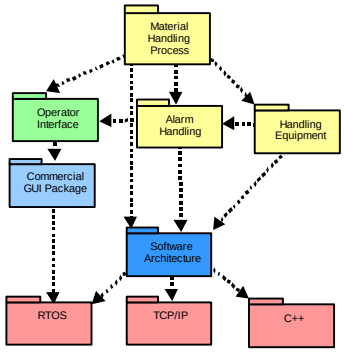
# Domains:
# The Good, the Bad, and the Ugly

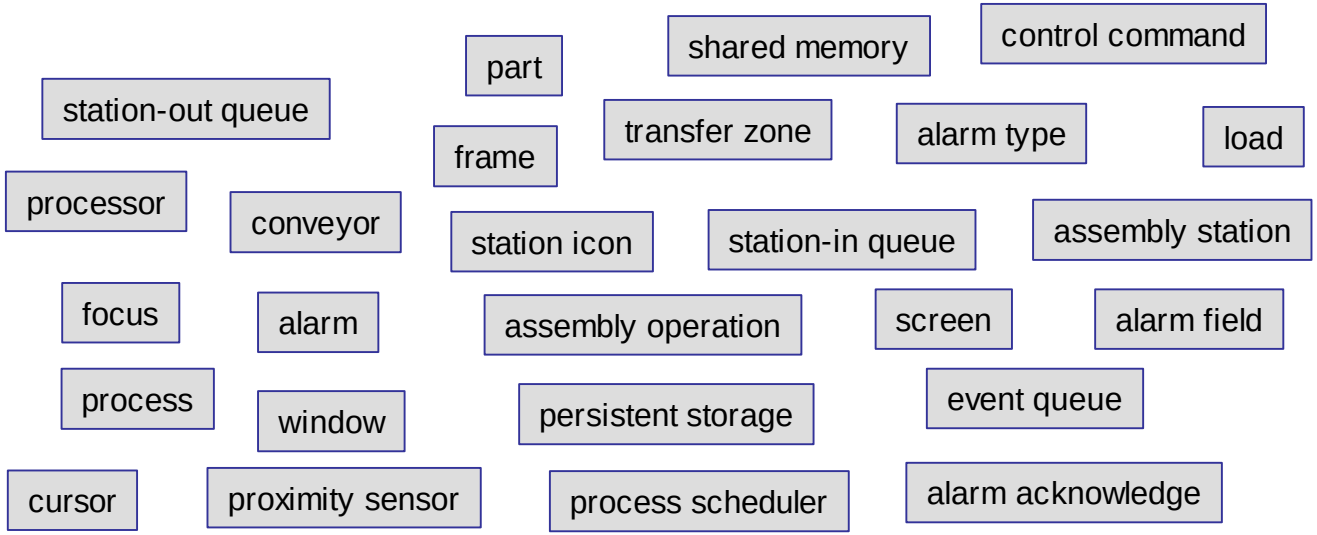
Model Integration, LLC
www.modelint.com

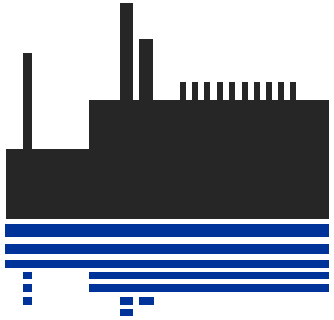



M O D E L  I N T E G R A T I O N

# Nature of Systems

Systems typically contain 10's, 100's, and sometimes 1000's of objects.



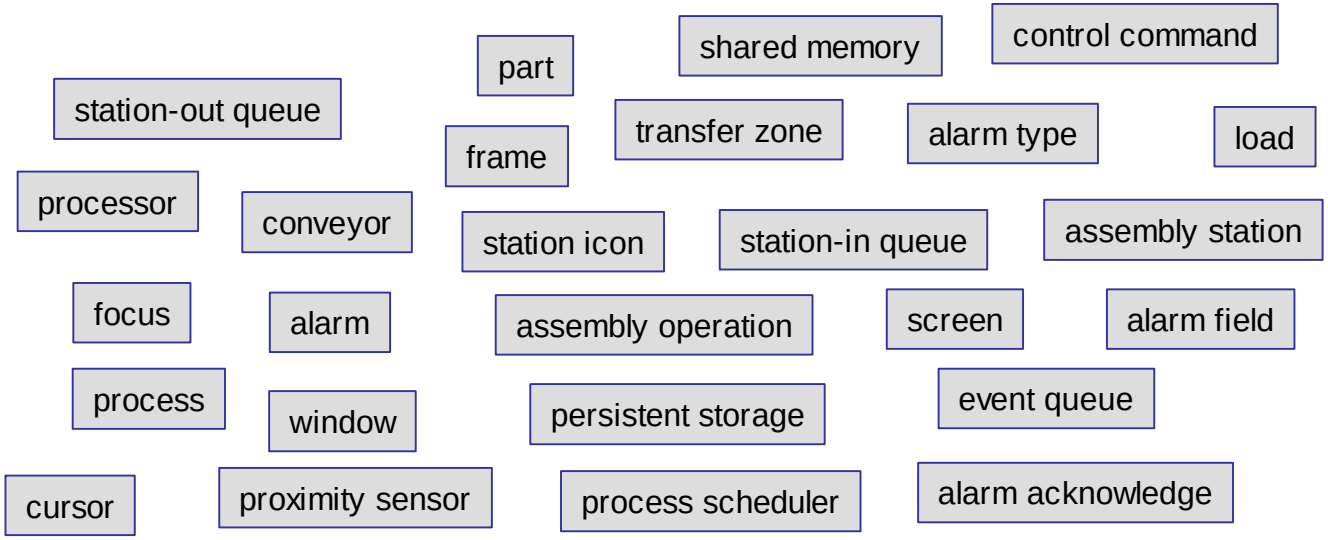



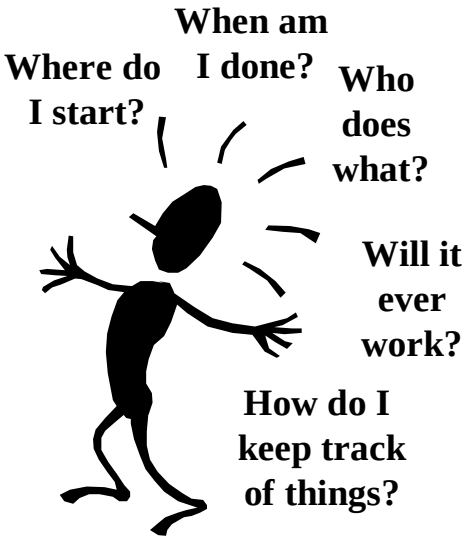
**Factory Material Handling System**

© 1999 Model Integration, LLC

# Systems and Size

This number of objects makes understanding and building the system a significant problem.

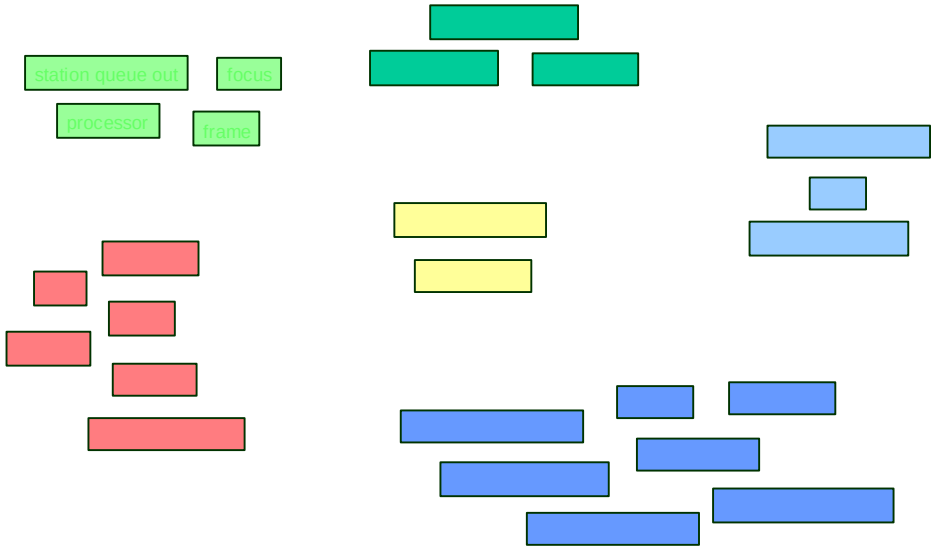



© 1999 Model Integration, LLC

# Managing Size

An effective way to manage size is to partition a large problem into distinct, smaller problems.



© 1999 Model Integration, LLC

# Partitioning Issues

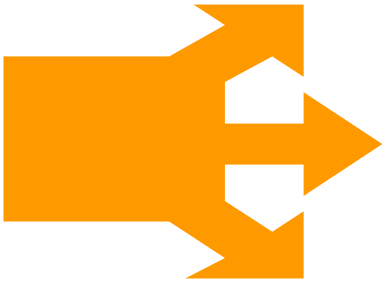
When something is partitioned, however, the following issues arise.

- Components - What's the criteria for partitioning?
- Interfaces - How do the partitioned elements interact?
- Representations - How do we document the partitioning?

© 1999 Model Integration, LLC

# Partitioning a House

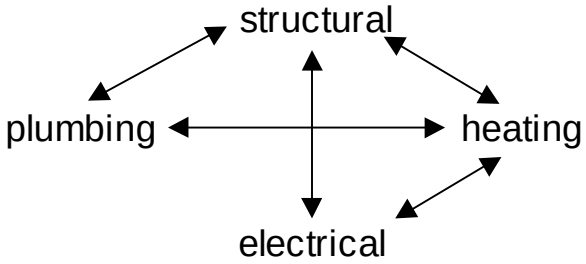
House Specification (model)



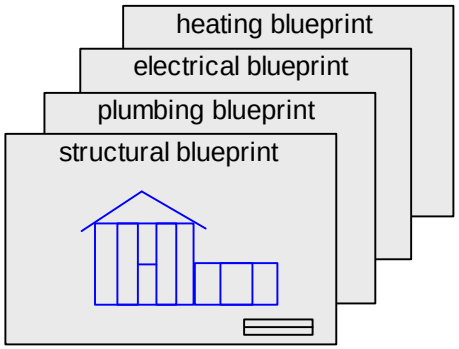
## Components

- structural
- plumbing
- electrical
- heating

## Interfaces



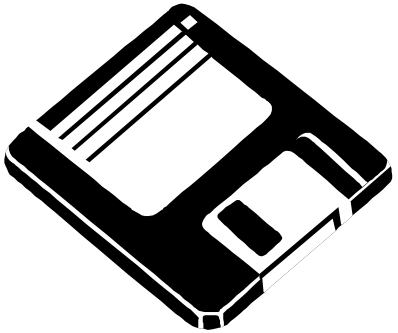

## Representations



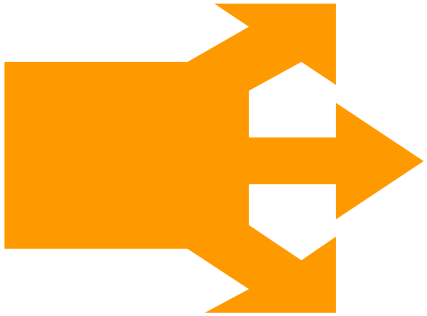

© 1999 Model Integration, LLC

# Partitioning Software

With the OOA/RD:



Software Specification (model)



Components

**Domains based on Subject Matter**

Interfaces

**Bridges based on modeled interactions**

Representations

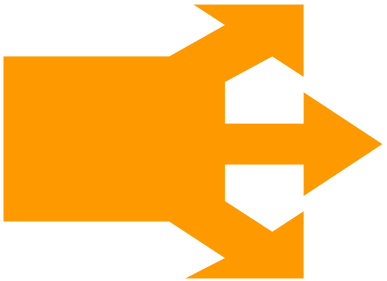
**Domains represented as UML packages**
**Bridges represented as UML package dependency**

© 1999 Model Integration, LLC

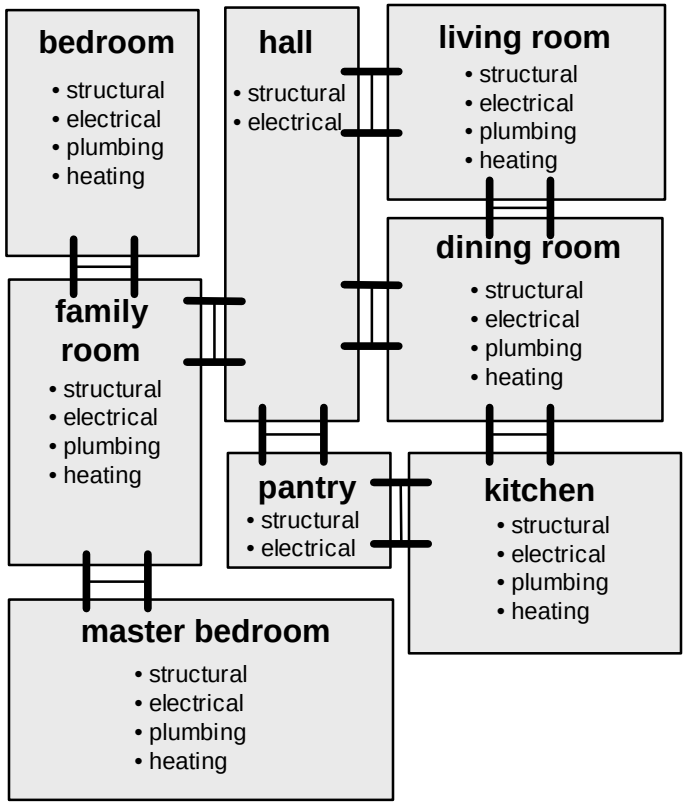
# But Suppose for a house...

...the partitioning criteria was rooms?

House Specification (model)

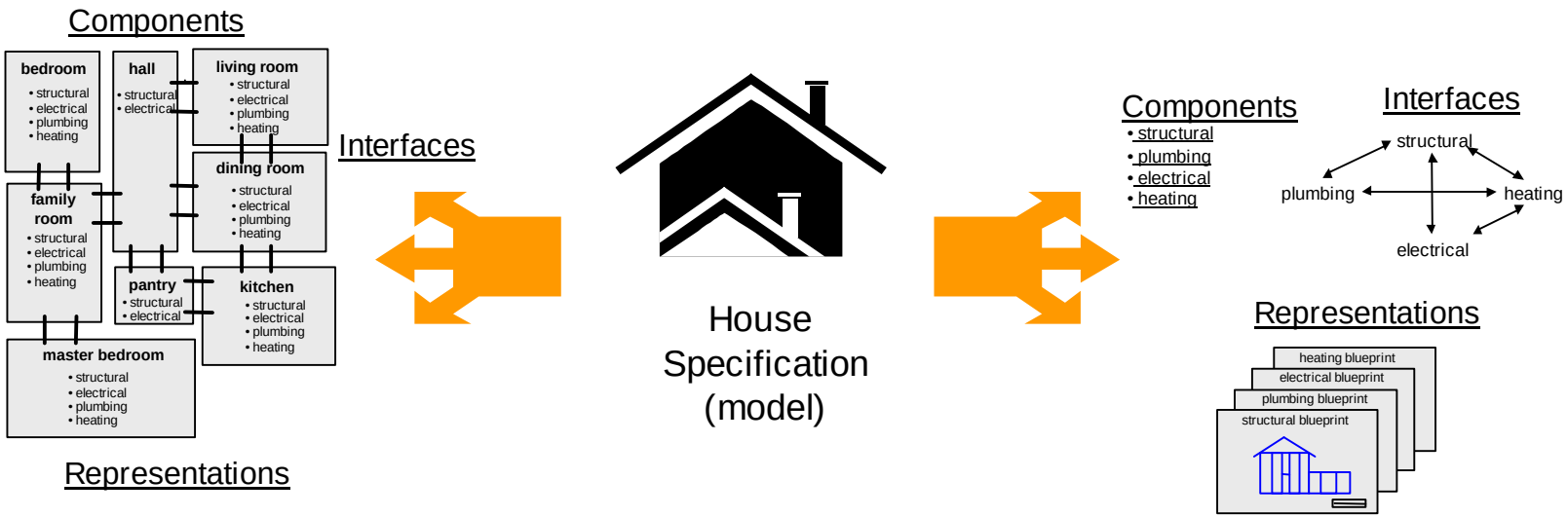


Components




© 1999 Model Integration, LLC

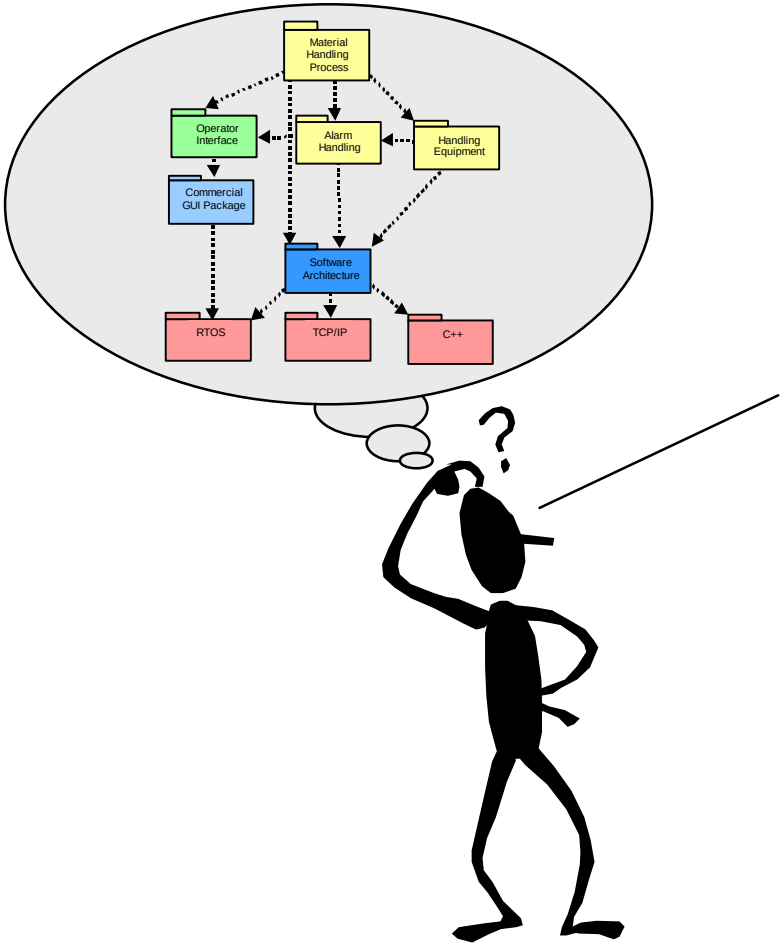
# Thus...

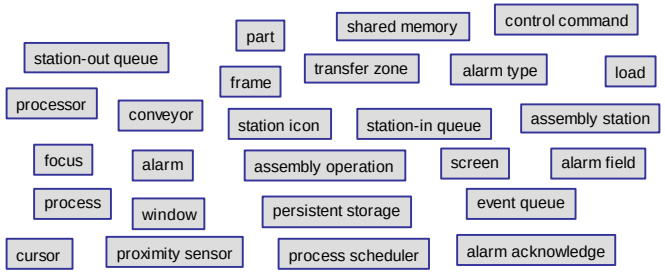



Proper partitioning is essential for reducing complexity

© 1999 Model Integration, LLC

# What to do? What not to do?

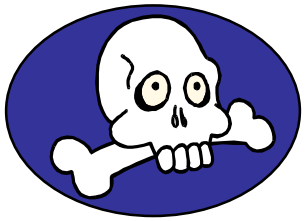


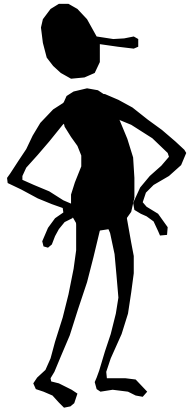
**Factory Material Handling System**

The **Good**,
the **Bad**,
and the **Ugly**

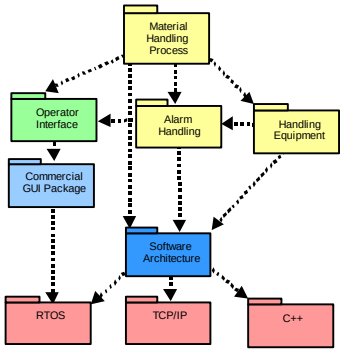
© 1999 Model Integration, LLC



# Non Subject Matter Partitioning



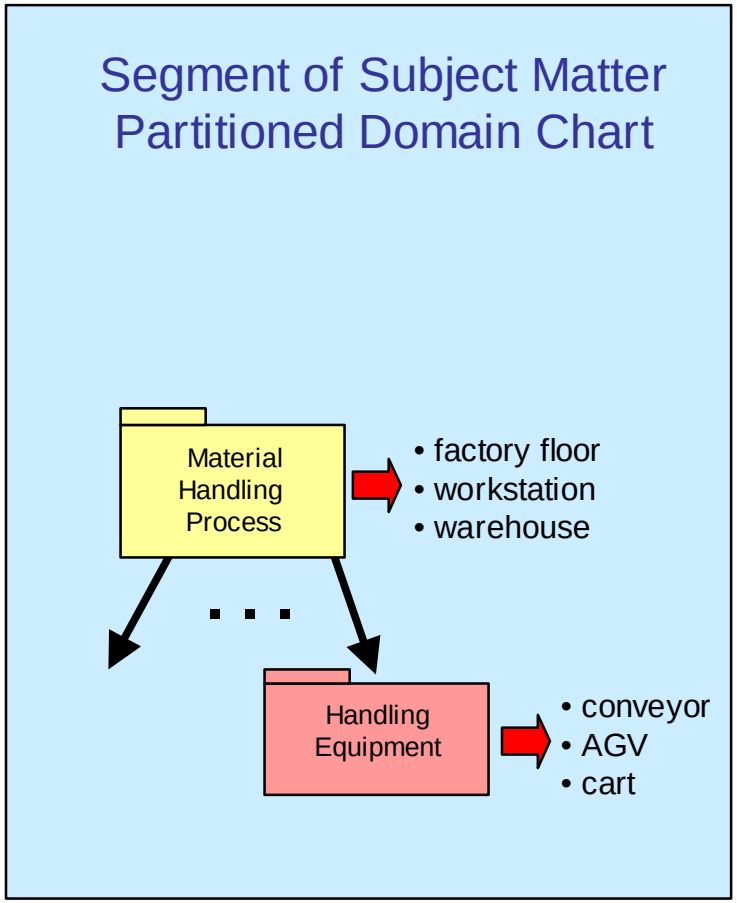
**"Senior" Developer**

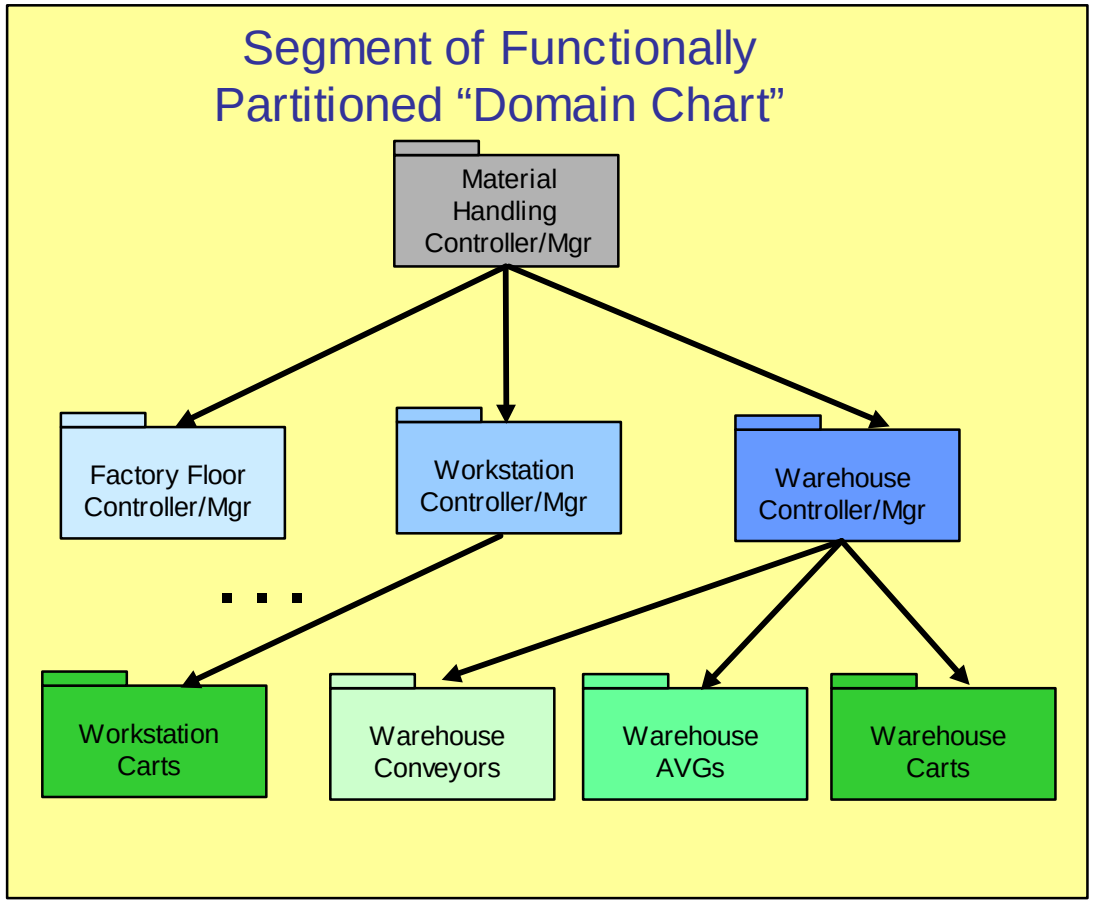



- Functional
- Hardware Components
- Execution Units
- Organizational Boundaries

© 1999 Model Integration, LLC

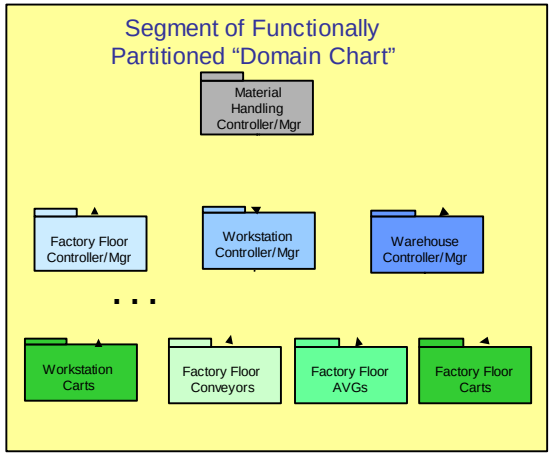
# The Functional Approach




© 1999 Model Integration, LLC

# Non Subject Matter Partitioning




## Driven By:

- Common habit
- Poor/No abstraction

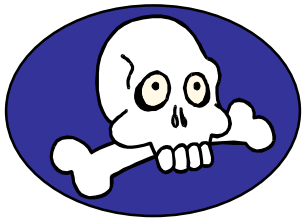
## Results:

- Model the same, or similar, thing in many places
- Significantly increases the effort, size and complexity

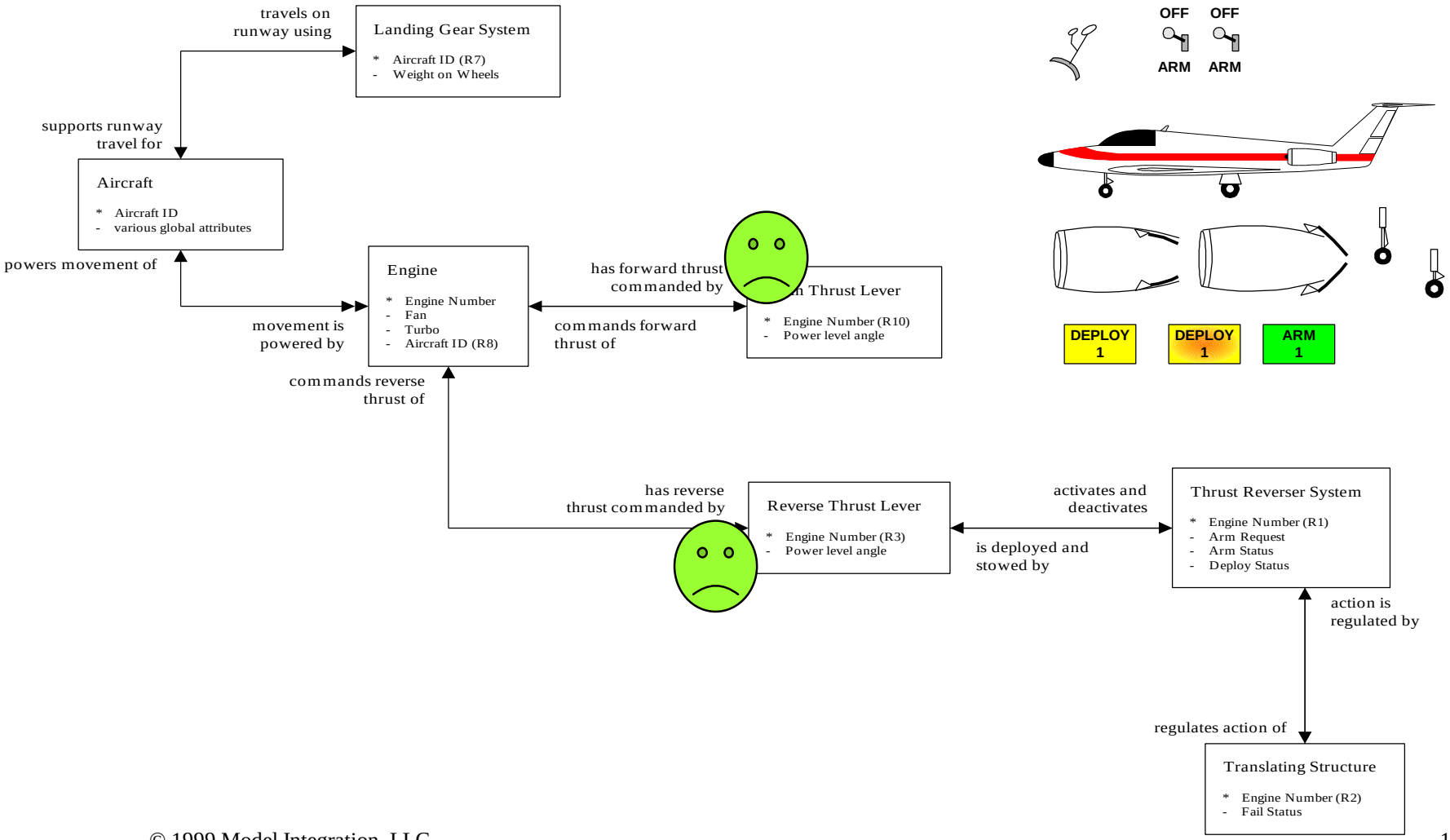
## Recovery:

- Repartition based on subject matter
- Offer the "senior" engineer early retirement

© 1999 Model Integration, LLC



# UI mixed into the Application




© 1999 Model Integration, LLC

# UI and Application Separated

| Internal Events | External Actions |
|---|---|
| PIO 1 : TR Unarmed | Turn Arm Lights OFF |
| PIO 2 : TR Armed | Turn Arm Lights ON |
| PIO 3 : TS Deployed | Turn Deploy Lights ON, Release Reverse Thrust Lever Idle Lock |
| PIO 4 : TS Not Deployed | Turn Deploy Lights OFF |
| PIO 5 : TS Stowed | Release Main Thrust Lever Idle Lock |
| | |
| TR 8 : Deploy | Reverse Thrust Lever Pulled UP |
| TR 11 : Stow | Reverse Thrust Lever Pushed DOWN |
| TR * Set Arm Request ON/OFF | Arm Switch ON/OFF |
| | |
| EN 1 : New Thrust | Change Main or Reverse Thrust Lever Angle |
| EN 3 : Enter Idle | Main or Reverse Thrust Lever enters IDLE position |
| | |
| APS 1 : Arm for Deploy | Open hydraulic selector valve (simulate) |
| APS 2 : Deploy TR | Set timer, trigger deploy sound |
| APS 3 : Stow TR | Set timer, trigger stow sound |
| APS 4 : New Forward Thrust | Adjust engine sound |
| APS 5 : New Reverse Thrust | Adjust engine sound |
| APS 6 : Disarm for Deploy | Close hydraulic selector valve (simulate) |

* This designates an event which triggers asynchronous action, so there is no event number

© 1999 Model Integration, LLC

# Leon's rule for separating the UI from App

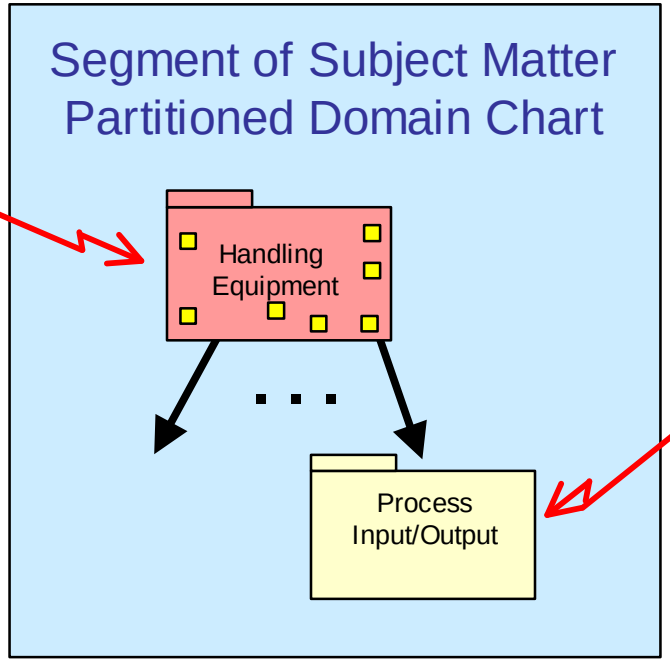
- Start with the feature in question
- Assume multiple interfaces (GUI, TUI, RUI, VUI) even if they don't exist
- Then ask the question "Must the proposed feature exist in each xUI?"
- If the answer is "yes", then it's in the Application!

© 1999 Model Integration, LLC

# Poorly Factored Behavior

Common behavior pattern.

- **Handling Equipment acts on changes in data**
- **It polls looking for changes**



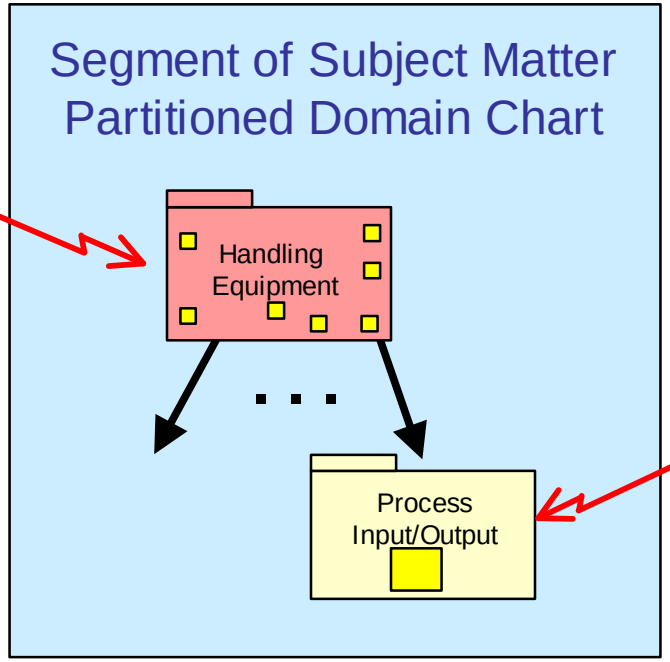

- **H/W periodically samples data**
- **PIO synchronously updates**

© 1999 Model Integration, LLC

# Refactoring Behavior

Common behavior pattern.

- **Specifies event criteria**
- **Waits for PIO events**



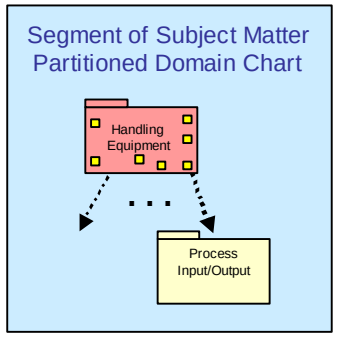

- **H/W periodically samples data**
- **PIO synchronously updates**
- **PIO <u>generates events</u> on specified changes**

© 1999 Model Integration, LLC

# Poorly Factored Behavior

## Driven By:

- Literal modeling of real world
- Separate groups doing domains




## Results:

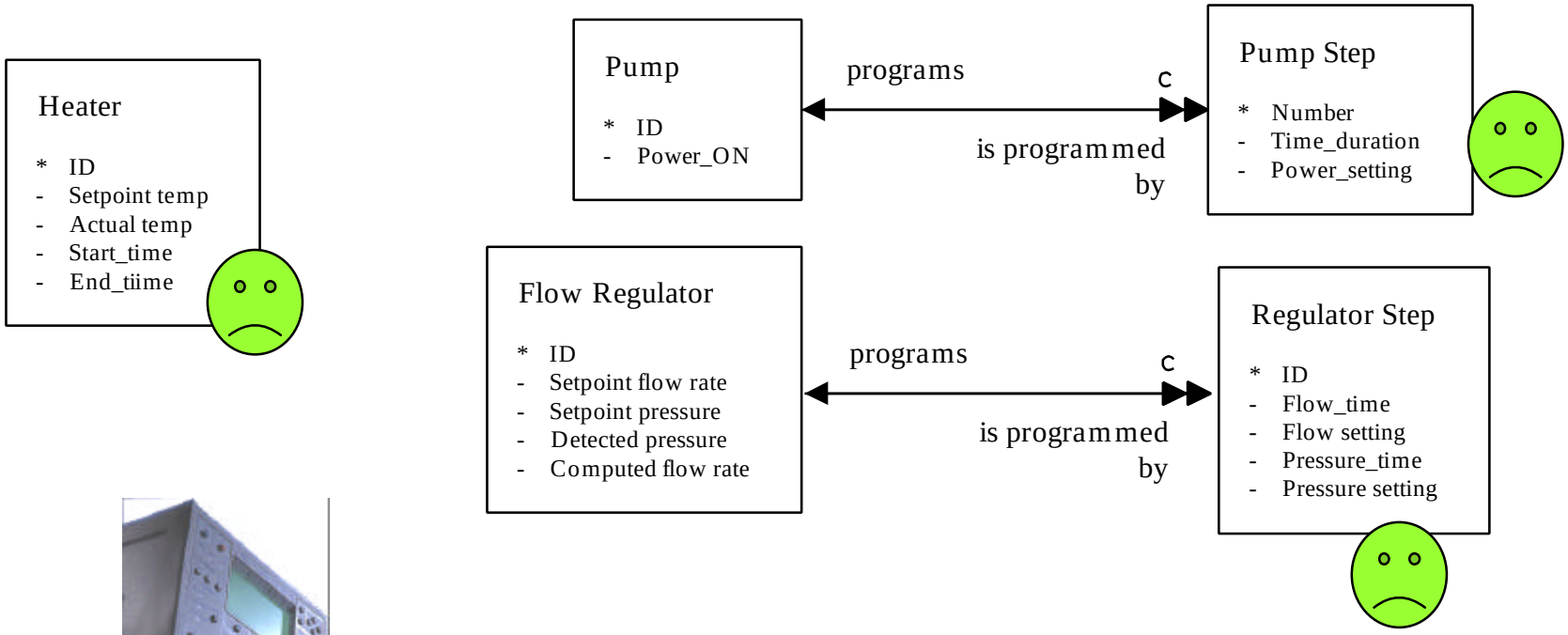
- Repeating model pattern in the client domain
- Moderately increases the effort, size and complexity

## Recovery:

- Factor the behavior pattern out of client
- Place behavior pattern in service

© 1999 Model Integration, LLC

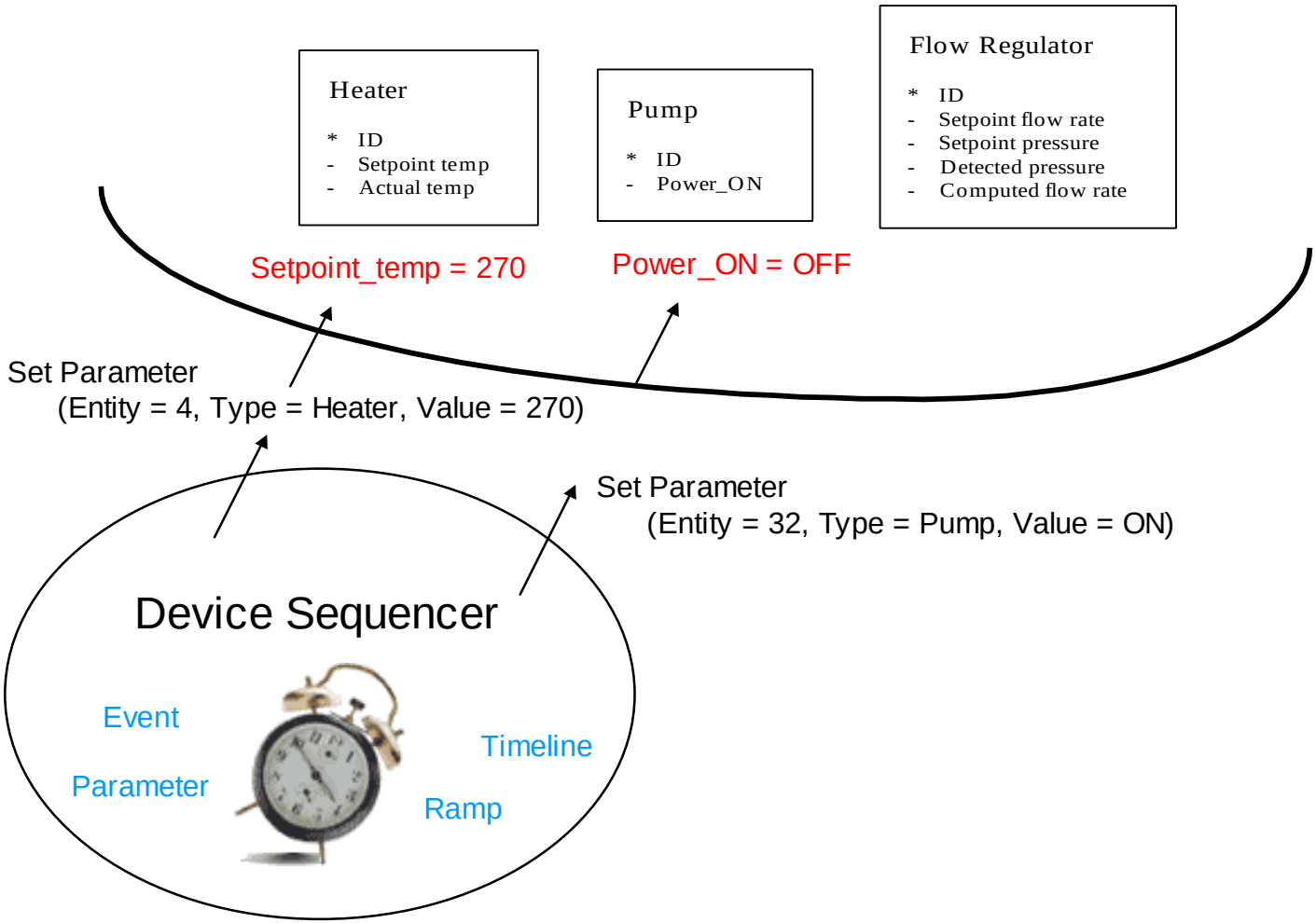
# Time control mixed into the App



**Gas Chromatograph**

© 1999 Model Integration, LLC

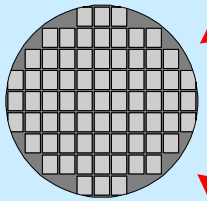
# Concentrate time control in one place




© 1999 Model Integration, LLC

# Overachieving Object

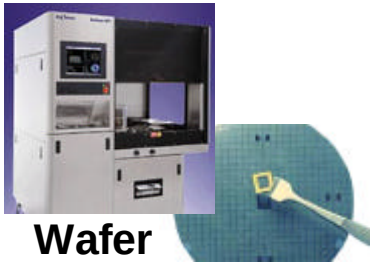
"Everything There is to Know About Wafer" Domain



- wafer map
- die geomety
- defects

- scan area
- scan geometry
- scan parameters

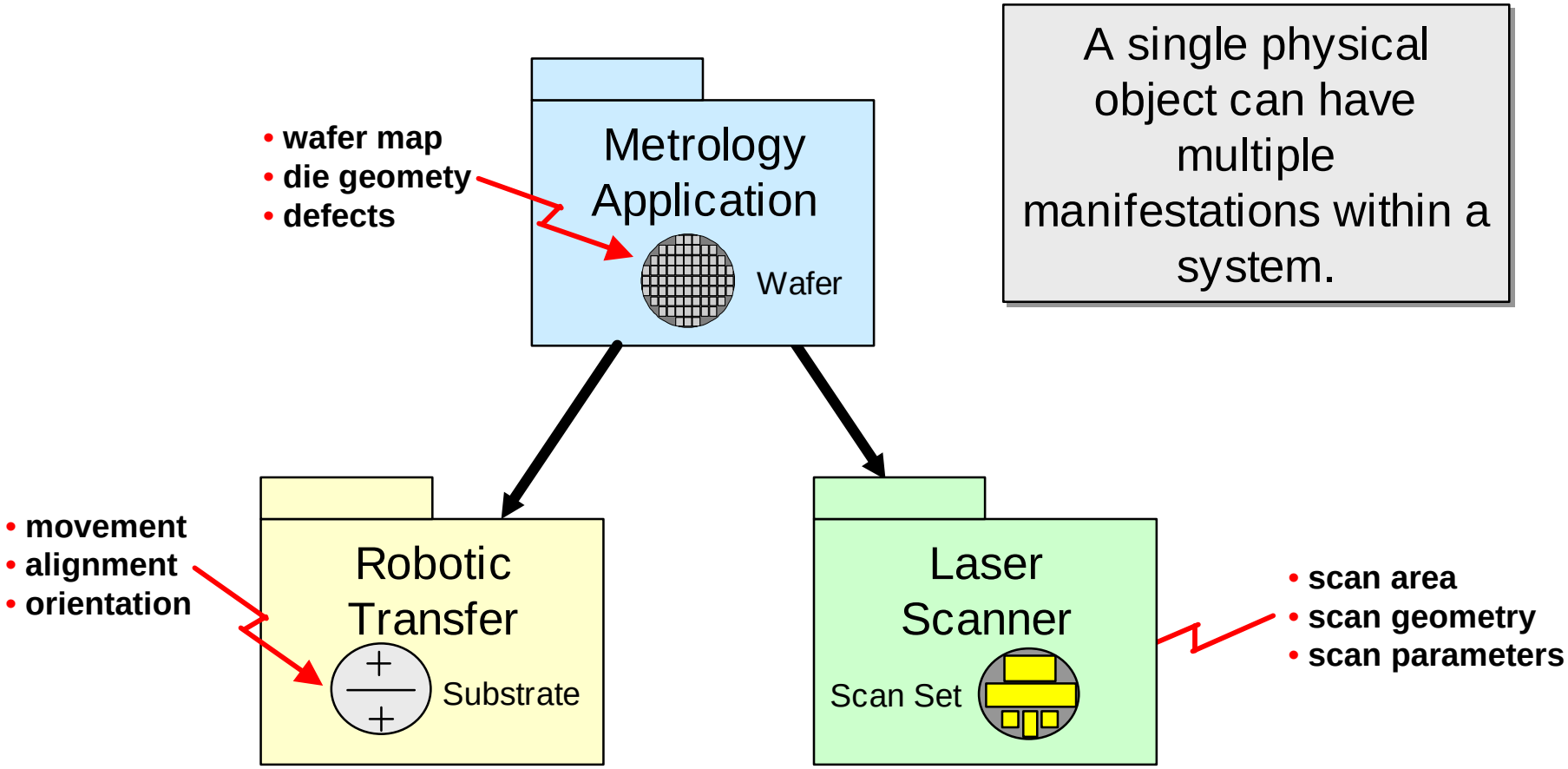
- movement
- alignment
- orientation



**Wafer Inspection**

© 1999 Model Integration, LLC

# Counterpart Objects



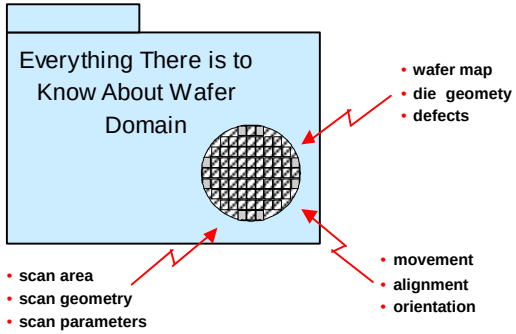

A single physical object can have multiple manifestations within a system.

© 1999 Model Integration, LLC

# Over Achieving Object

## Driven By:

- Literal modeling of real world




## Results:

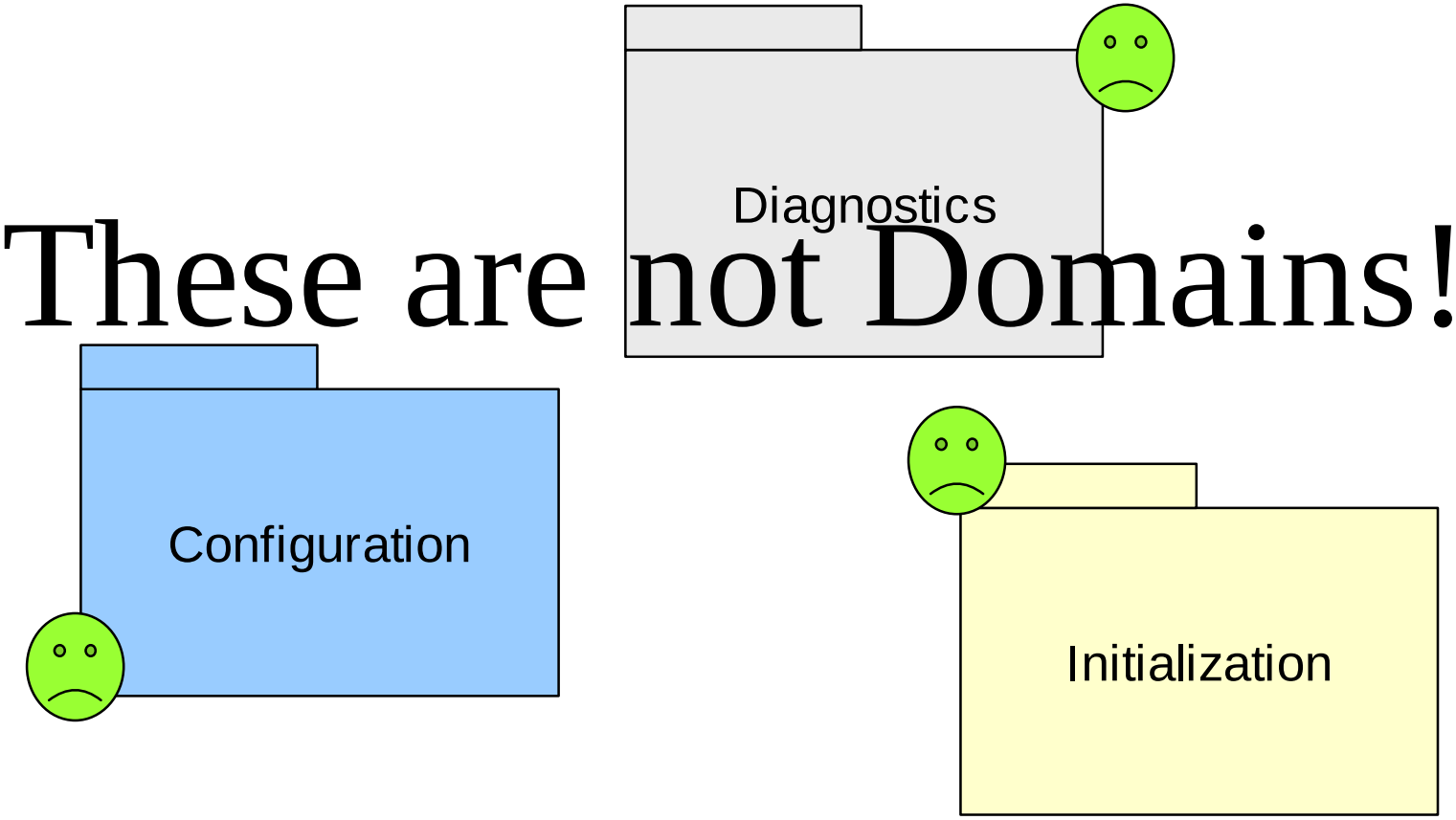
- Mixing multiple subject matters in single domain
- Significantly increases the complexity of the model

## Recovery:

- Abstract the object at multiple levels
- Ensure counterpart linking

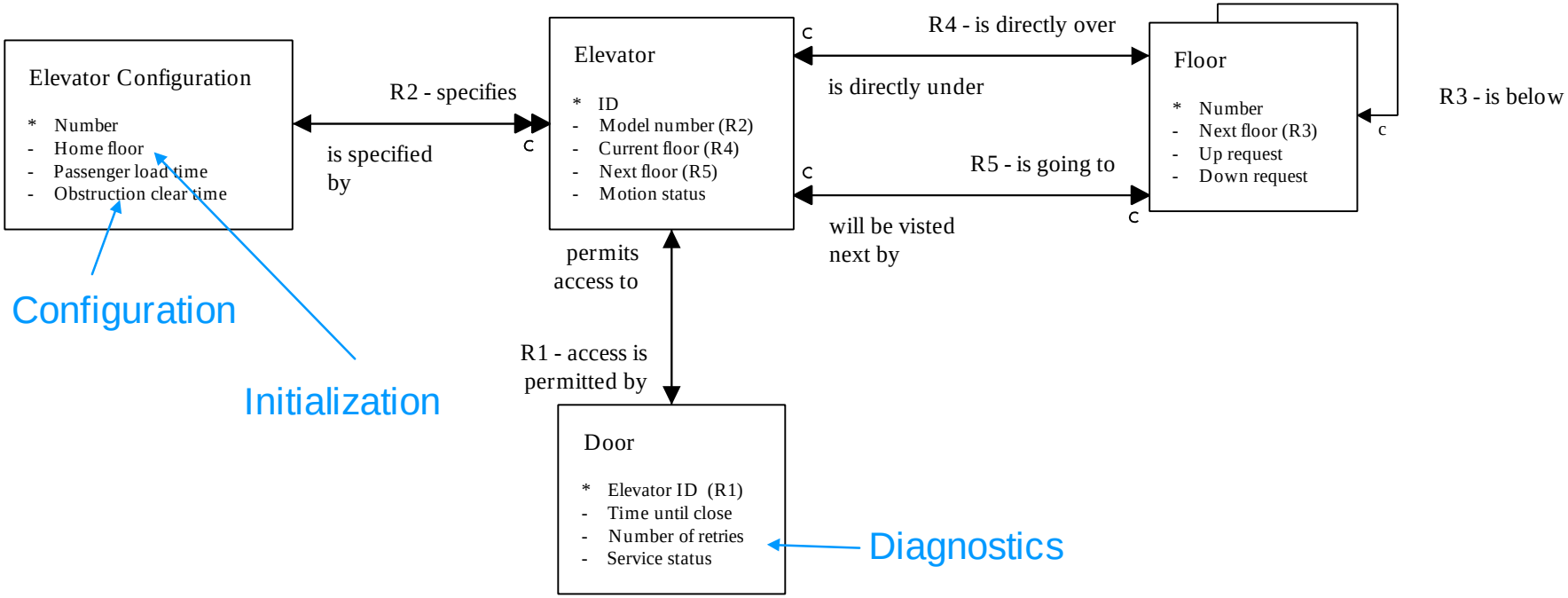
© 1999 Model Integration, LLC

# Critical System Services




© 1999 Model Integration, LLC

# It’s all in the objects




© 1999 Model Integration, LLC

# A Good Domain Chart is a High Leverage Item

**Don’t leave it to luck!**

## Do...

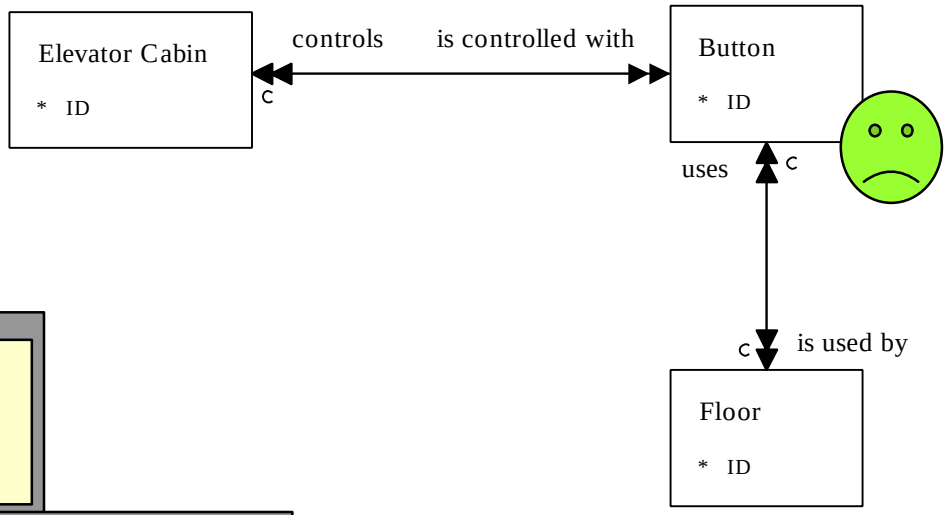
- Invest time up front
- Review as modeling proceeds
- Revise when necessary

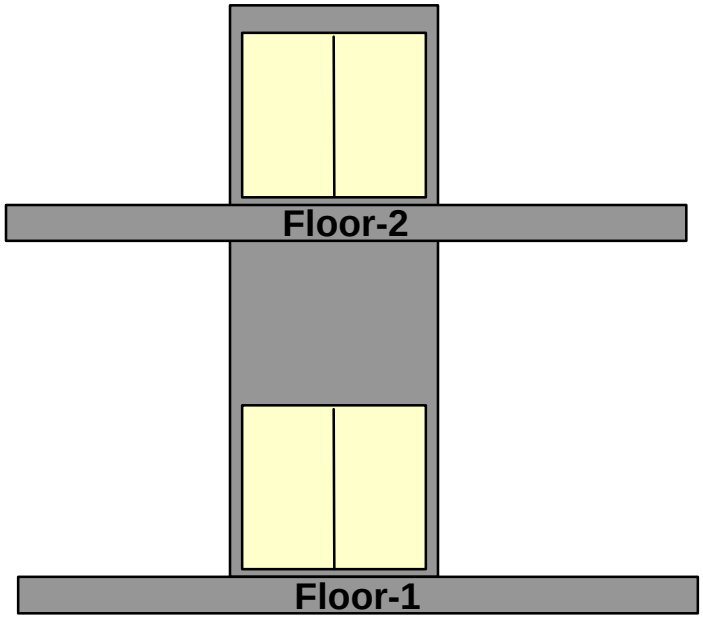
## Don’t...

- Functionally partition
- Mix subject matters together
- Model the world too literally
- Leave it to luck!

© 1999 Model Integration, LLC

# A common mixed UI / App




© 1999 Model Integration, LLC