

model driven software engineering

Introduction

4DV651

lecture package

introduction to Models

introduction to MDSE

MDSE Examples

understand what a model is

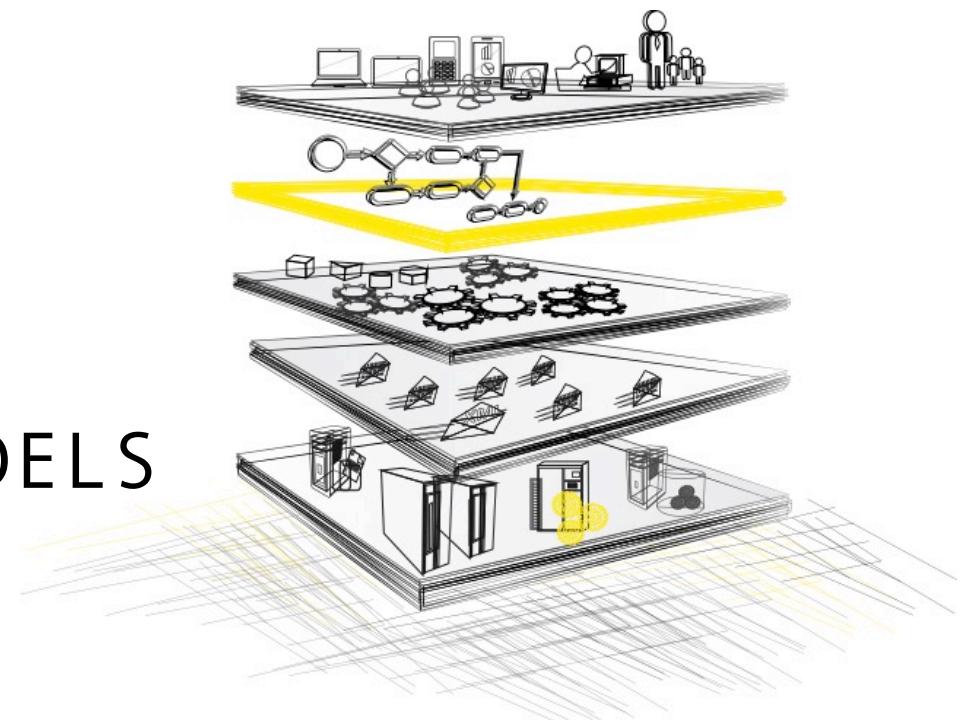
understand the basics of MDSE

model driven software engineering

Introduction - models

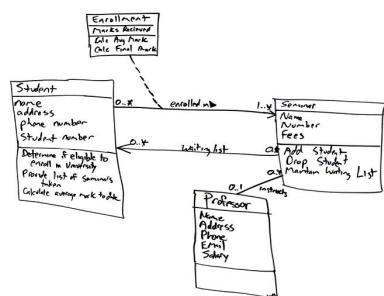
4DV651

SOFTWARE is MODELS

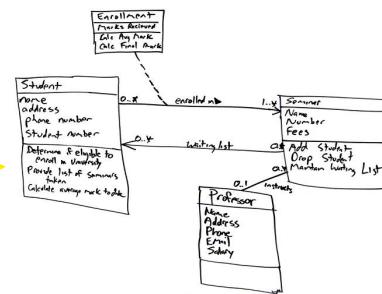


algorithms + data structures = programs

model driven



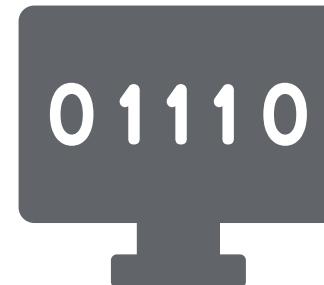
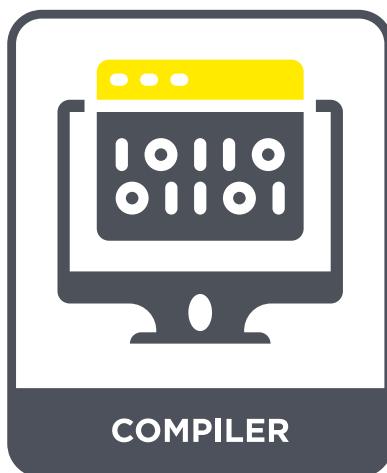
models as sketches



models with semantics

transformations

code



01110 binary

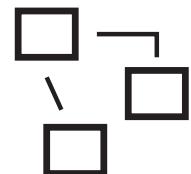
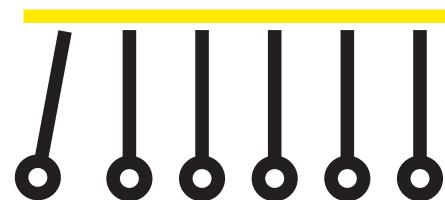


def - a Model

a model is a theoretical construct that represents

- physical,
- biological or
- social processes

with a set of variables and a set of logical and quantitative relationships between them.

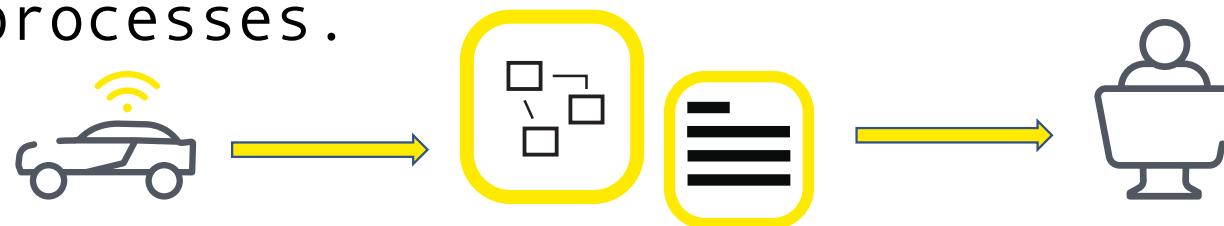


code



models

models are **constructed to enable reasoning** within an idealized logical framework about these processes.

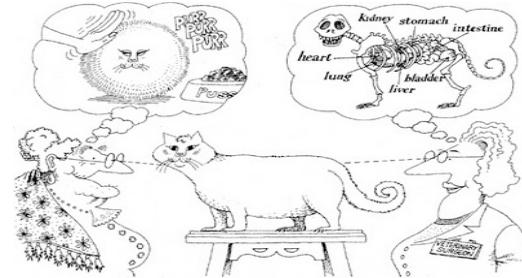


idealized means that the model may make explicit assumptions that are known to be false in some detail → simplifications!.

abstractions

examples

- object
- class
- interfaces
- operation



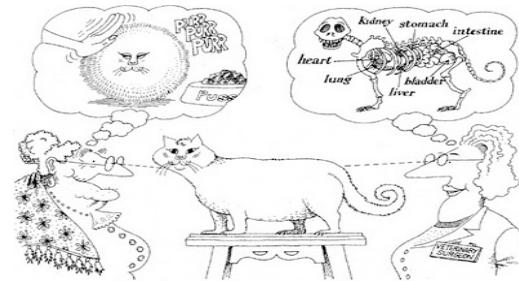
"a *simplified* description, or specification, of a system that *emphasizes some* of the system's *details* or properties *while suppressing others*.

a good abstraction is one that emphasizes details that are significant to the reader or user and suppresses details that are, at least for the moment, immaterial or diversionary." -- Shaw, M. 1984

abstractions

examples

- data
- behavior



"everything in a computer is represented as sequences of "0" and "1"

"this is not an efficient level to work at"

"data abstractions" -> numbers, characters, pictures
"behavior" -> assign, compare, repeat-until

abstractions in models allows for higher-level representations
more efficient and more effective

model - views

all models depict elements and relationships

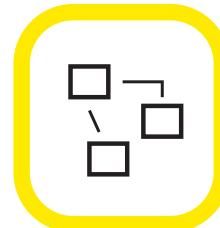
purpose

static - does not depict any change!

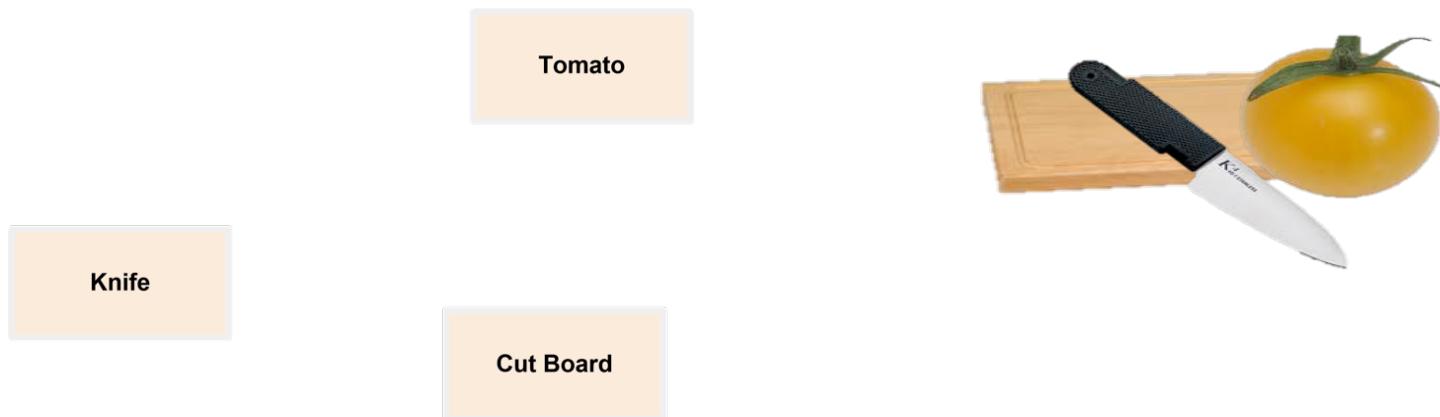
dynamic - illustrates change!

conceptual - a model mainly for reasoning and decisions

physical - a model which models physical - real - entities



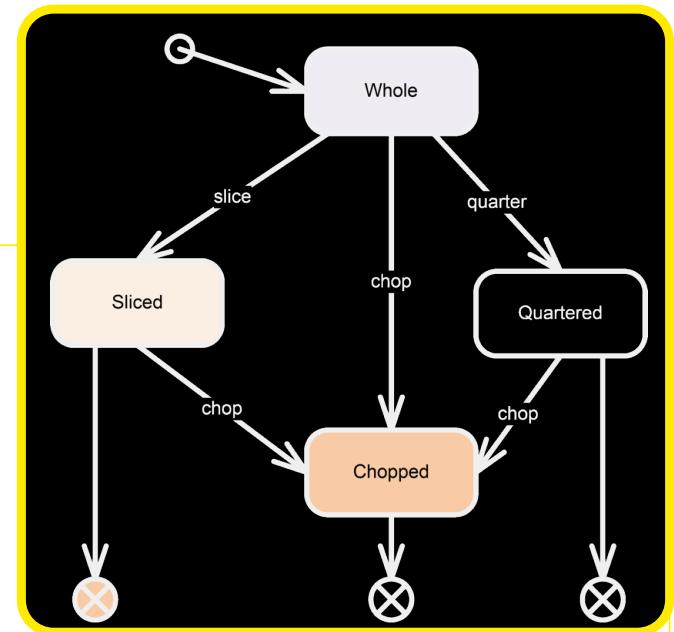
conceptual vs. physical



```
theCutBoard.put(theTomato);  
theKnife.chop(theCutBoard.getItems());
```

static vs. dynamic

```
class Knife {  
  
    private float length;  
    private Manufacturer make;  
    ...  
  
    public void chop() { Collection<IChoppable> objects }  
    public void stab() { IStabbable object ...}  
    public void slice() { ISliceable object ...}  
    ...  
}
```



conceptual - static

Tomato

what do we have!

Knife

which elements

Cut Board

describe a problem

describe a solution?

does not change

conceptual - dynamic

what is happening in a system?

what happens in
a problem
a solution

describes change

