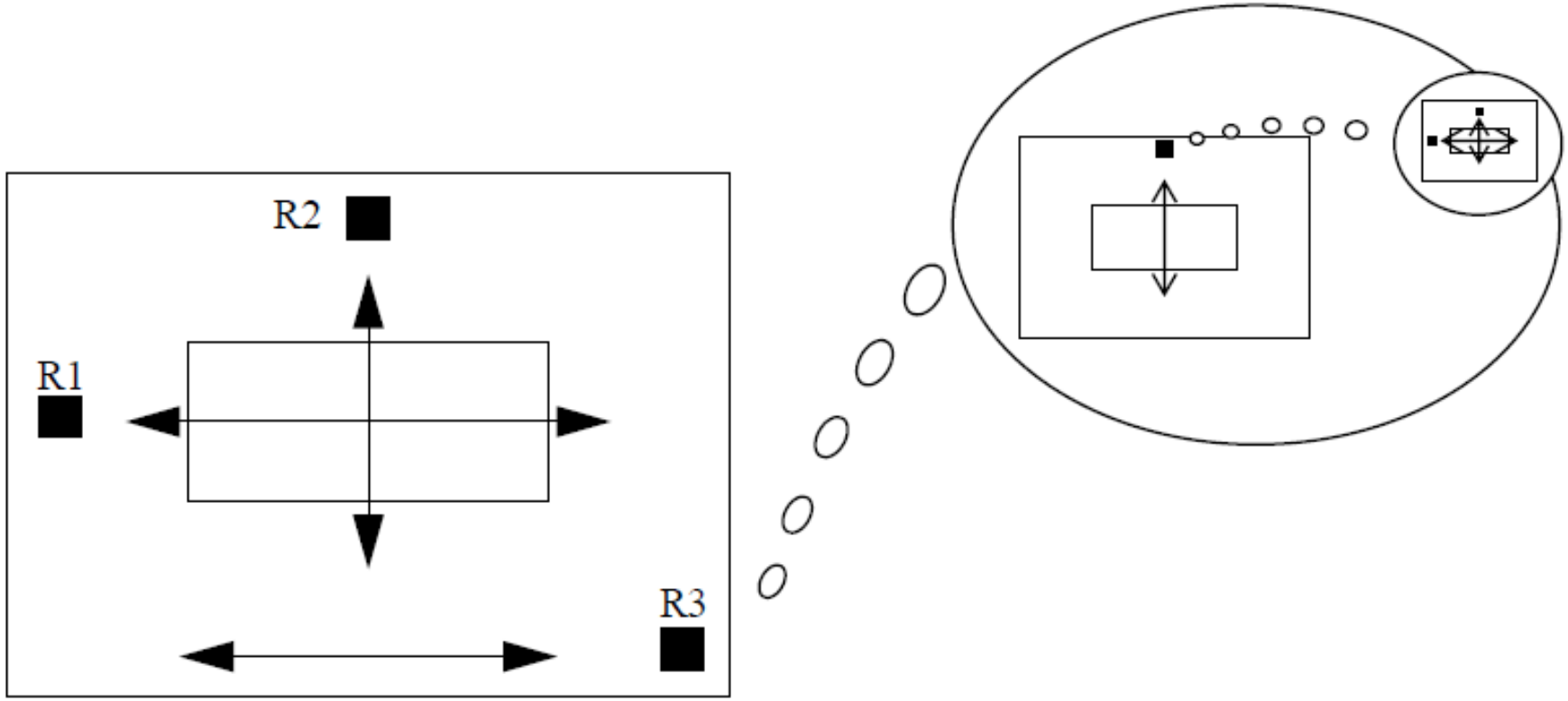
A machine park of robots

- One robot simulates itself and another robot



Or further elaborated

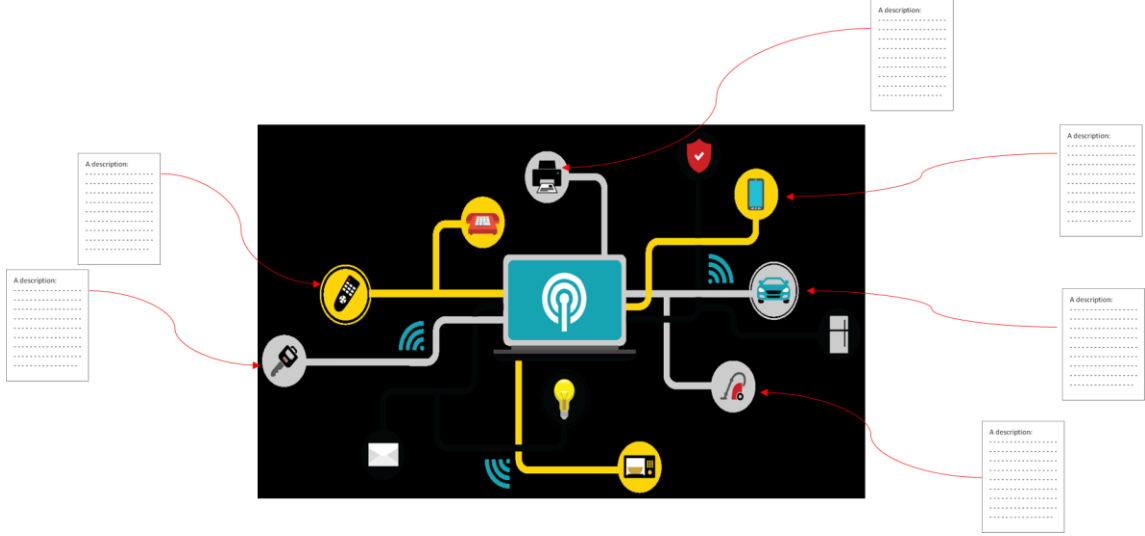
- A third robot simulates one that is simulting another



On the model-oriented (MO)approach

- An extension to the OO paradigm, in the direction of real-world modeling, which is in line with the Scandinavian School.
- A modeling language was developed (not implemented) to meet the MO approach, including meta-concepts for
  - Autonomous active entities
  - Asynchronous communication
  - Abstracted time concepts (as well as specialized)
  - Hierarchical modeling
  - …and more

Finally



- The Scandinavian school has obviously had impact on programming, and OO A&D, for instance UML-based processes addresses that in much
- Still, what about the development of the meta-concepts?
  - Is there a need for new exhaustively developed modeling languages (such as Simula and Beta)?
  - Or a we finished? Do we have all that is desired for?
  - When complexity keeps scaling up, will development keep up with that, or do we have to basically rethink?
- THANX! ☺