

From Systems to Software

The Digital Machine

Nº 2, *Design of Digital Machines*
Tim Sheiner

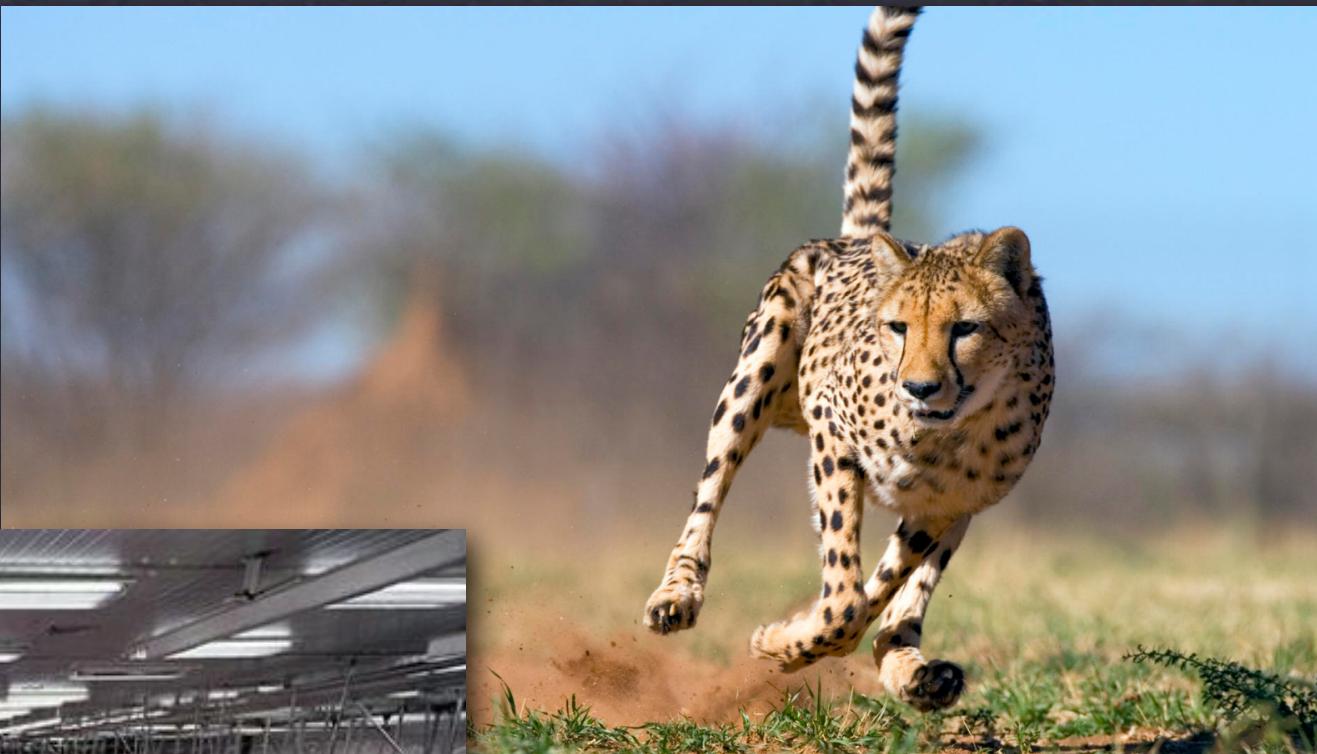
Begin with a definition....

A system is an interconnected set of elements that is coherently organized in a way that achieves something.

Donella Meadows, Thinking in Systems

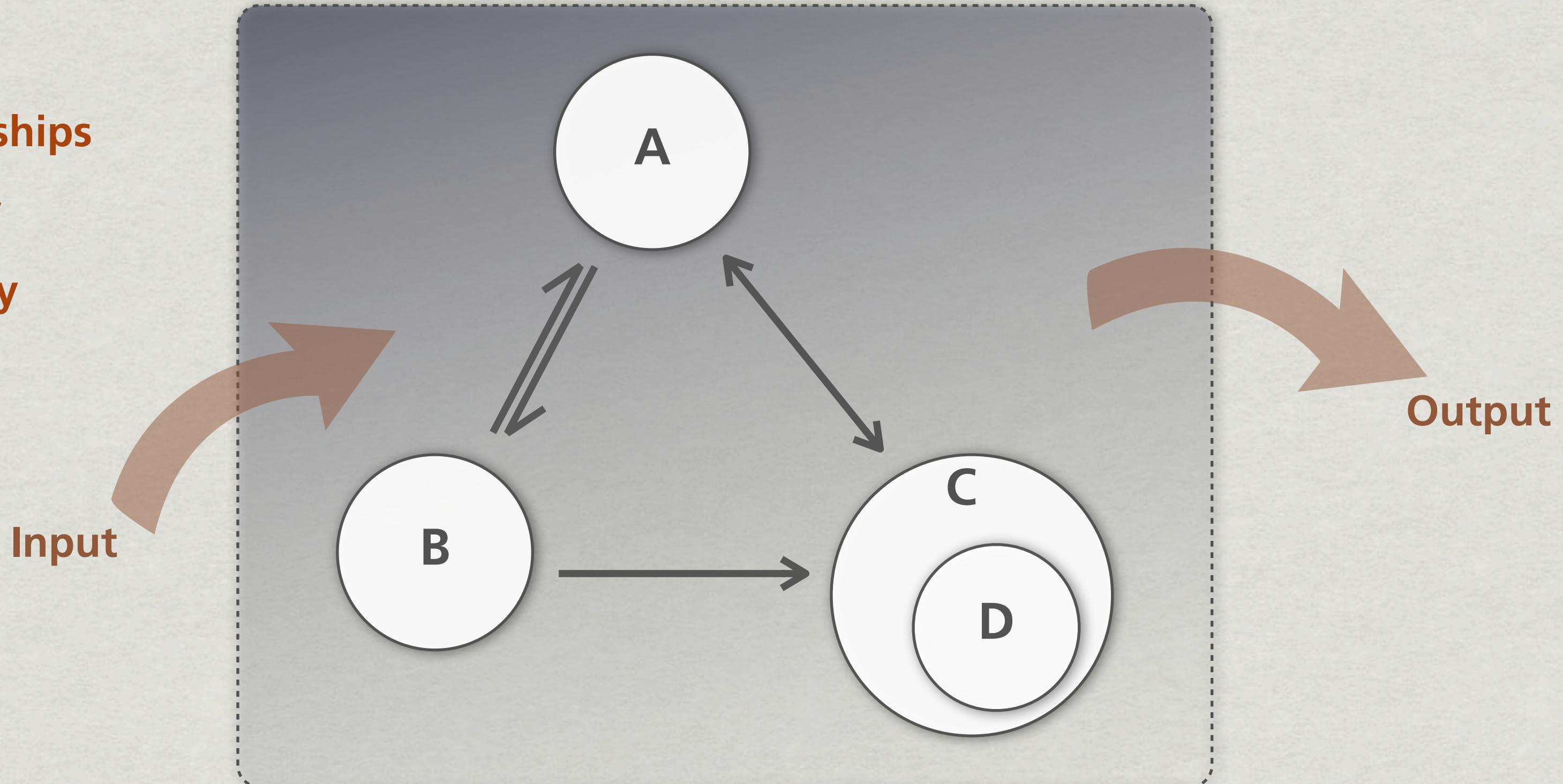
Make an assumption...

These real systems...



...can be abstracted to this form:

1. Objects
2. Relationships
3. Currency
4. Boundary
5. Purpose

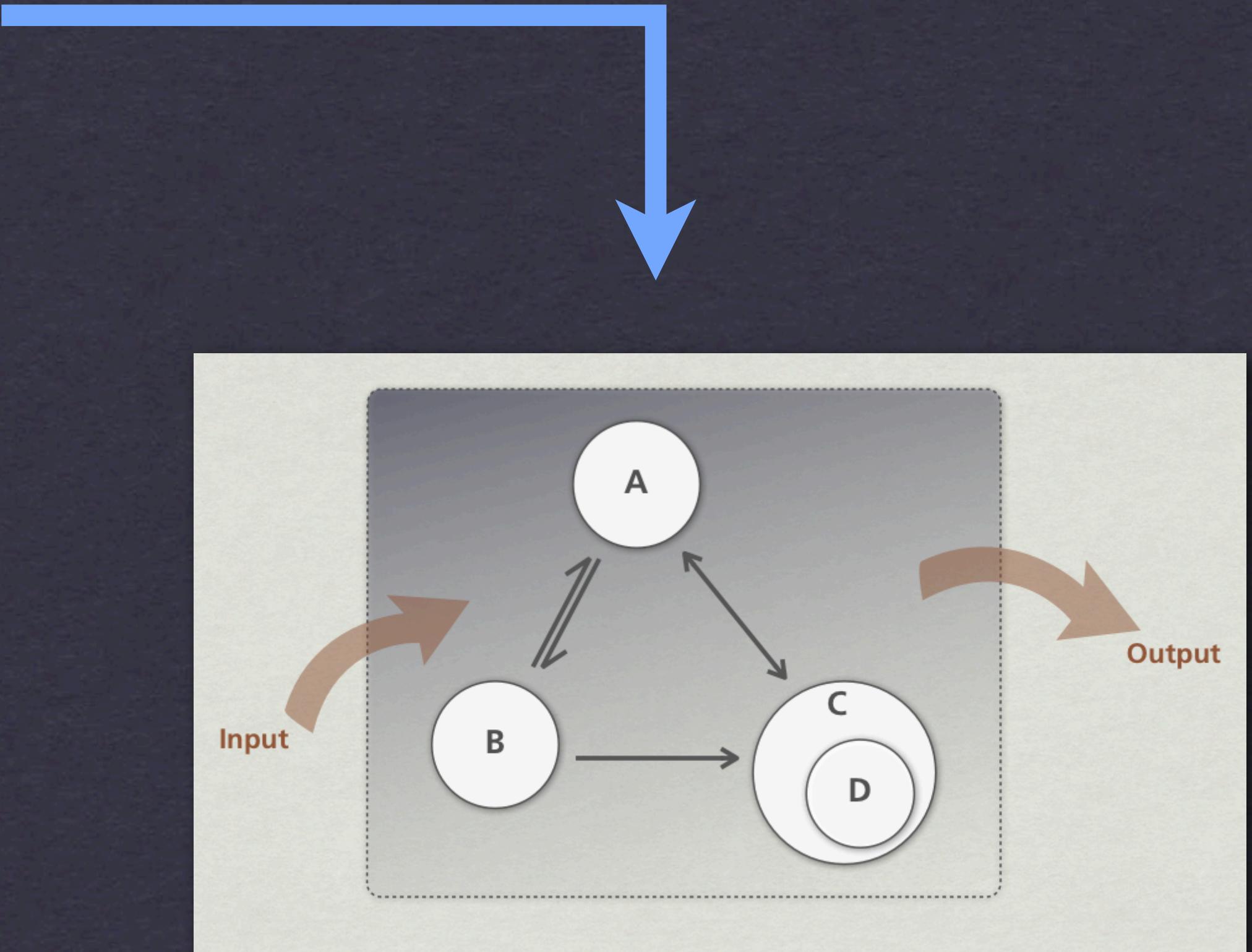


In practice, how is this done?

Step 1: define the abstraction



1. Observe the world
2. Learn the terminology
3. Find the currency
4. See relationships
5. Compare to archetypes
6. Play with the puzzle
7. Settle on simple



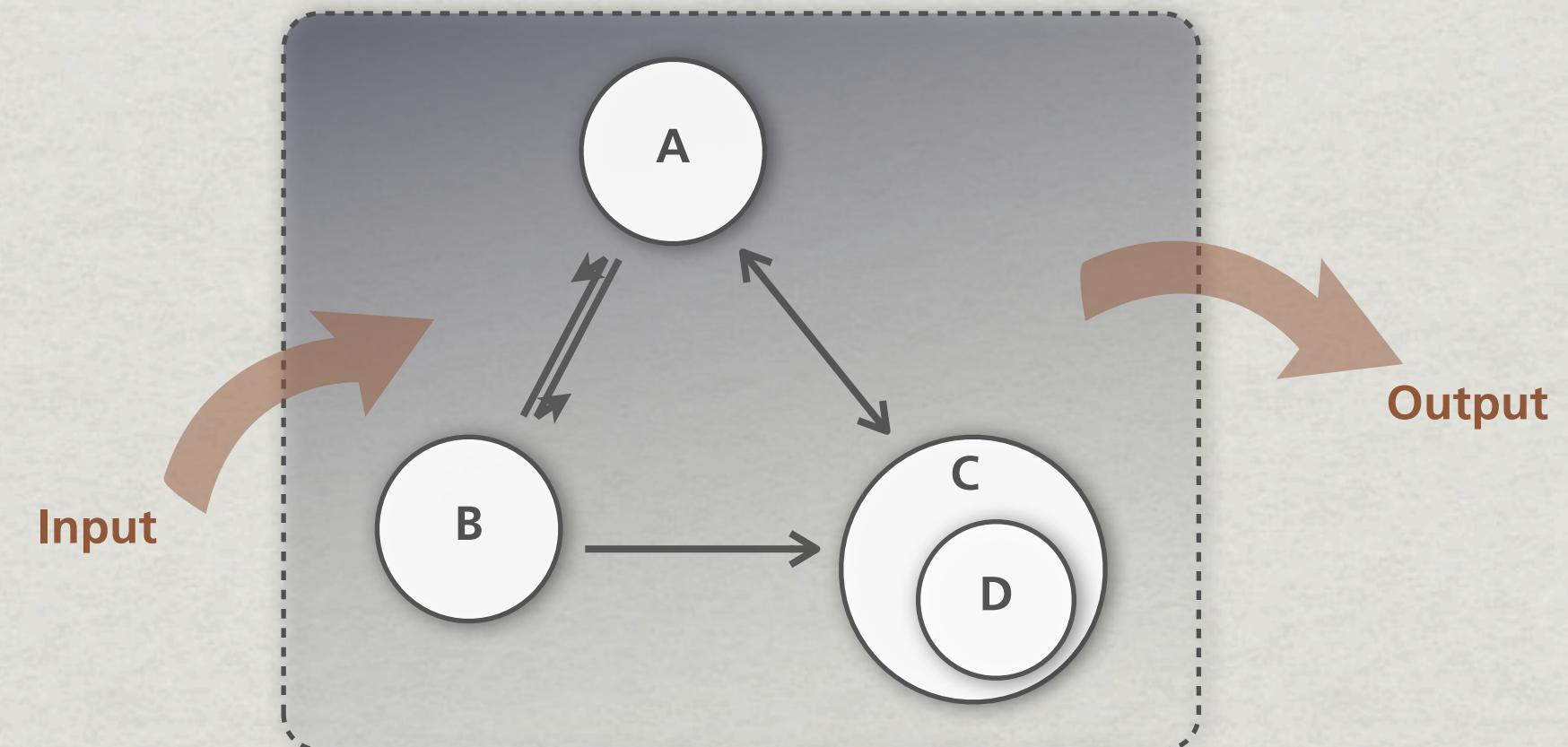
Step 2: abstraction to machine model

Software Machine

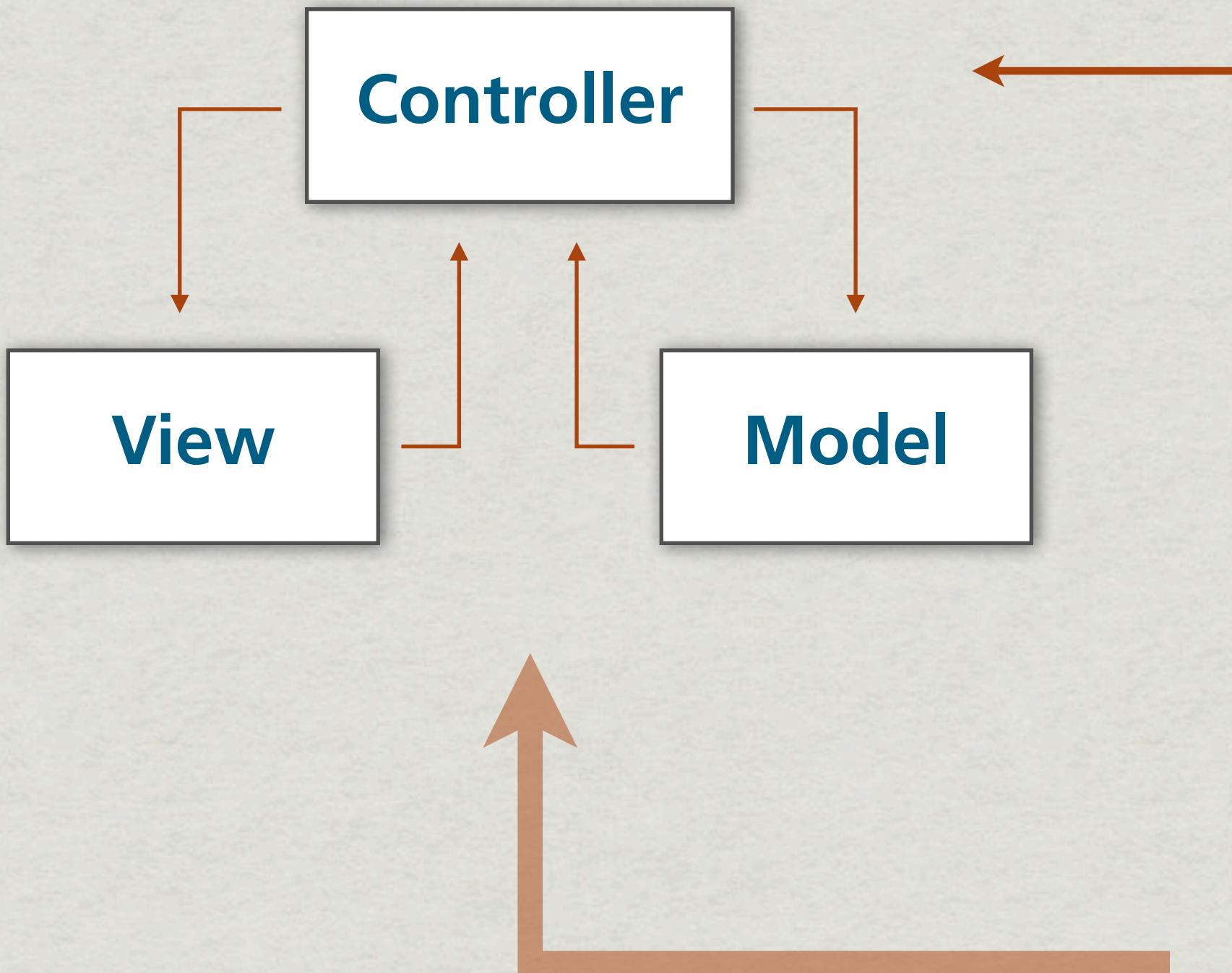


Machine Model

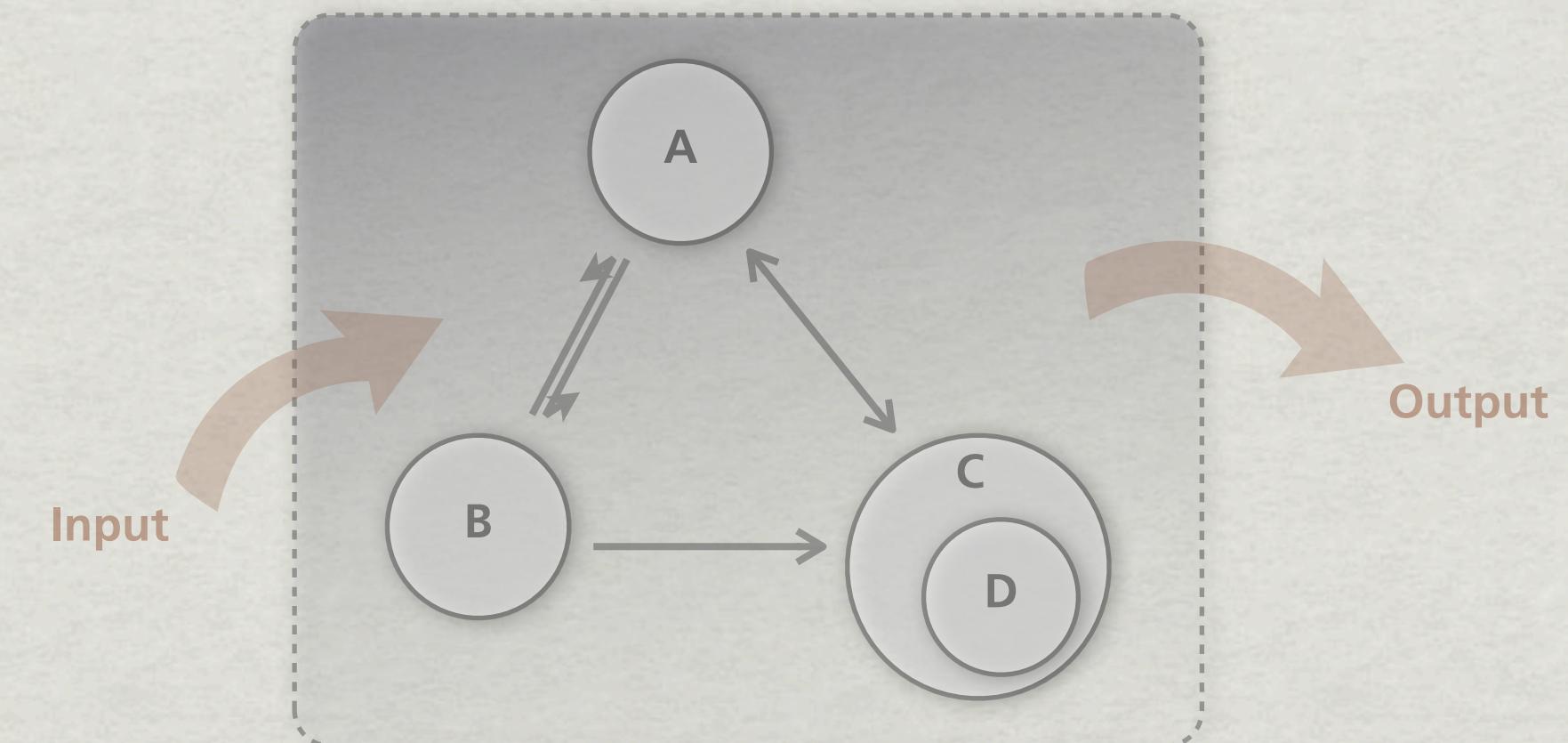
- Conceptual Model
- Object Model
- Interaction Model
- Data Model



The machine has a standard form



This is the digital machine.
All digital machines are of this form.

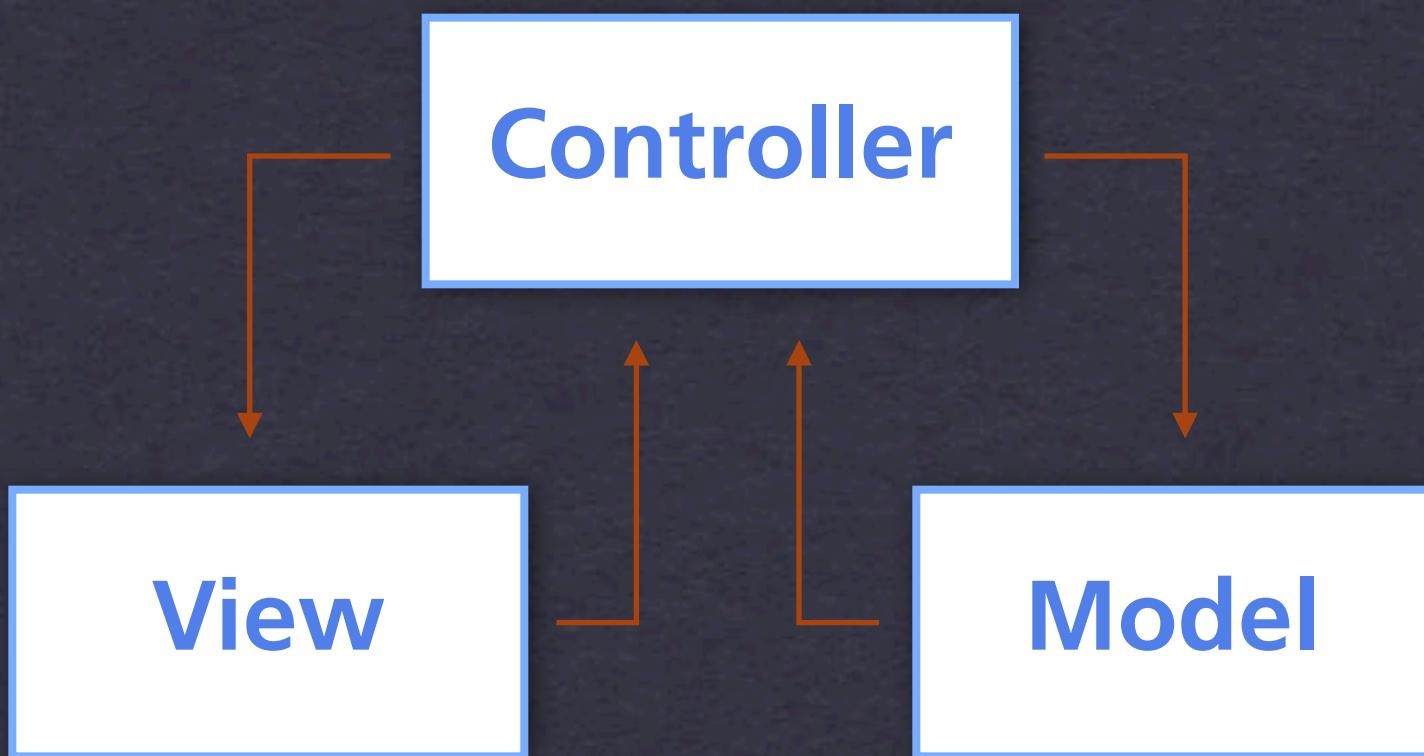


What is the MVC idea?

The idea is separation of function.

The value is

- modularity
- flexibility
- data integrity

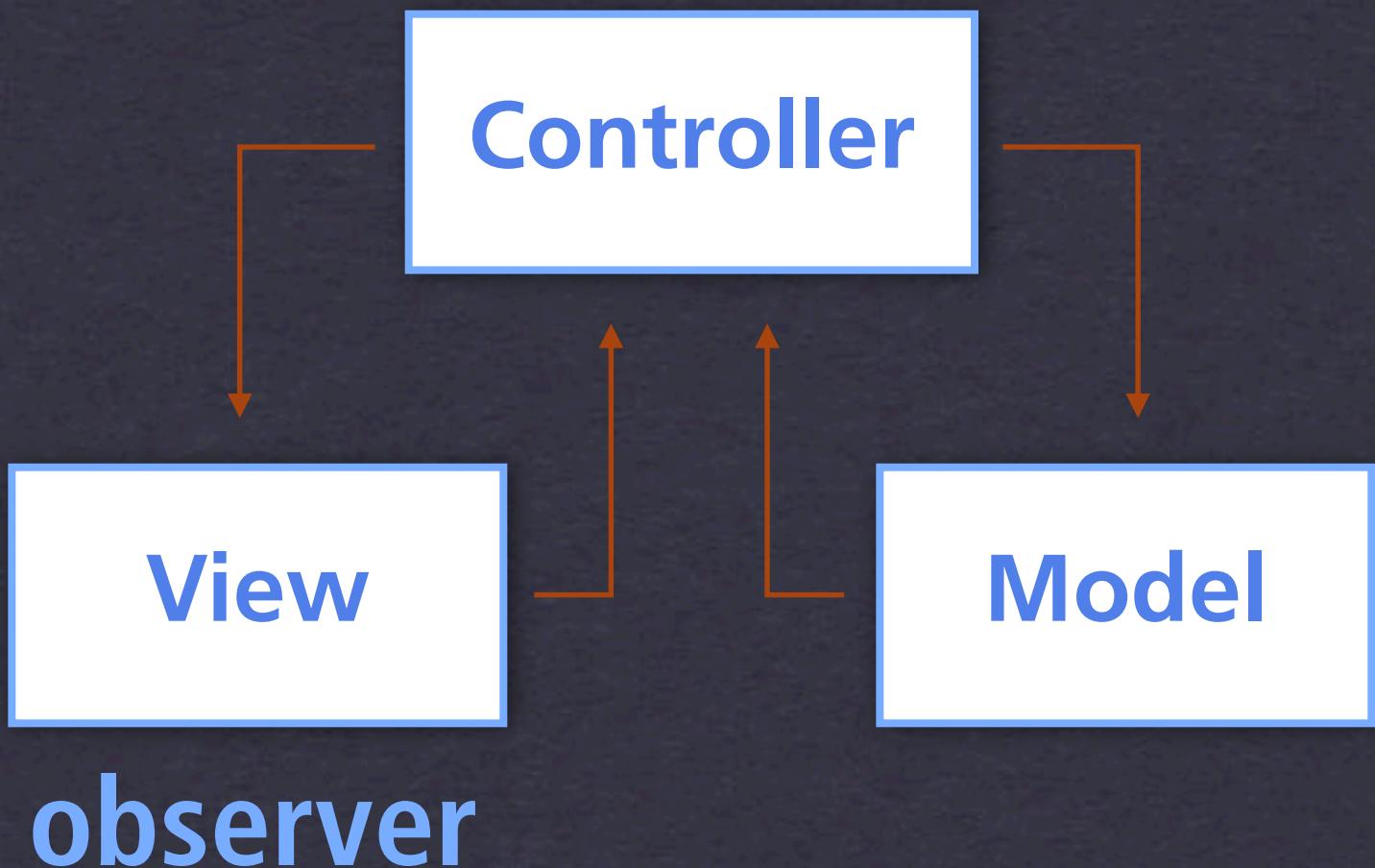


What is the MVC idea?

The idea is separation of function.

The value is

- modularity
- flexibility
- data integrity

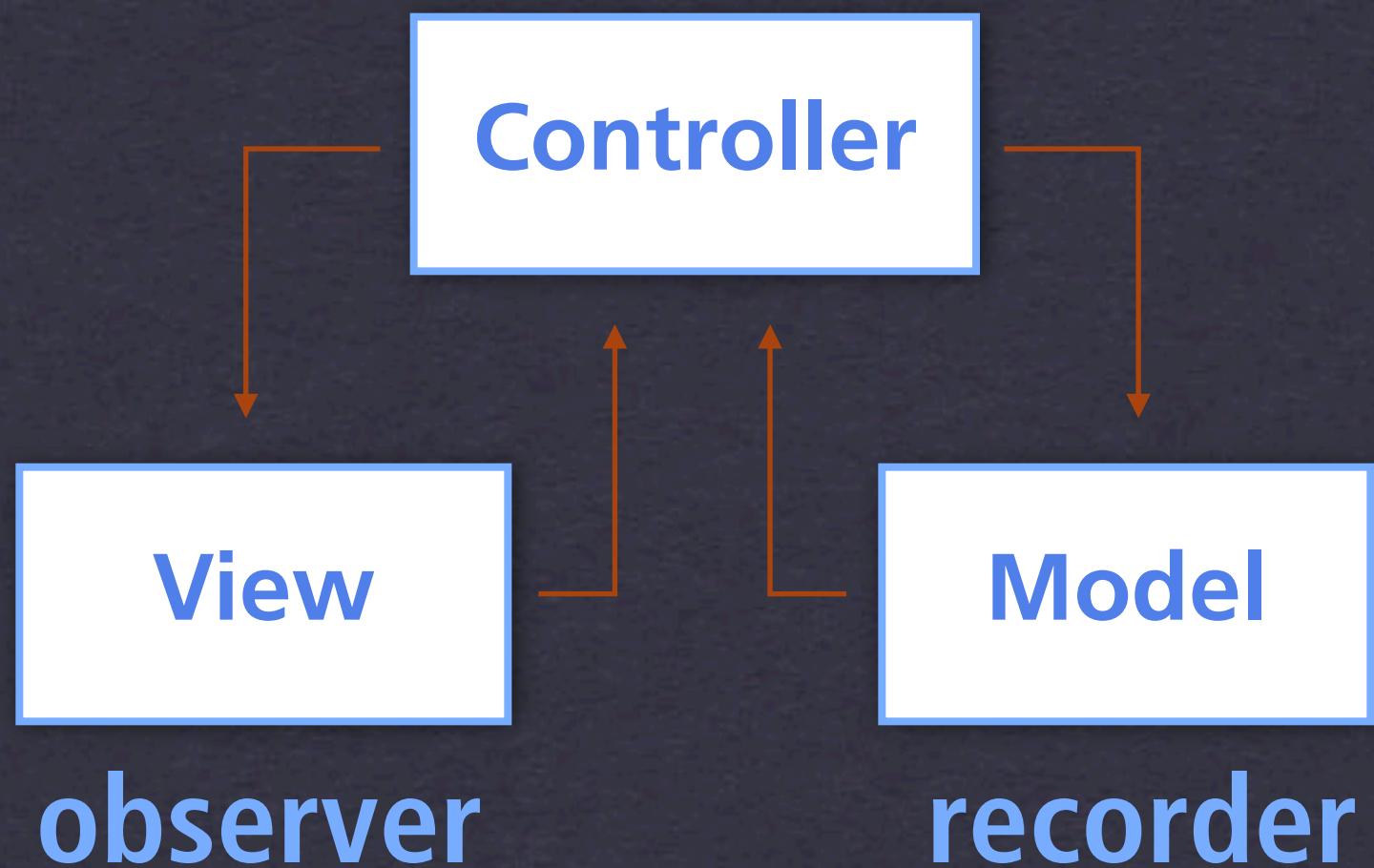


What is the MVC idea?

The idea is separation of function.

The value is

- modularity
- flexibility
- data integrity

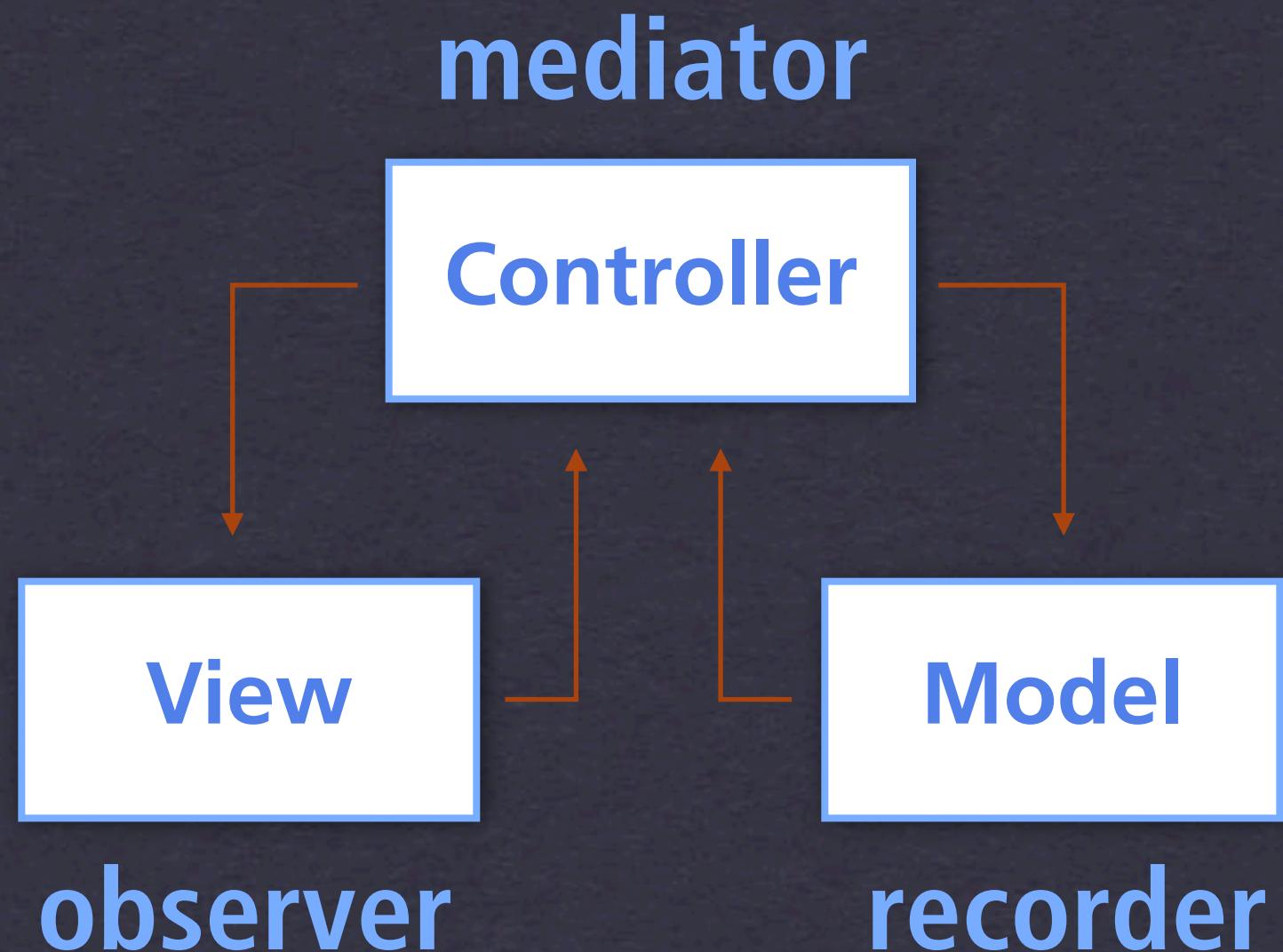


What is the MVC idea?

The idea is separation of function.

The value is

- modularity
- flexibility
- data integrity

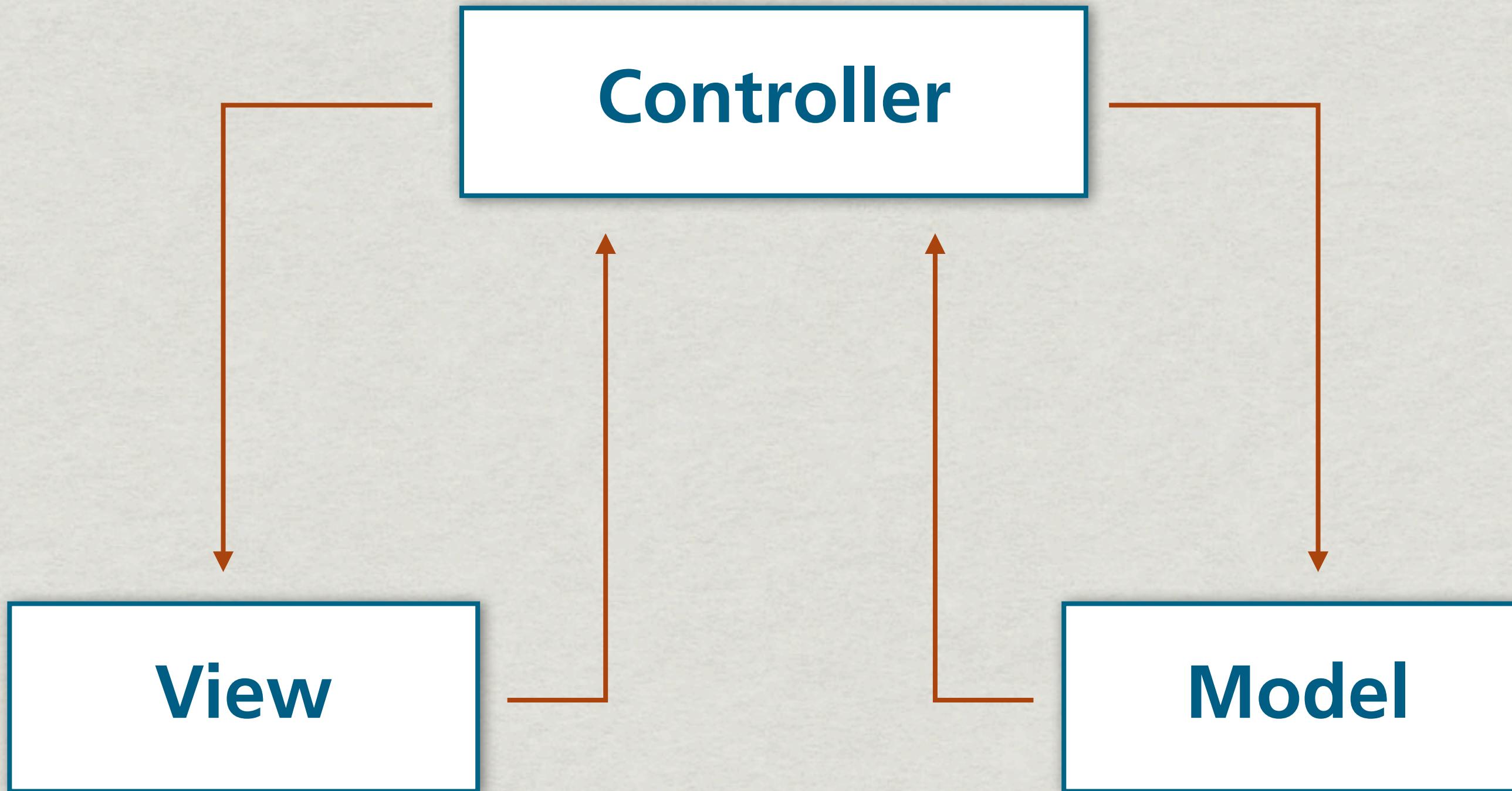


MVC is everywhere in digital machines

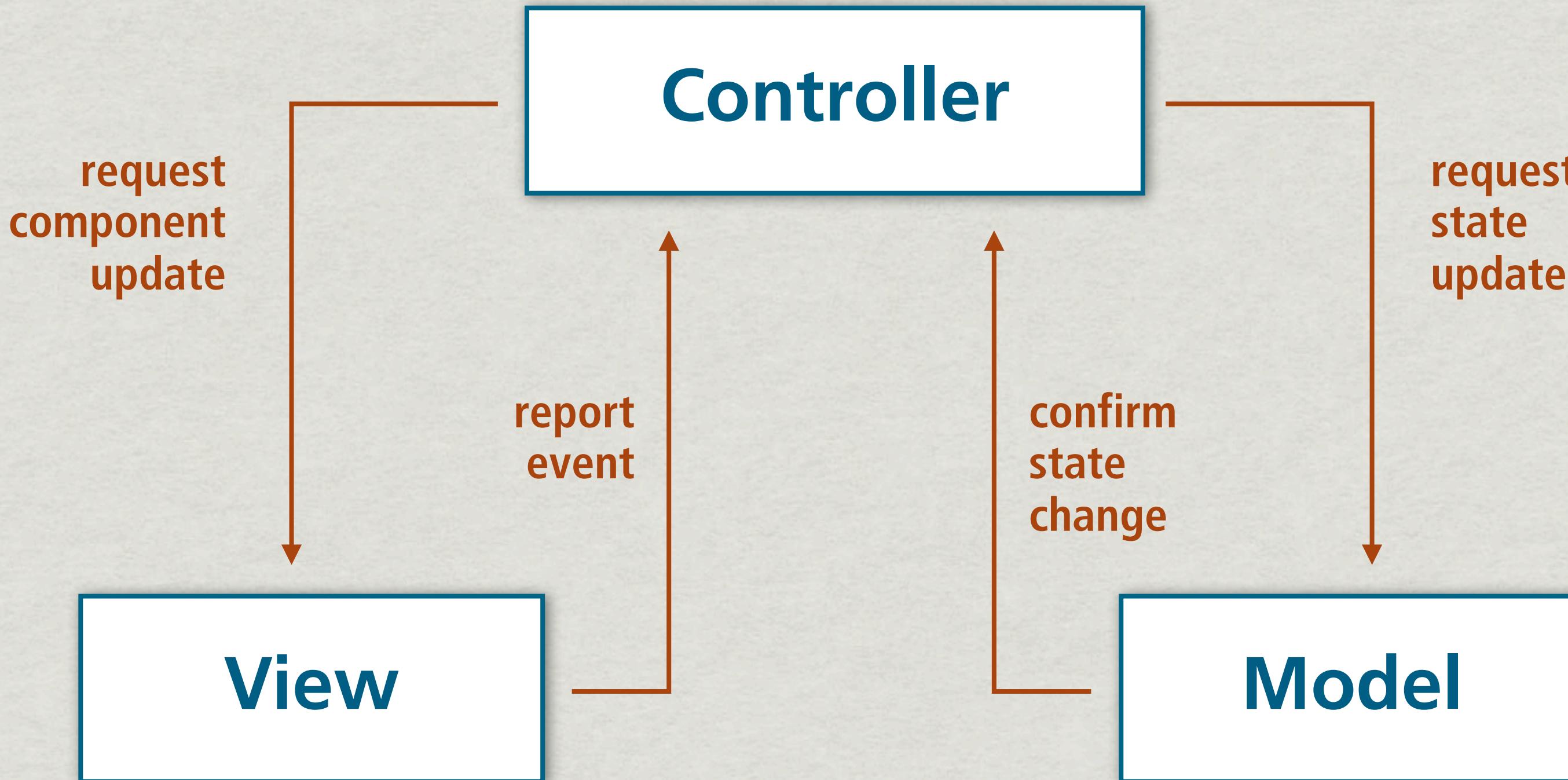
The MVC concept is relevant to the well-designed machine literally, conceptually, tactically, strategically and at every level of analysis.

View	Controller	Model
Show	Do	Save
Objects	Actions	Memory
Observer	Mediator	Recorder
Presentation	Event	Structure
Front End	Middle Tier	Back End
Interface	Application	Database
CSS	Javascript	HTML

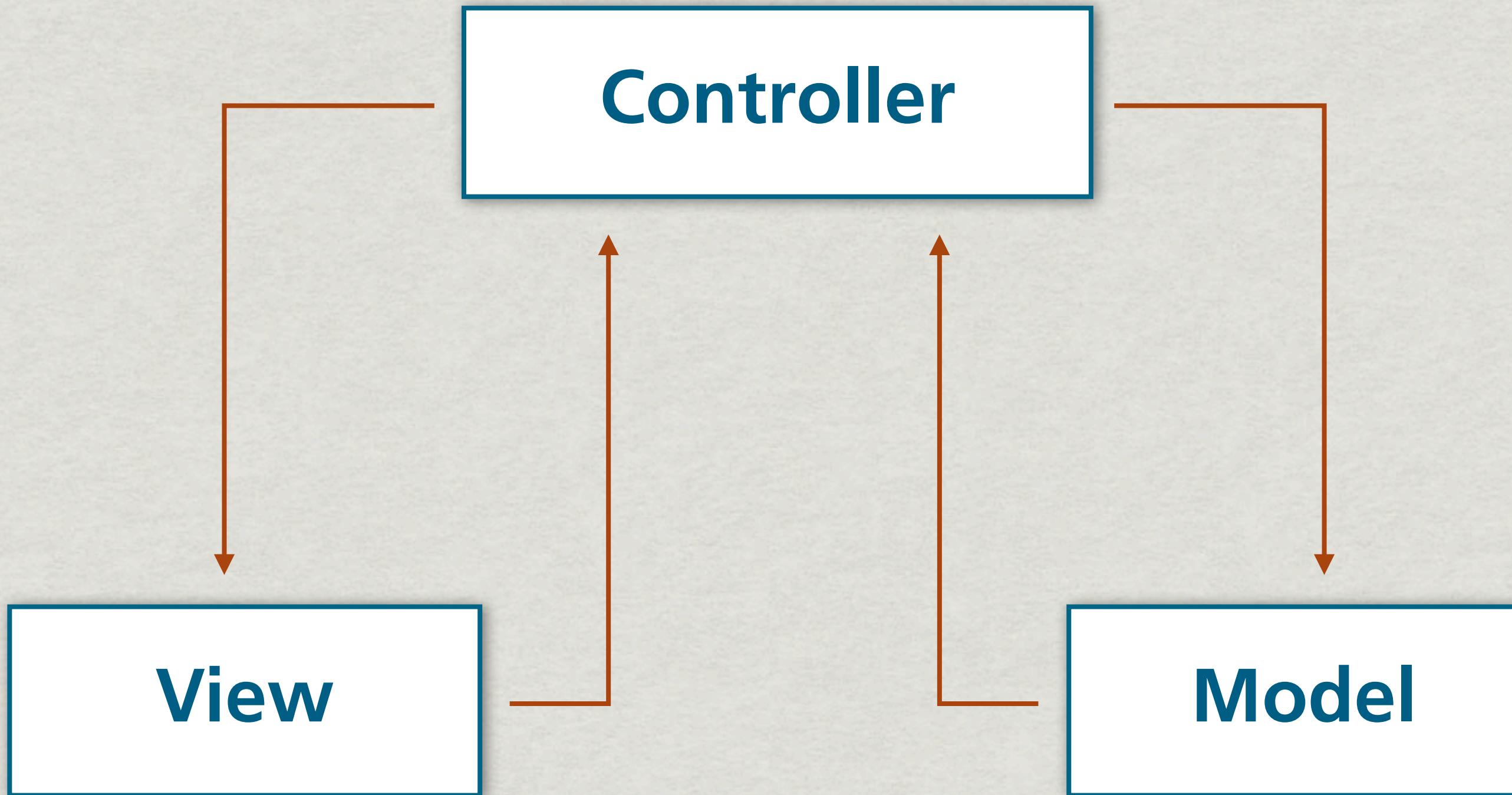
The MVC Cycle



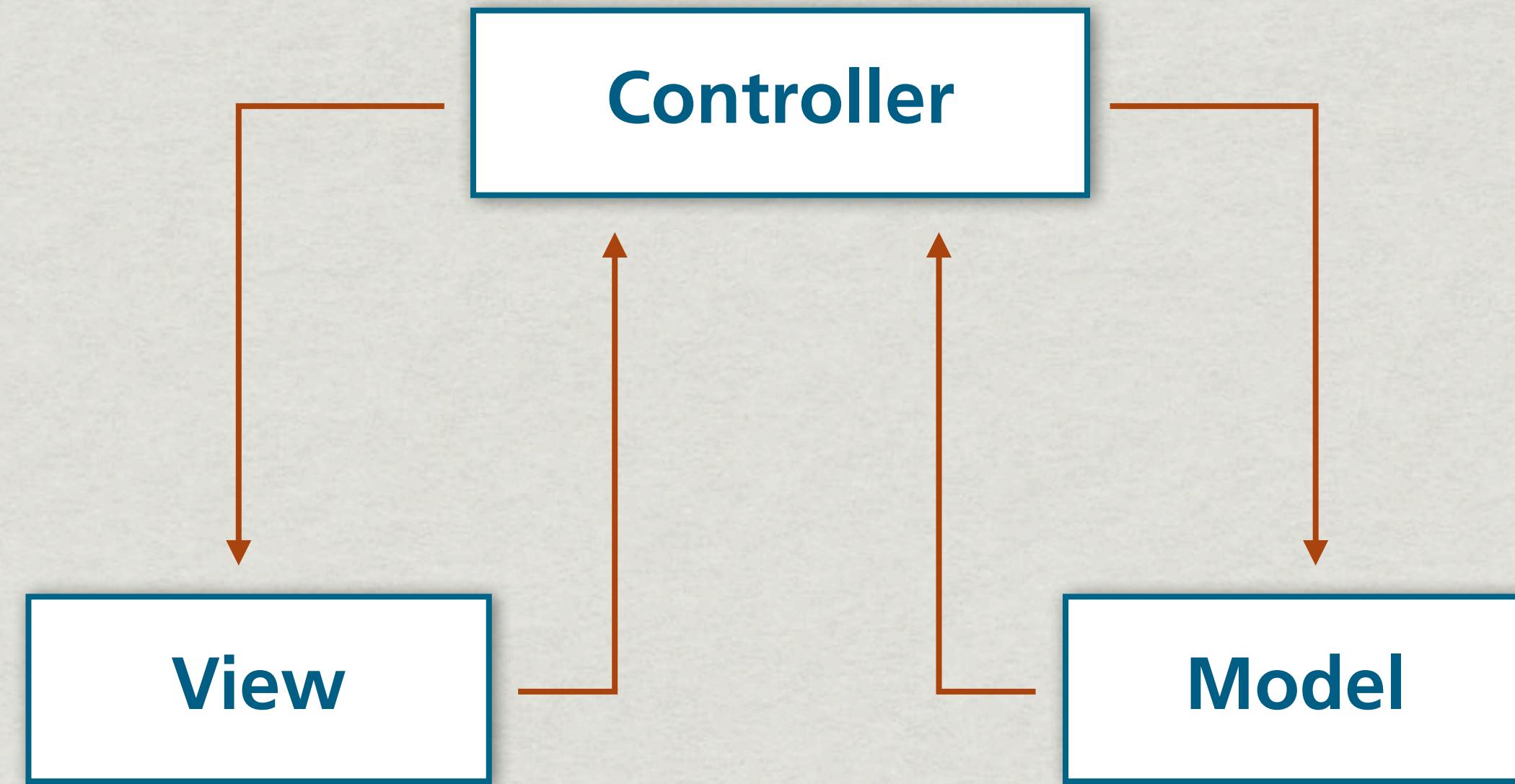
The MVC Cycle



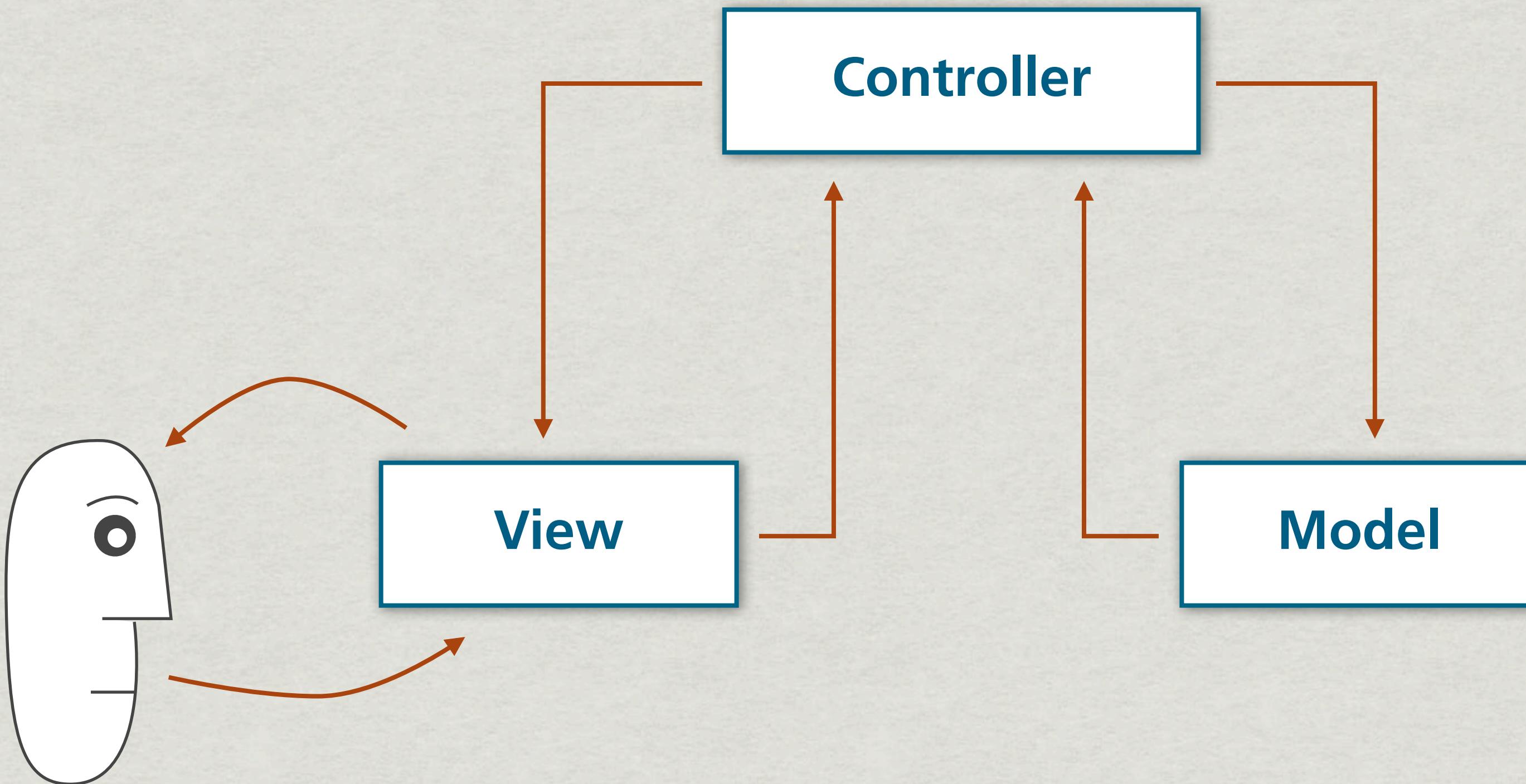
The Design Problem = MVC + User



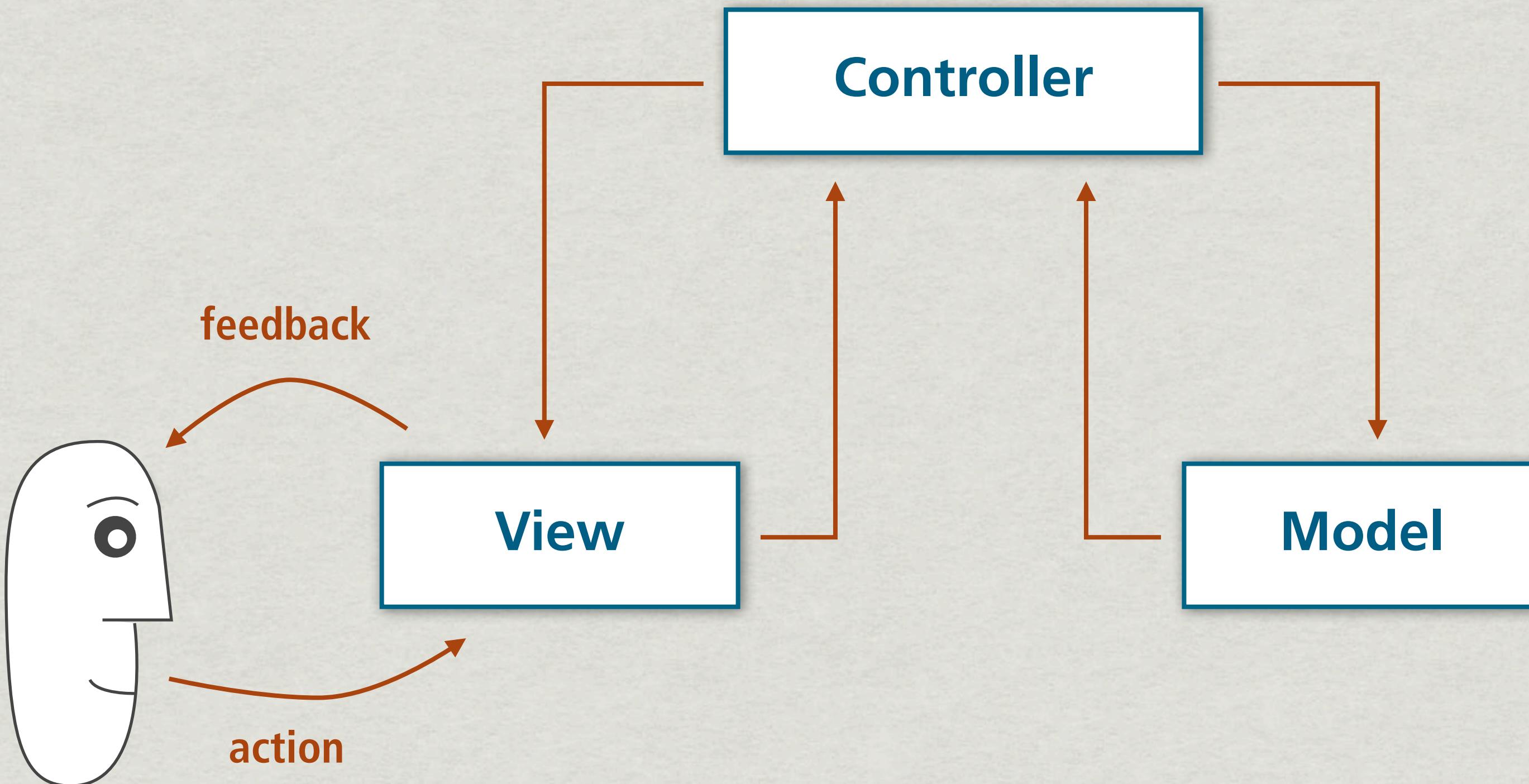
The Design Problem = MVC + User



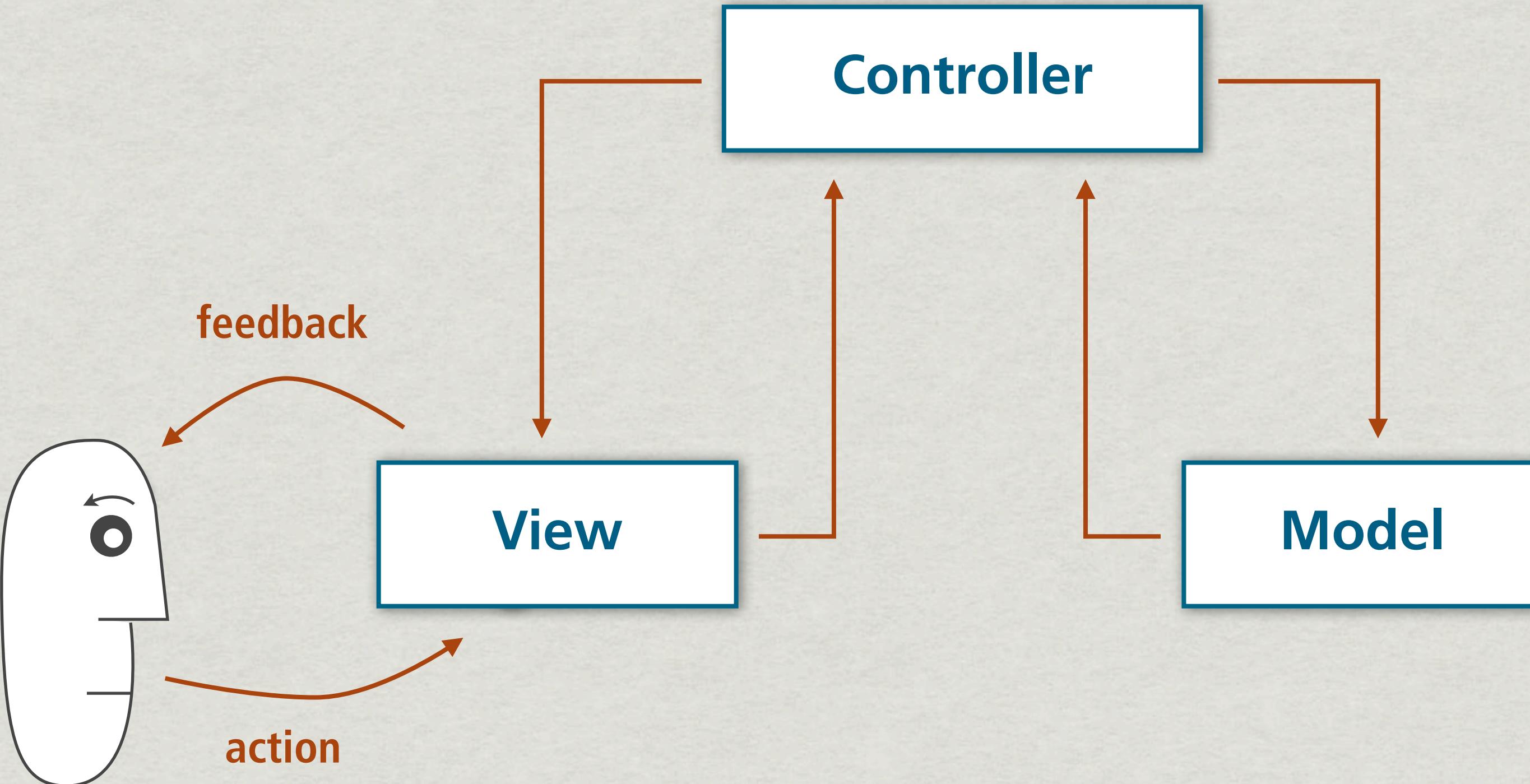
The Design Problem = MVC + User



The Design Problem = MVC + User

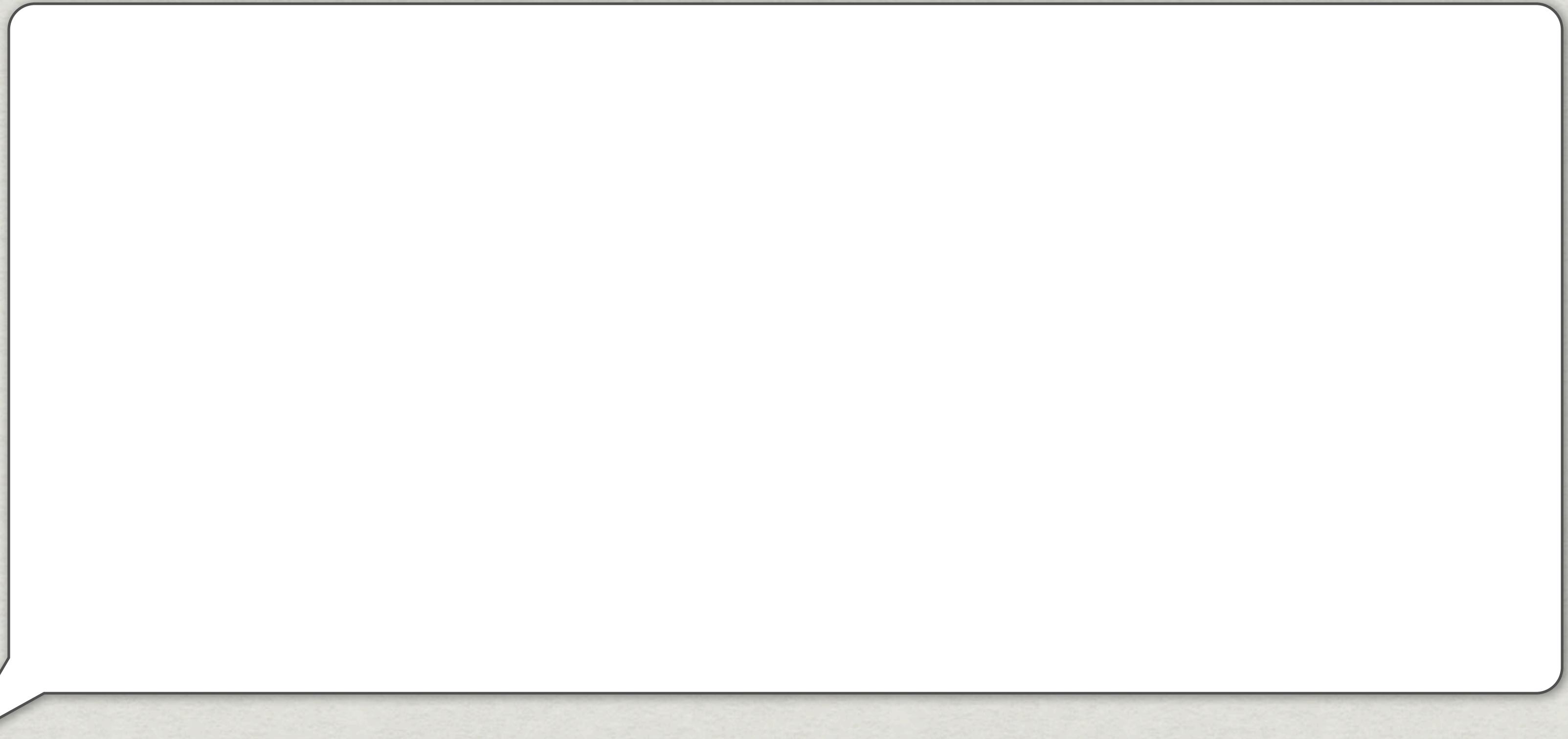
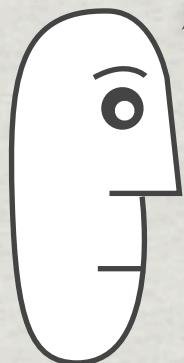


The Design Problem = MVC + User



We use models to design this system.

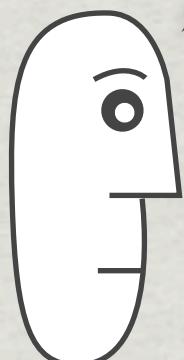
The models answer these questions



The models answer these questions

What does the machine do?

Conceptual Model
Concept



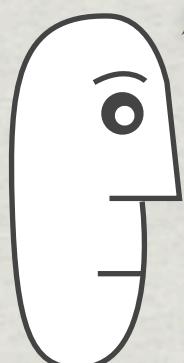
The models answer these questions

What does the machine do?

Conceptual Model
Concept

How do I change the output?

Interaction Model
Flow



The models answer these questions

What does the machine do?

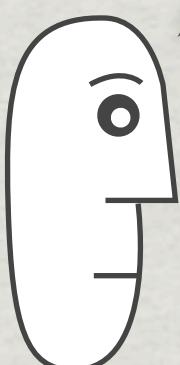
Conceptual Model
Concept

How do I change the output?

Interaction Model
Flow

How does it work?

Object Model
Structure



The models answer these questions

What does the machine do?

Conceptual Model
Concept

How do I change the output?

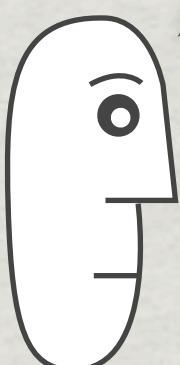
Interaction Model
Flow

How does it work?

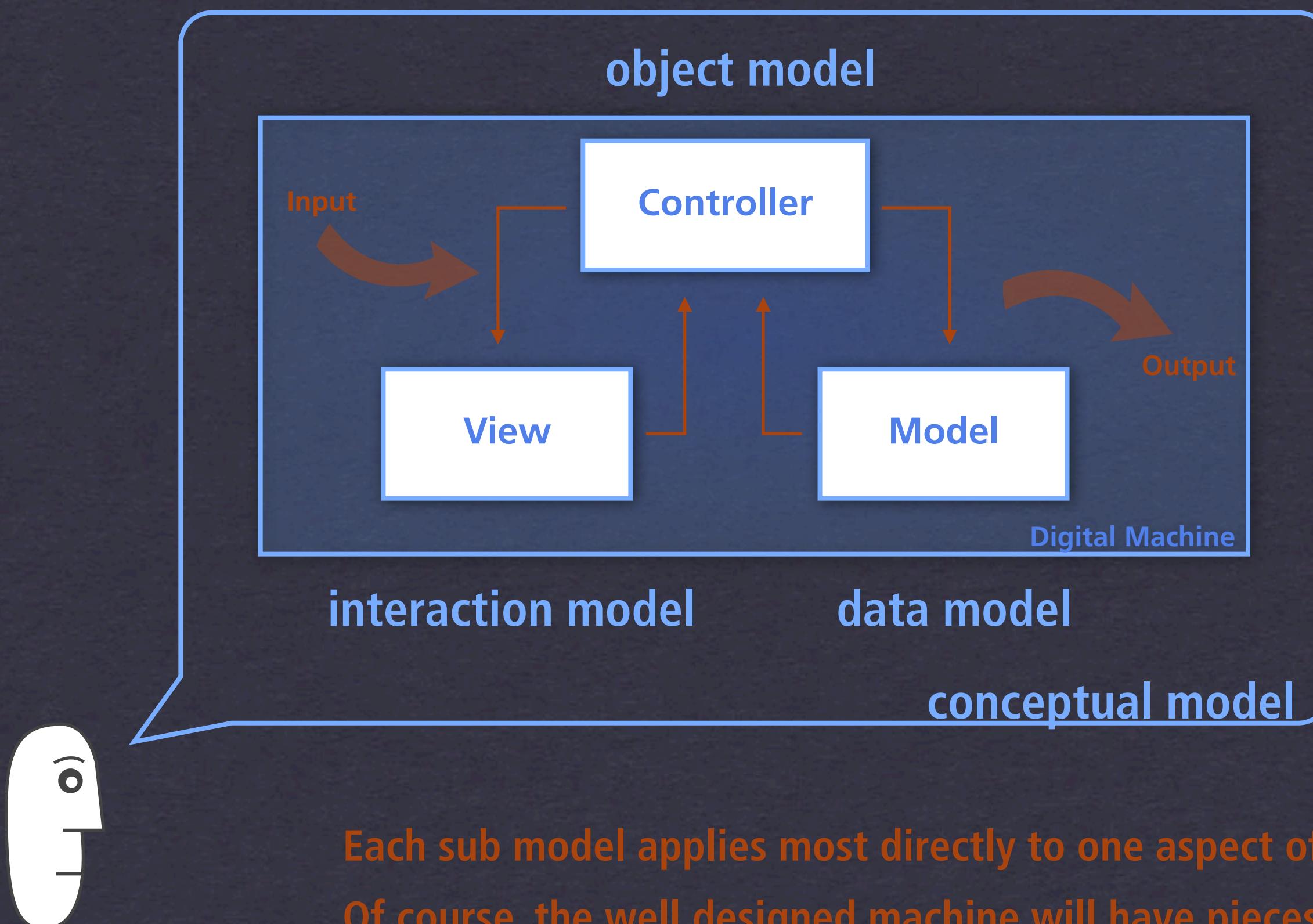
Object Model
Structure

How are the parts related?

Data Model
Inheritance



Correspondence between digital machine and the models



fin