University of Zurich[UZH]

Department of Informatics

Harald Gall Martin Glinz Philipp W. Leitner

Advanced Software Engineering

Problem Solving, Problem Frames

General Problem Solving Strategies

- Decomposition
- Aspects/Views
- Patterns and metaphors
- Taxonomy of problem classes
- Normal vs. radical design
- Means vs. end

The World and the Machine

- ❍ Why not just specify and design the machine?
- ❍ What does S, W |– R actually mean?
- ❍ Illustration: the turnstile problem

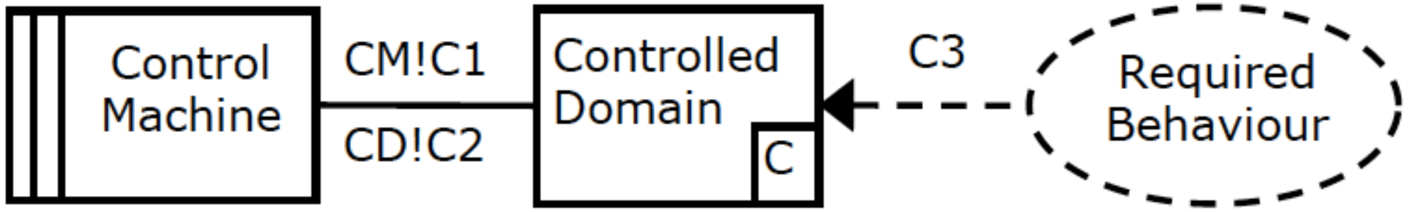
Problem Frames

- What is a problem frame?
- How / why do problem frames help?
- What kinds of frames are defined by Jackson?

The Required Behavior Problem

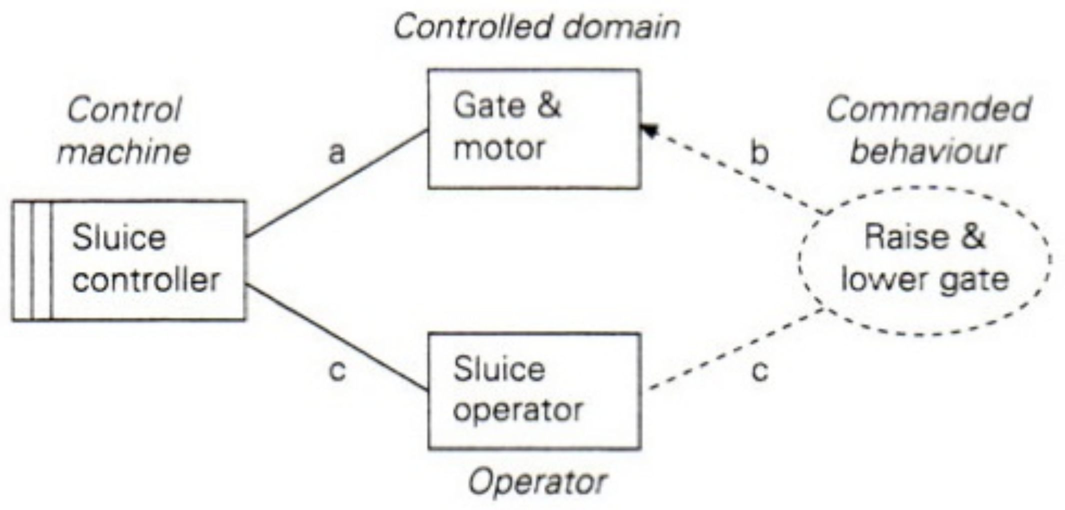
[Jackson 2006]

Application context:
Achieve/maintain a required behaviour in a given problem domain



The Commanded Behavior Problem

Application context:
Achieve a required behaviour in a given problem domain by commands issued by an operator



a: SC! {Clockw, Anti, On, Off} [C1]
GM! {Top, Bottom} [C2]
b: GM! {Open, Shut, Rising, Falling} [C3]
c: SO! {Raise, Lower, Stop} [E4]

[Jackson 2001, p. 112]

The Frame Concern

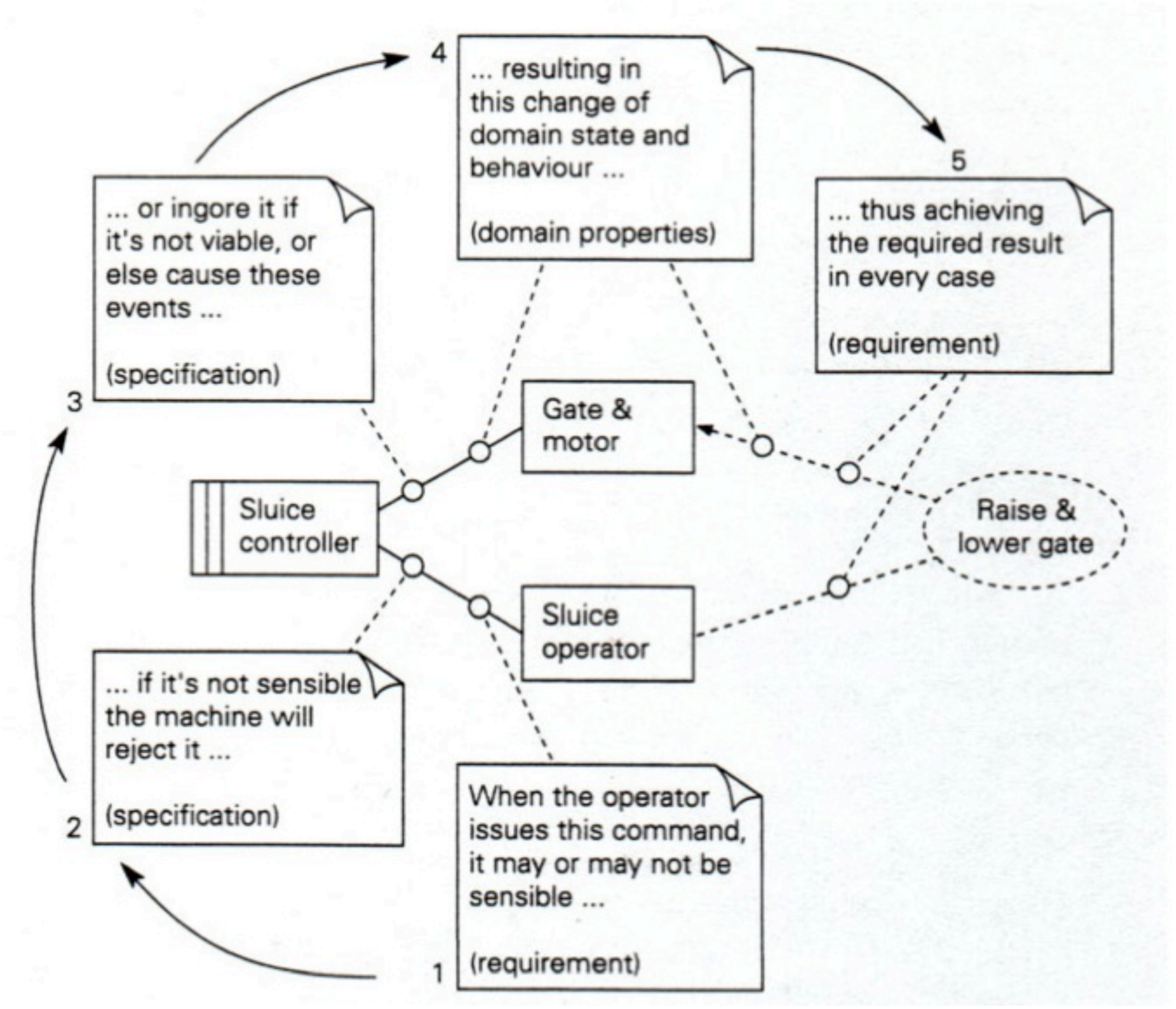
Arguing that

- the machine behavior

together with

- the domain properties

satisfy the requirements

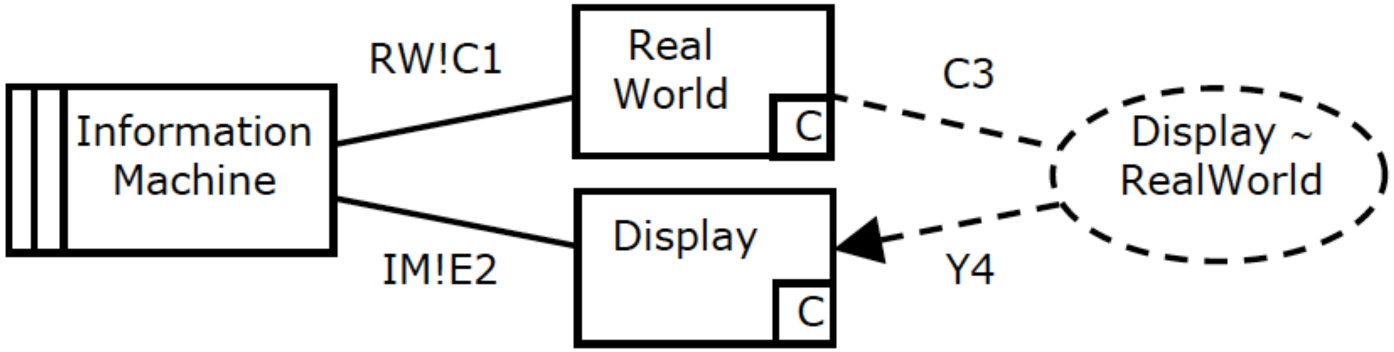

4
... resulting in this change of domain state and behaviour ...
(domain properties)
5
... thus achieving the required result in every case
(requirement)
... or ingore it if it's not viable, or else cause these events ...
(specification)
3
Gate & motor
Sluice controller
Raise & lower gate
Sluice operator
... if it's not sensible the machine will reject it ...
(specification)
2
When the operator issues this command, it may or may not be sensible ...
(requirement)
1

Commanded Behavior Frame Concern [Jackson 2001, p. 113]

The Information Display Problem

[Jackson 2006]

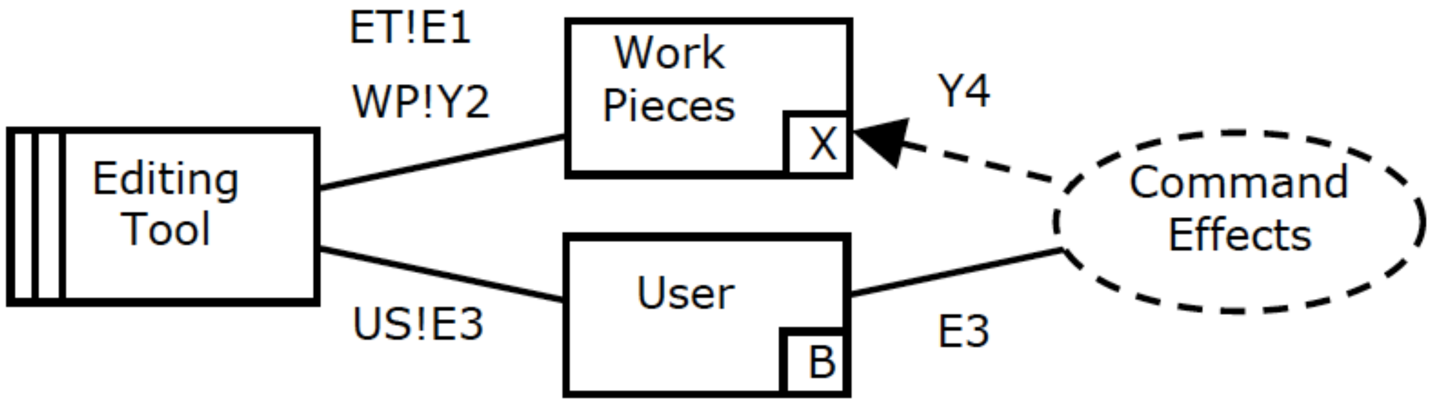
Application context:
Display information about a part of the world



The Workpieces Problem

[Jackson 2006]

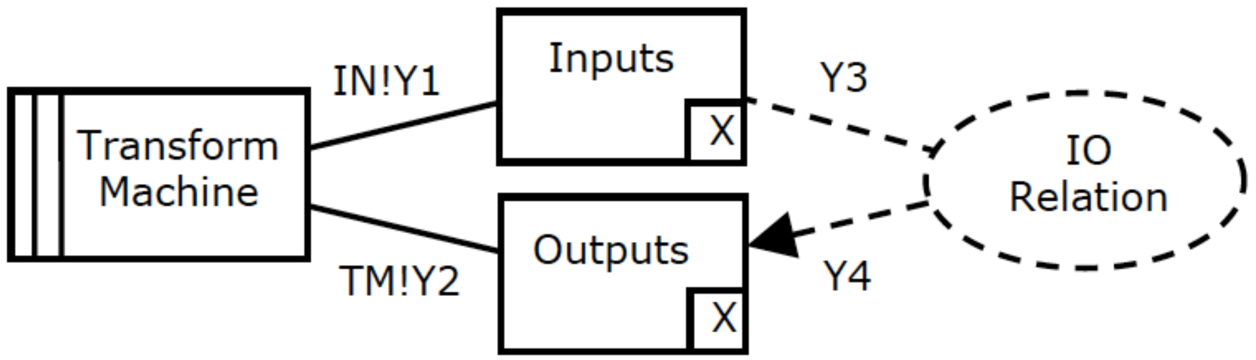
Application context:
Provide a tool for editing a work piece such as text, graphics, etc.



The Transformation Problem

[Jackson 2006]

Application context
Transform input data to output data



Decomposing and recomposing problem frames

- Separate the problem into individual frames
- Model the frames
- Address the frame concern
- Re-integrate the frames into a single design

A hard problem!

Other problem patterns: The four-variable model

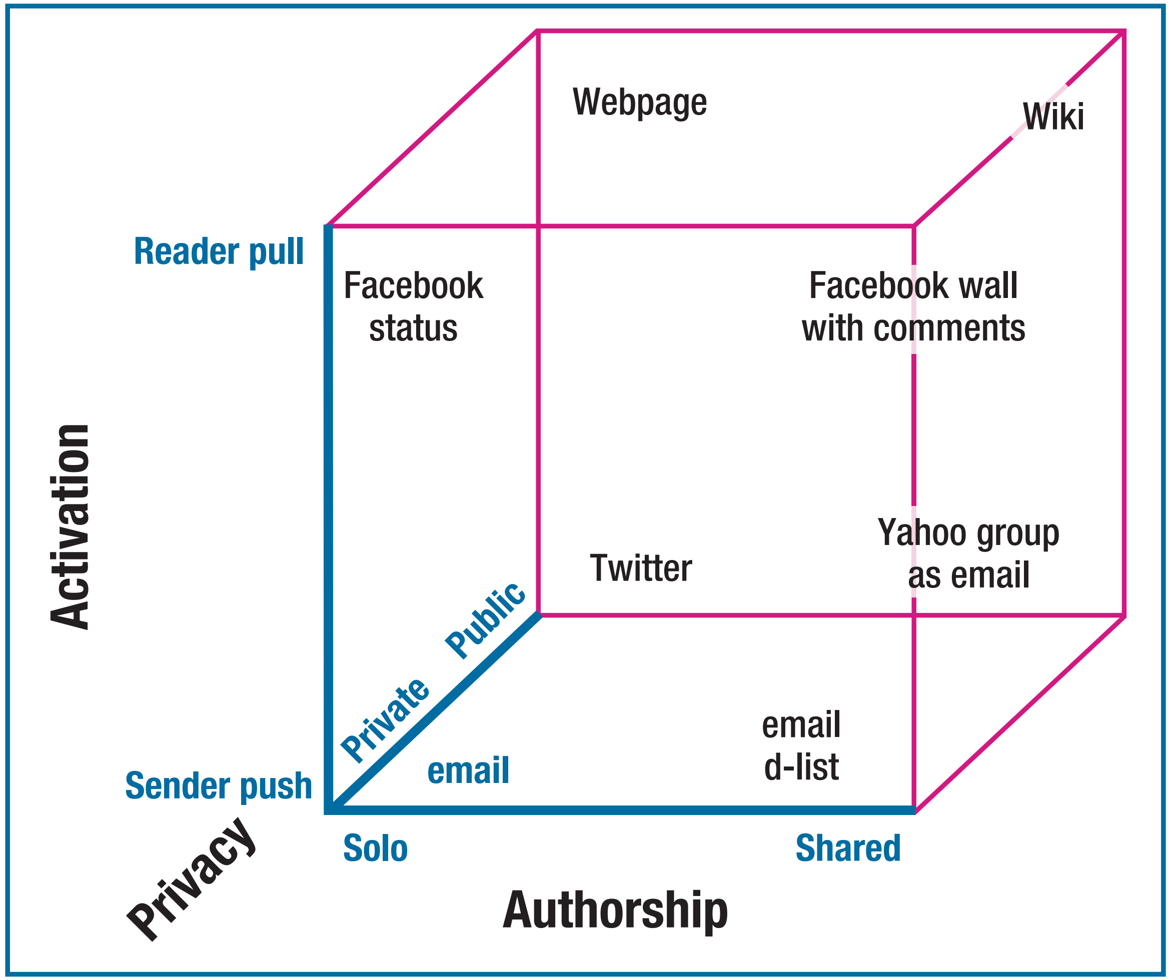
[Parnas and Madey 1995]

- Four variables: monitored, controlled, input, output
- Three relations: NAT (the constraints), REQ (the requirements), SOFT (machine input to machine output)
- Two mappings: MON → INPUT, OUTPUT → CON

$$\begin{tikzcd}
\text{MON} \arrow[r, "\text{REQ}"] \arrow[d, "\text{IN}"'] \arrow[r, leftrightarrow, bend left=60, "\text{NAT}"] & \text{CON} \\
\text{INPUT} \arrow[r, "\text{SOFT}"'] & \text{OUTPUT} \arrow[u, "\text{OUT}"']
\end{tikzcd}$$

Design Spaces

- What?
- Why?
- How?



[Shaw 2012]

Normal vs. Radical Design

P36 (1935)

P51 (1940)

Me 262(1942)

Design Process

- A rational design process?
- Innovation
- Mature systems

References

Booch, G. (2012). The Professional Architect. *IEEE Software* **29**, 1 (Jan 2012). 12-13.

Brooks, F.P. (2010). *The Design of Design: Essays from a Computer Scientist*. Boston, MA: Pearson Education.

Herrmannsdörfer, M., S. Konrad, B. Berenbach (2008). Tabular Notations for State Machine-Based Specifications. *Crosstalk*, issue 3/2008. 18-23.

Jackson (2001). *Problem Frames: Analysing and Structuring Software Development Problems*. Harlow: Addison-Wesley.

M. Jackson (2005). Problem Frames and Software Engineering. *Information and Software Technology* **47**, 14 (Nov 2005). 903-912.

M. Jackson (2006). *A Tutorial on Software Development Problem Frames*. BCS RESG Problem Frames Day, The Open University, Milton Keynes, UK. 10 May 2006.

Krasner, G.E., S.T. Pope (1988). A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming* **1**, 3. 26-49.

Kruchten, P.B. (1995). The 4+1 View Model of Architecture. *IEEE Software* **12**, 6 (Nov 1995). 42-50.

Meyer, B. (1992). Applying “Design by Contract”. *IEEE Computer* **25**, 10 (Oct. 1992). 40-51.

Parnas, D.L. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. *Com-munications of the ACM* **15**, 12 (Dec. 1972). 1053-1058.

Parnas, D.L. and Clements, P.C. (1986). A Rational Design Process: How and Why to Fake It. *IEEE Transactions on Software Engineering* **SE-12**, 2 (Feb 1986). 251-257.

Parnas, D.L., J. Madey (1995). Functional Documents for Computer Systems. *Science of Computer Programming* **25**, 1 (October 1995). 41-61.

Shaw, M. (2012). The Role of Design Spaces. *IEEE Software* **29**, 1 (Jan 2012). 46-50.