



Applied Software Architecture Patterns

Key to successful systems

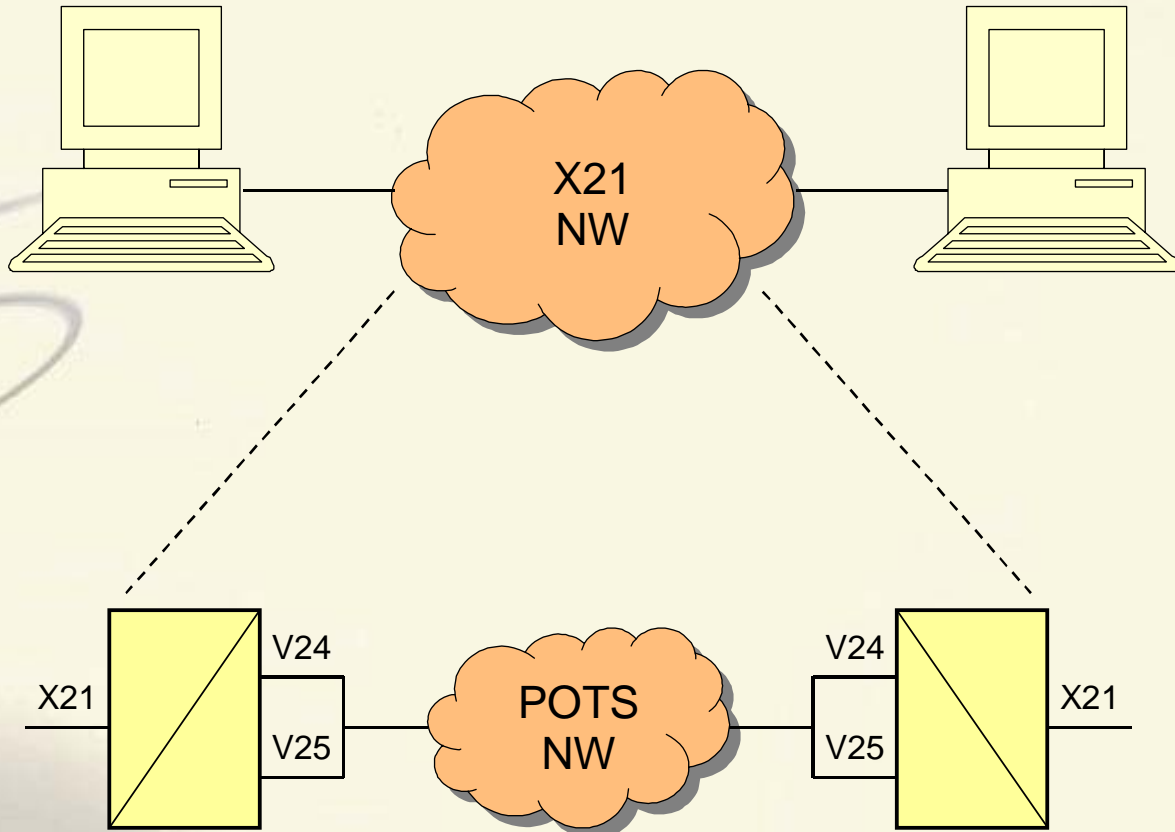
Sioux Embedded Systems

Chris Offerman

Agenda

- **A tour of some successful systems**
 - **Telecom** convertor system
 - State machine
 - **Medical** Imaging system
 - Software bus
 - **Factory Production Control** system
 - Software internet
 - **Semi Production Equipment** system
 - Black board
- **Some Theses...**

Telecom Converter System



Protocol Conversion...

- **ISO Protocol Stack**
 - 7 layers
- **Need to handle 3 lower layers**
 - Event-driven (**reactive**)
 - Complex
 - Completeness
 - Correctness
 - Robustness
 - Error Handling
 - Full-proof

Table-driven State Machine

- Event, State → Action, Next State

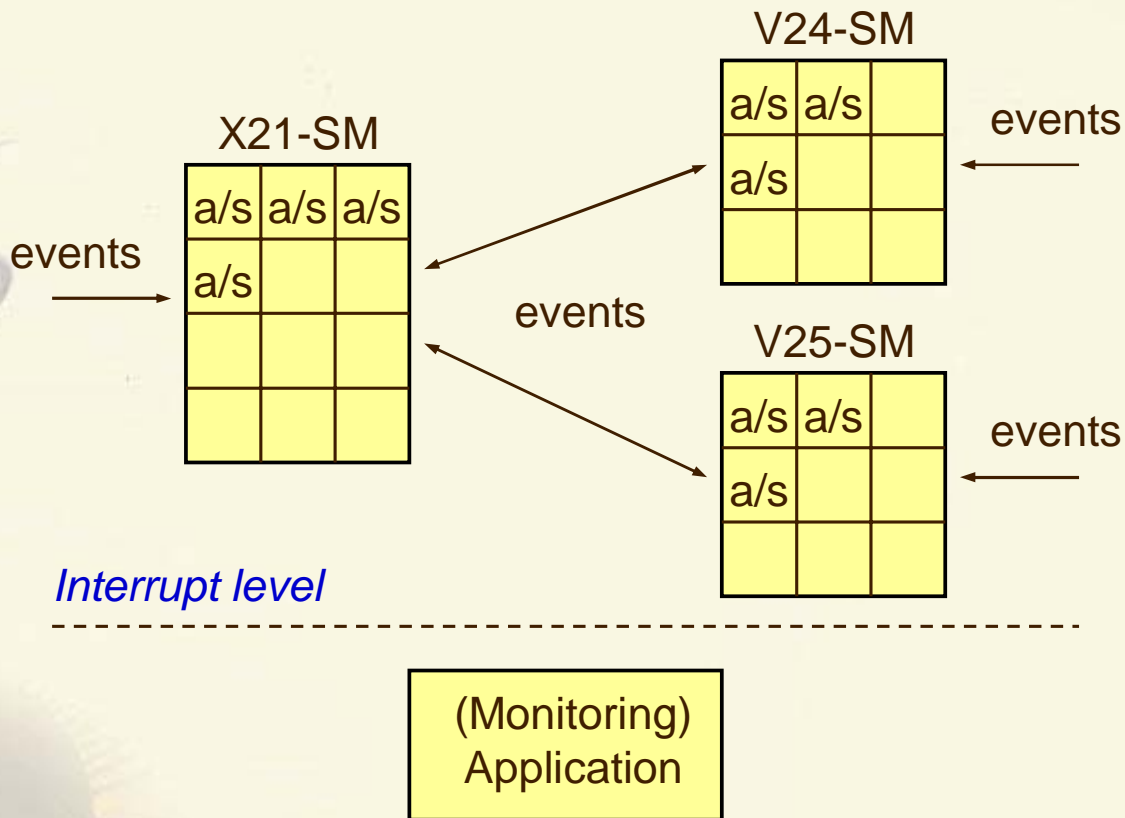
events →

s	a/s	a/s	a/s	
t	a/s	a/s		
a				
t				
e				
s				
↓				

- Combinatory Explosion...
 - Does not scale well

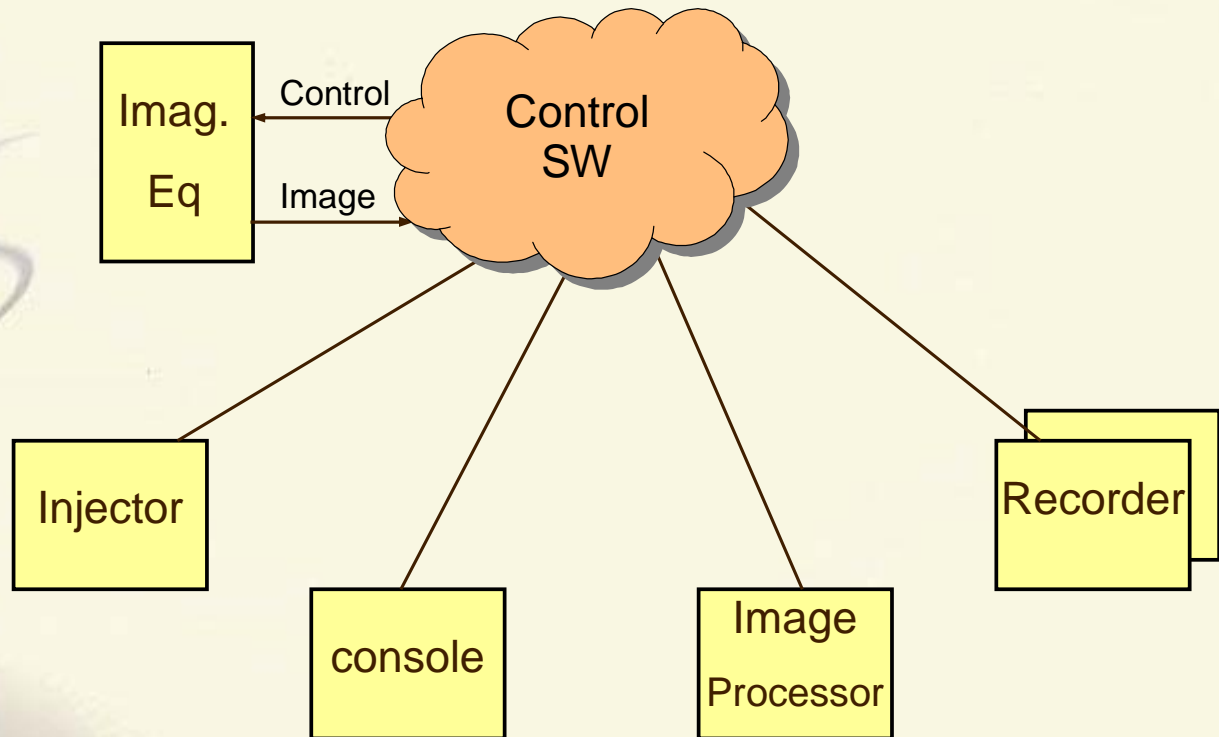
Convertor SW Architecture

- Communicating State machines



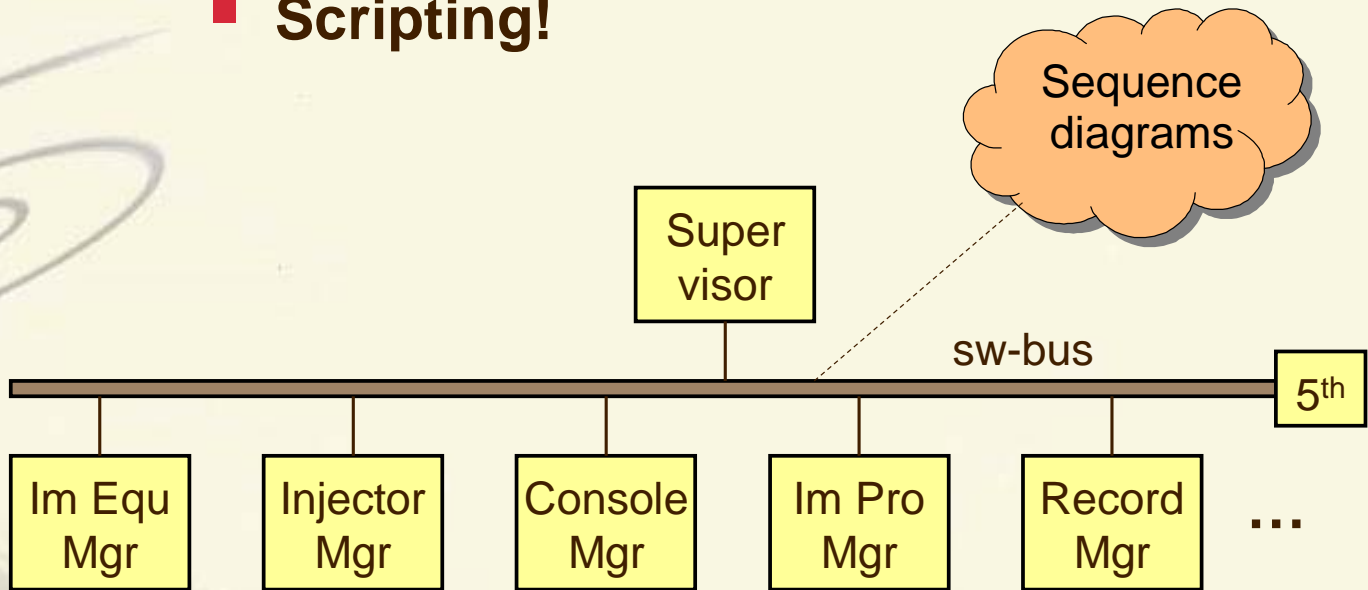
- Division of State-Space! (decoupling)

Medical Imaging Equipment

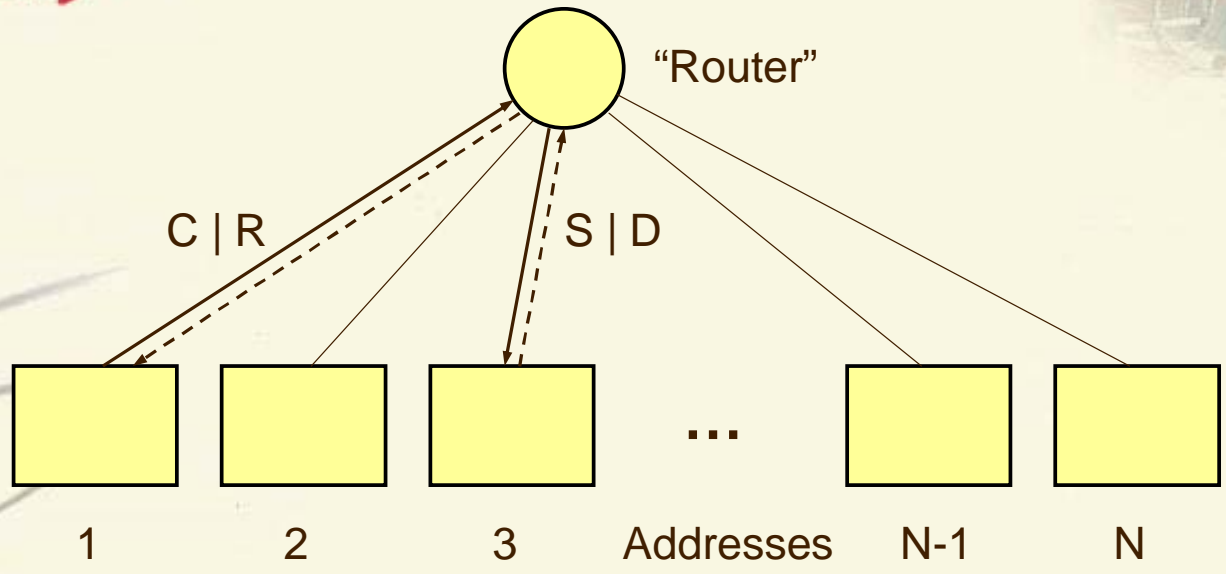


Imaging SW Architecture

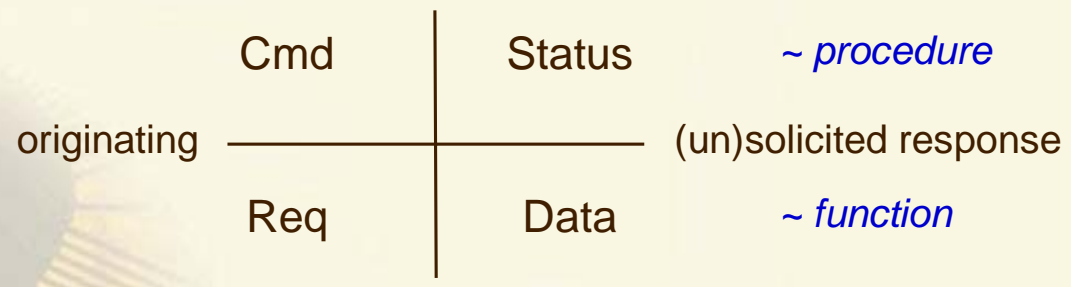
- SW Modularity = HW Modularity
- Software Bus...
- Scripting!



Software Bus...



■ Messaging Scheme...

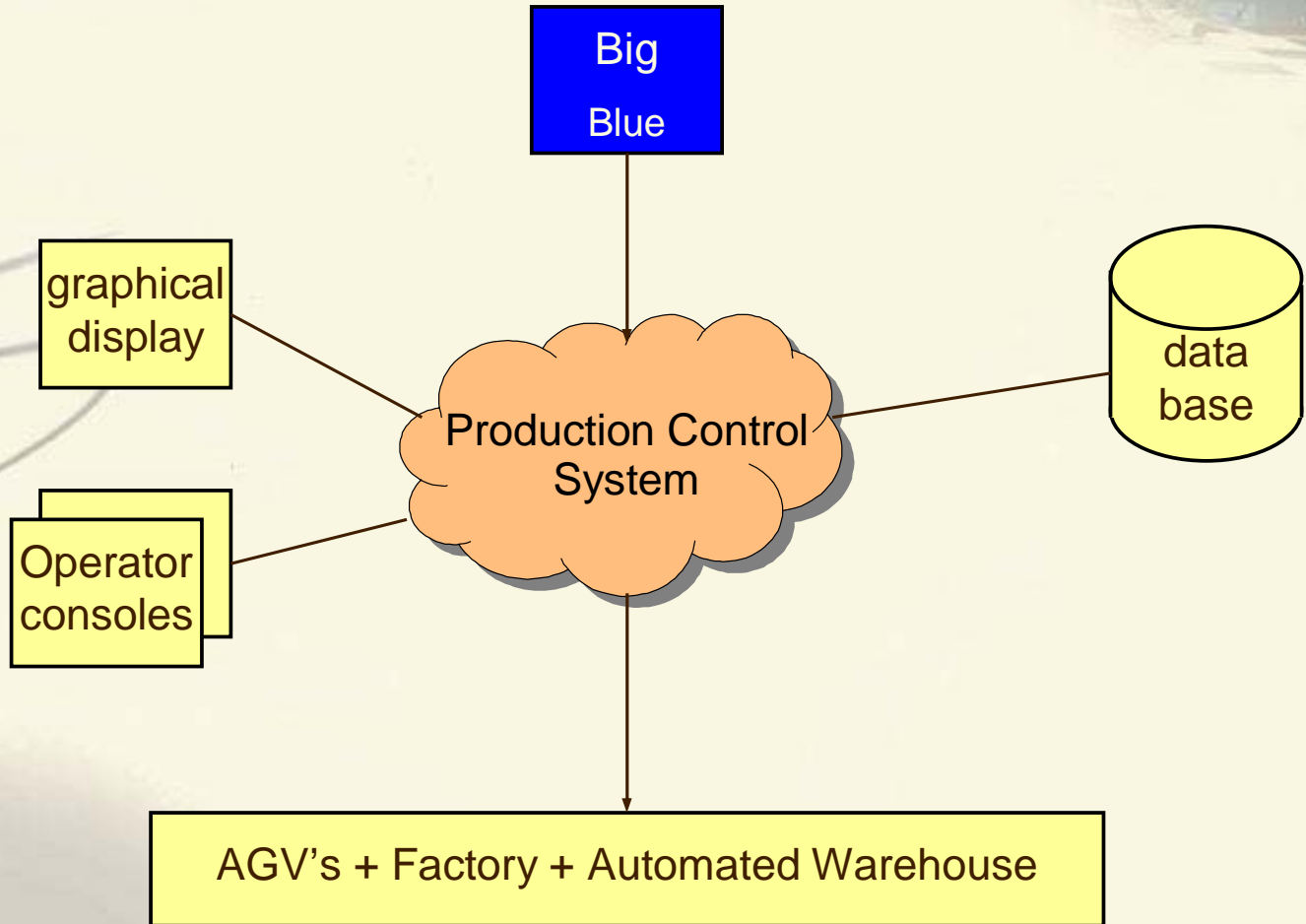


...Limitations

- System Extensions
- Varying Configurations
- Non-hierarchical (no abstraction)
- Router Bottleneck
- ...

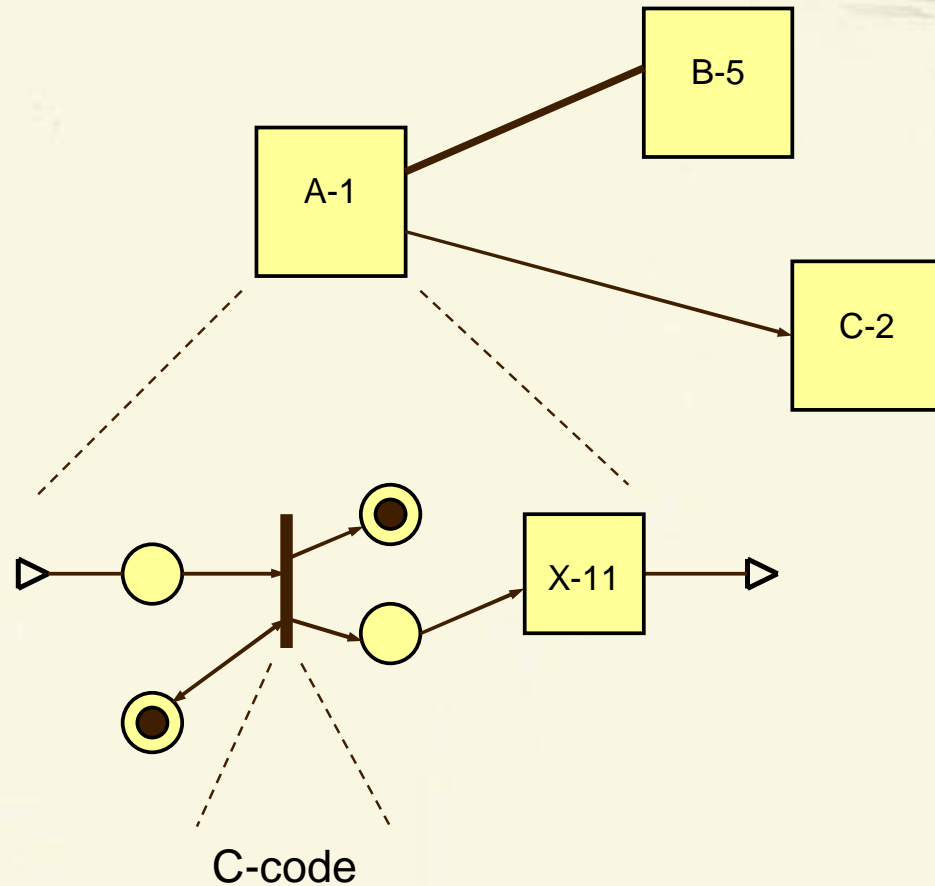


Factory Production Control

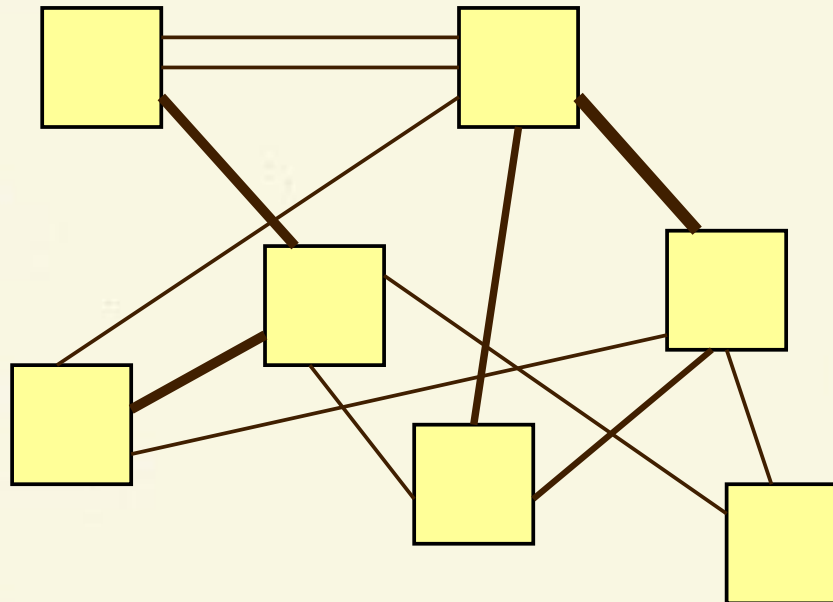




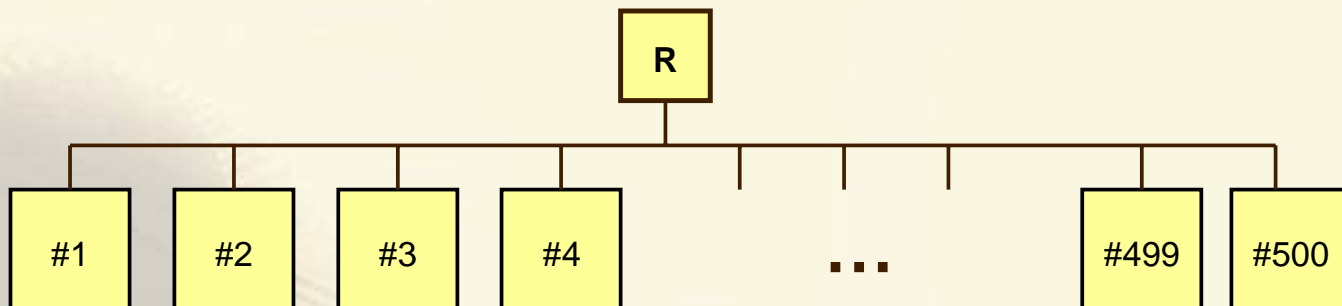
- **Model driven**
 - diagrams
- **Components**
 - object-based
- **Connections**
 - typed
- **Petri networks**
 - timed
 - colored
- **Operational lifecycle**
 - model
 - simulate
 - deploy



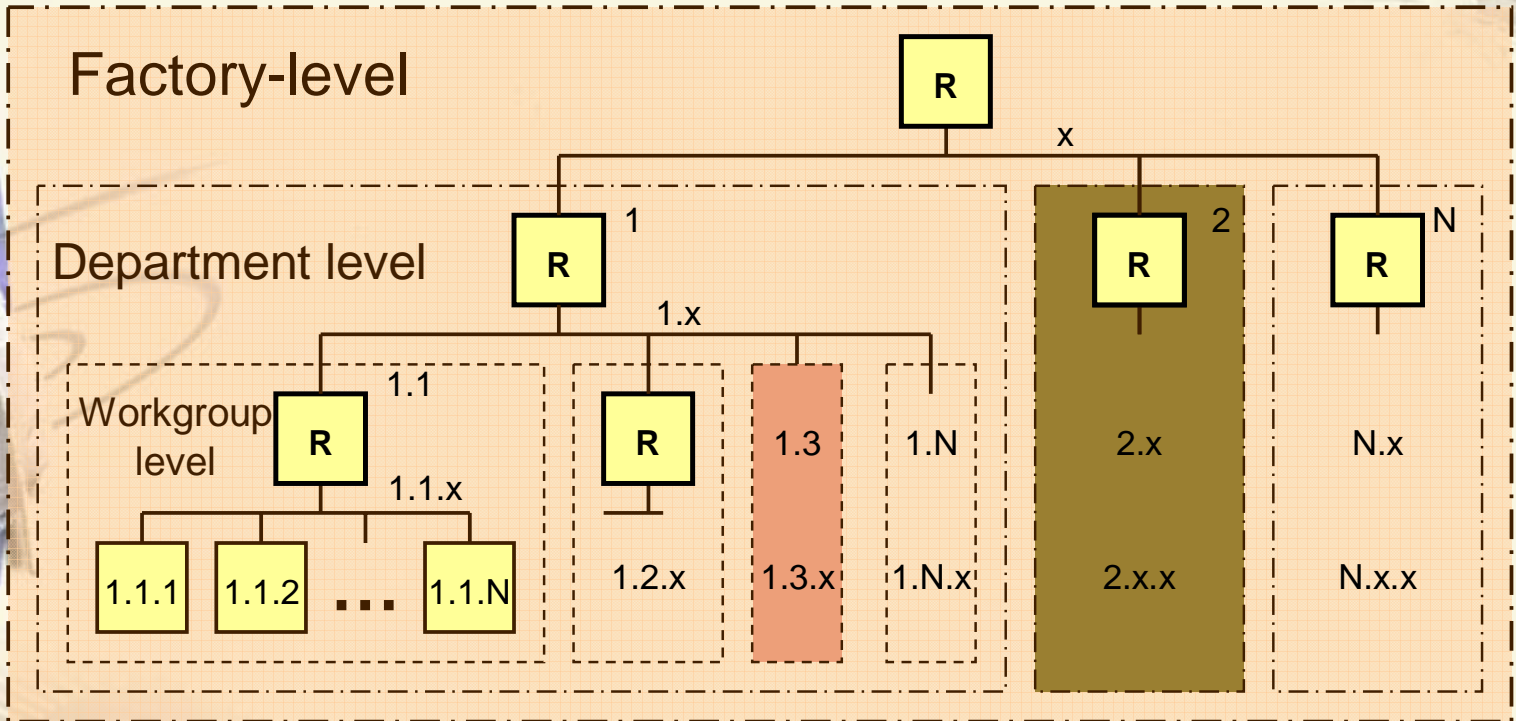
Factory Simulator...



■ **Software bus?**



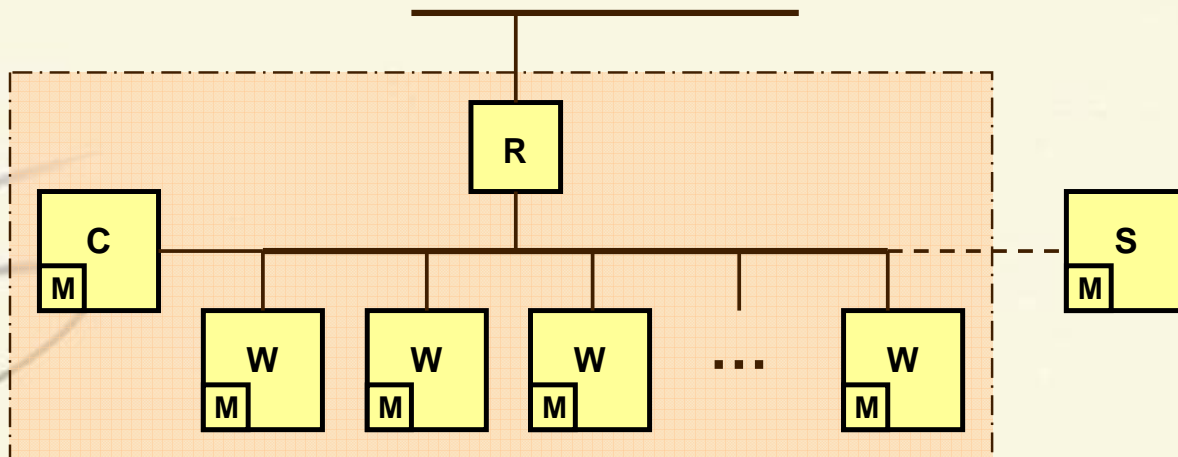
A software internet!





Matryoshka...

- Same Pattern at each level !



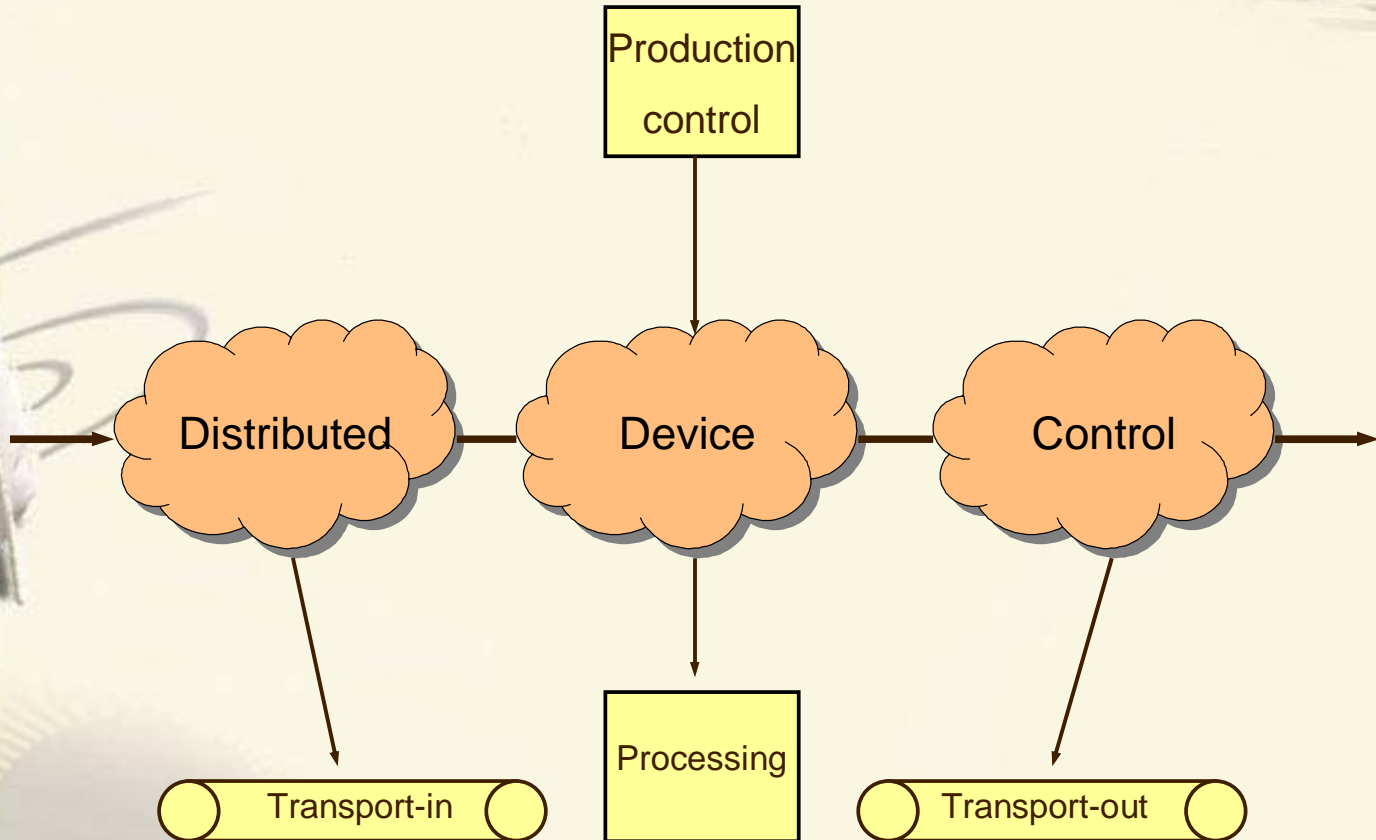
- Component**

- | | |
|----------------------------|---------|
| ▪ Session controller | 1 x |
| ▪ Router | 20 x |
| ▪ Configuration controller | 20 x |
| ▪ Main-state | > 200 x |
| ▪ Workers | 1..N x |

re-use



Semi Production Equipment





2nd Generation Architecture

■ Requirements

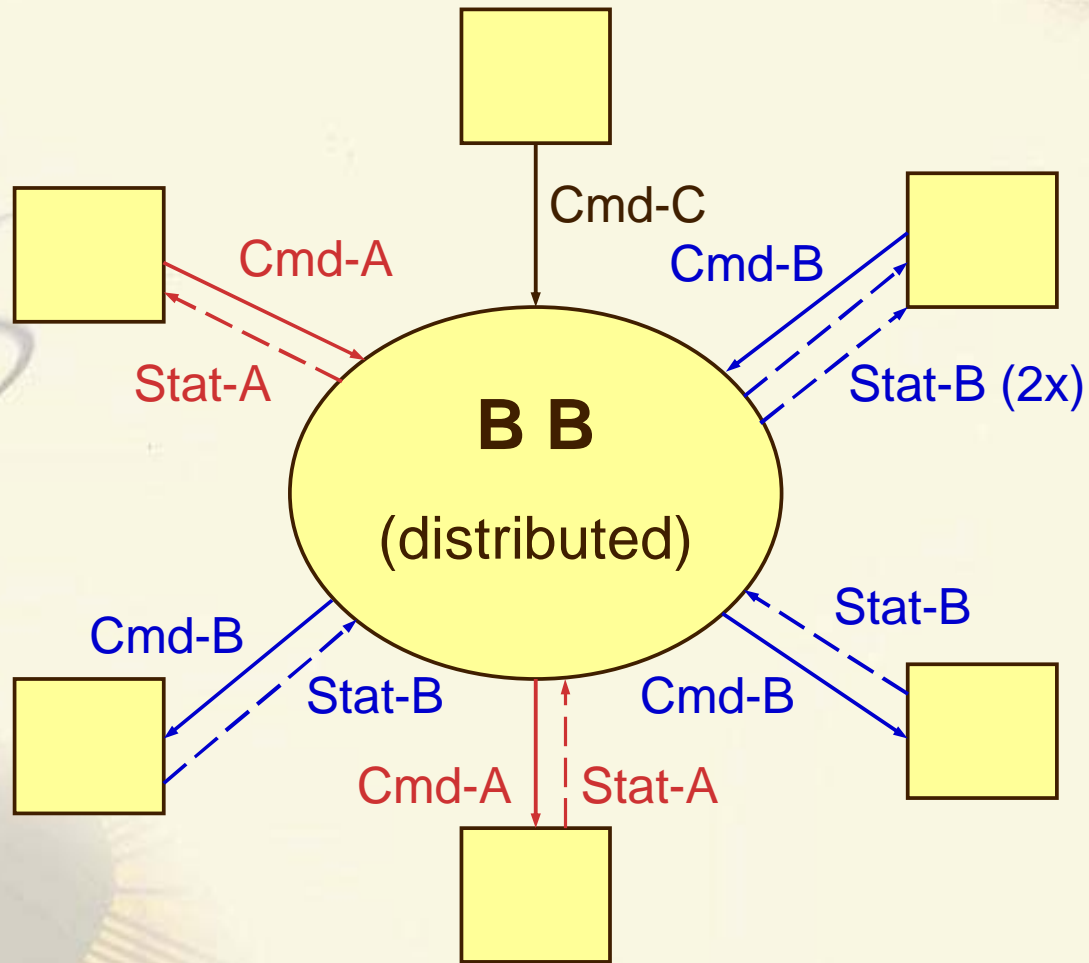
- Modular
- Distributed
- Good extensibility
- Varying configurations
- Location independence
- ...

■ Solution

- A Blackboard
 - Distributed
 - Message based



A Messaging Blackboard...





The Blackboard “Game”

- **Receivers subscribe** to the BB
 - to certain message **types**
- **Senders send messages** to the BB
 - specifying their **type**
- **The BB delivers them**
 - to all subscribers
- **Receivers respond** to the BB
 - specifying the **response-type**
- **The BB delivers them**
 - to the sender of the originating msg
- **The BB**
 - sends substitute time-out responses
 - discards too-late / random responses

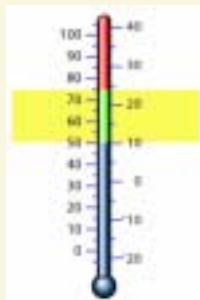


Simulation Models

- Discrete (time) phenomena...



- Continuous (time) phenomena...



- Yourdon : **flows vs stores**
- Petri nets : **Input vs Input & output places**
- Blackboards : **messages vs “tuple space”**

“Tuple Space” BB’s

- **“Tuple Space”**

- Linda (Yale)
- JavaSpaces, Jini

- **Applied in**

- Defence Systems
 - Thales SPLICE
- Building automation
 - Nedap AEOS Security Management (Jini)
- ...





Theses

- A “good” architecture
 - can be sketched on a page
 - is tailored to the problem
 - is as simple as possible (*Occam’s razor*)
- Aids
 - decoupling
 - abstraction - **multiple levels**
 - re-use - **no surprises...**
 - location-independence
 - ...

Questions...

