



Practically

# Perfect Problem Frames

Problem Frames and Problem Orientation

Jon G. Hall  
CRC, The Open University

with Lucia Rapanotti  
Michael Jackson  
Andrés Silva  
Bashar Nuseibeh  
Robin Laney  
Shahbaz Ali  
John Brier  
Derek Mannering  
Zhi Li





# Agenda

---

- ✻ Problem orientation
- ✻ Problem oriented development processes
  - ✻ Enabling development through problem transformation
- ✻ How attitudes to risk can influence development
  - ✻ as part of a wider adequacy argument building framework
- ✻ Expanding applicability of problem orientation
  - ✻ How far can it go?





# Problem Orientation





Problems, problems...





---

“We can start with the obvious statement that  
*engineering is a problem solving activity.*”

Walter Vincenti (1990)  
*‘What engineers know and how they know it’*  
Johns Hopkins Press





# A view of software engineering

---

‘Engineering refers to the practice of *organising the design and construction of any artefact* which *transforms the physical world* around us *to meet some recognised need*’

G.F.C. Rogers (1983)  
‘*The Nature of Engineering*’  
Macmillan Press





# A view of software engineering

‘Engineering refers to the practice of *organising the design and construction of any artefact* which *transforms the physical world* around us *to meet some recognised need*’

G.F.C. Rogers (1983)  
‘*The Nature of Engineering*’  
Macmillan Press

*Recognised need*

*Physical World*

*Artifice*



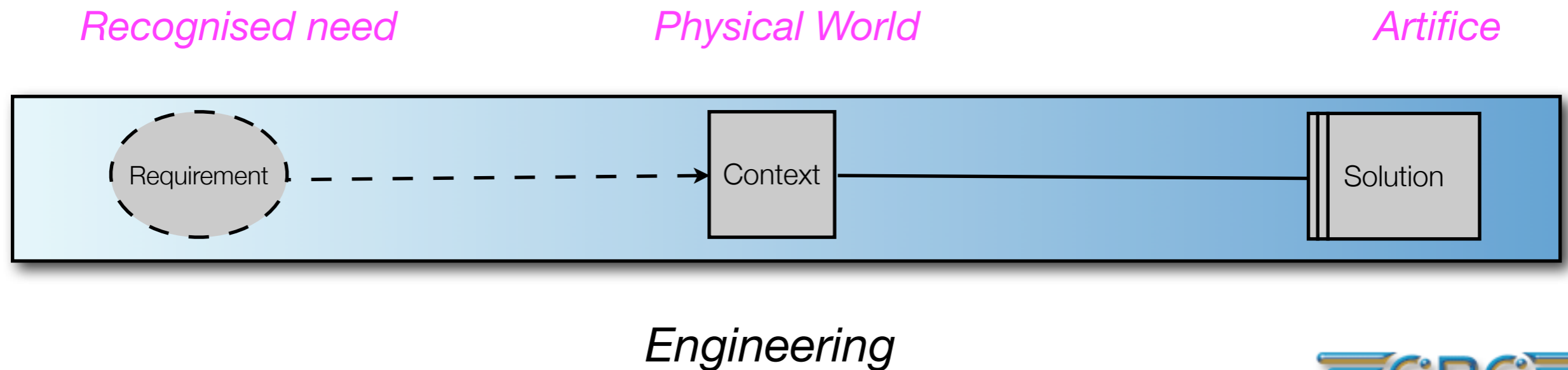
*Engineering*



# A view of software engineering

‘Engineering refers to the practice of *organising the design and construction of any artifice* which *transforms the physical world* around us *to meet some recognised need*’

G.F.C. Rogers (1983)  
‘*The Nature of Engineering*’  
Macmillan Press

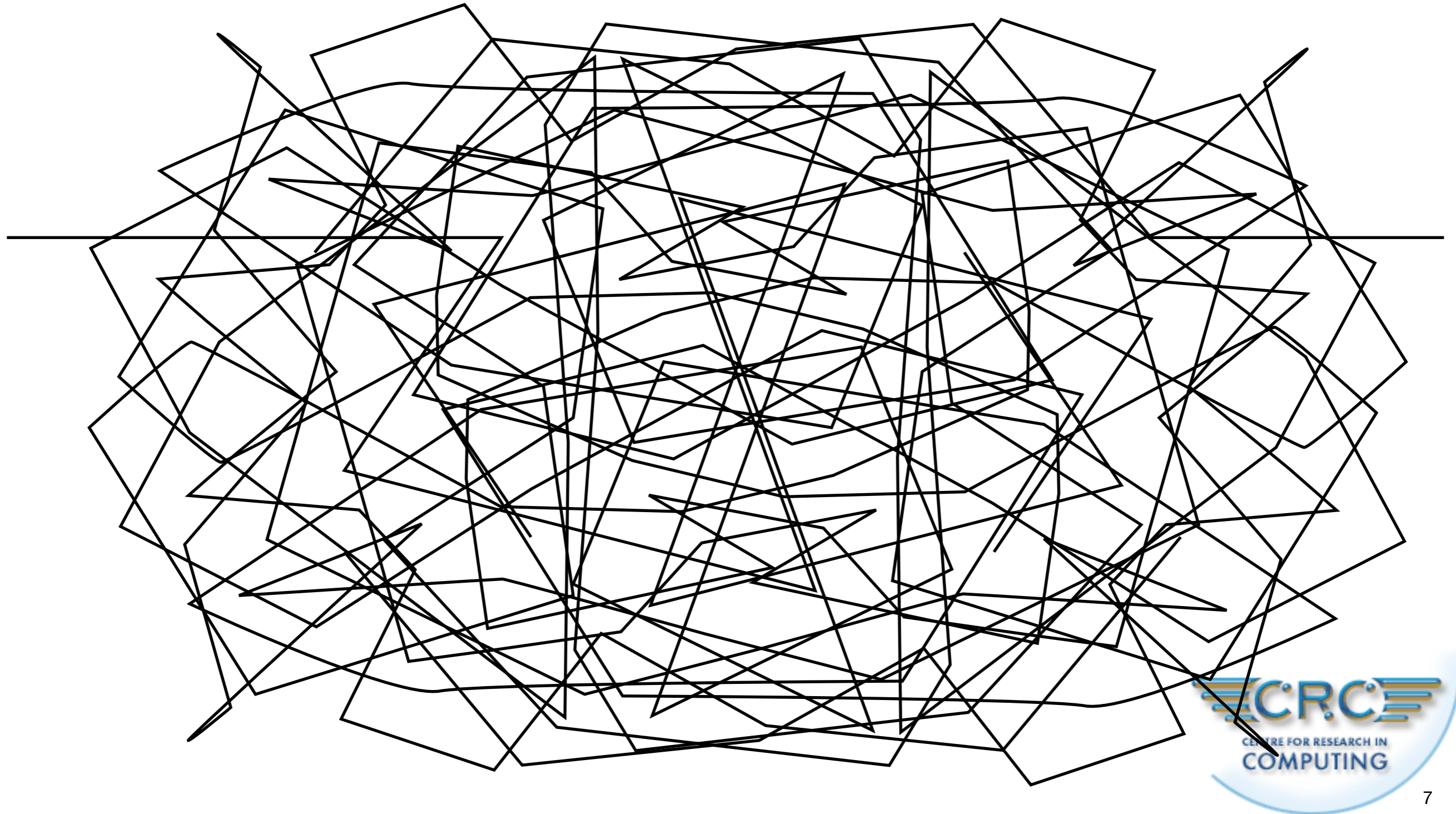






# The real picture ...

---





# Problem orientation

---

- ✻ Our challenge is to understand a problem and to create a fit-for-purpose solution

What is a problem?

How to develop problems using transformations

How to show fitness for purpose

How to make better use of expertise

What are the limits of extensibility?

Which tools are needed?





# Problems & Problem transformation





# A farmer writes ...

*The Farm  
9<sup>th</sup> May, 2006*

*Dear Sir/Madam,*

*I have a problem!*

*I have a sluice gate that was installed three years ago that controls water flow in my fields.*

*I need to be able to raise, lower and stop the sluice gate.*

*Water should flow for ten minutes every three hours.*

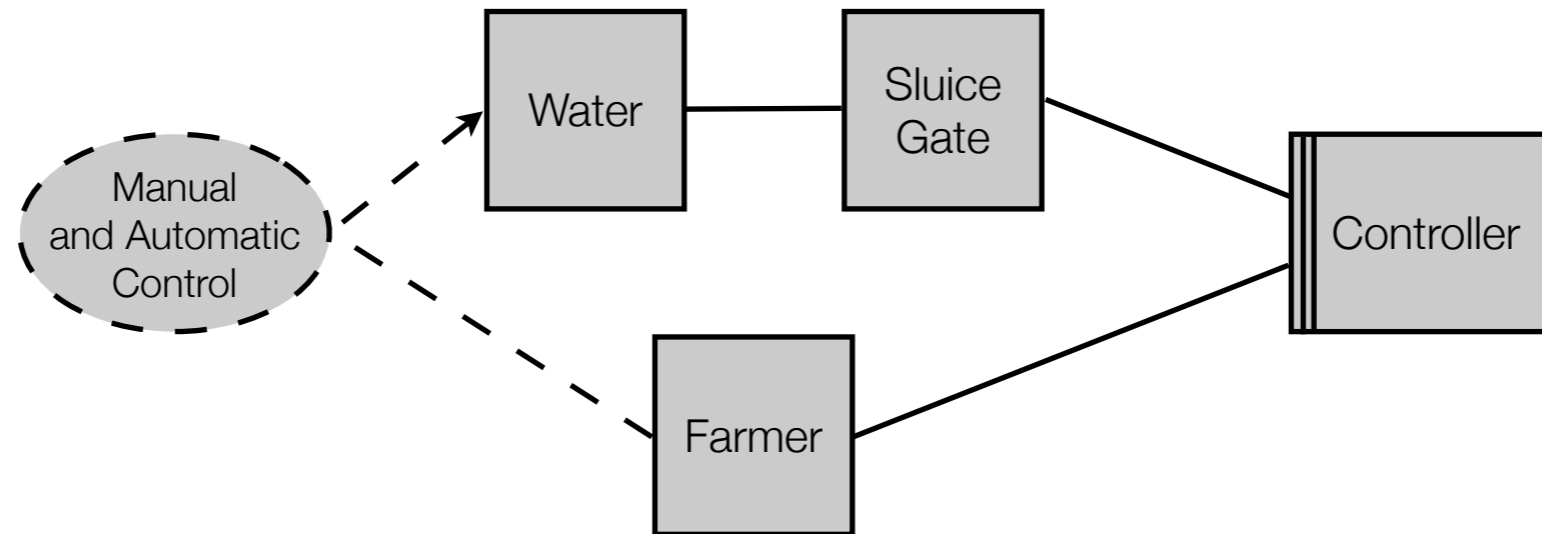
*Can you help me?*

*Yours sincerely,*

*A. Farmer*



# Sluice gate problem ...



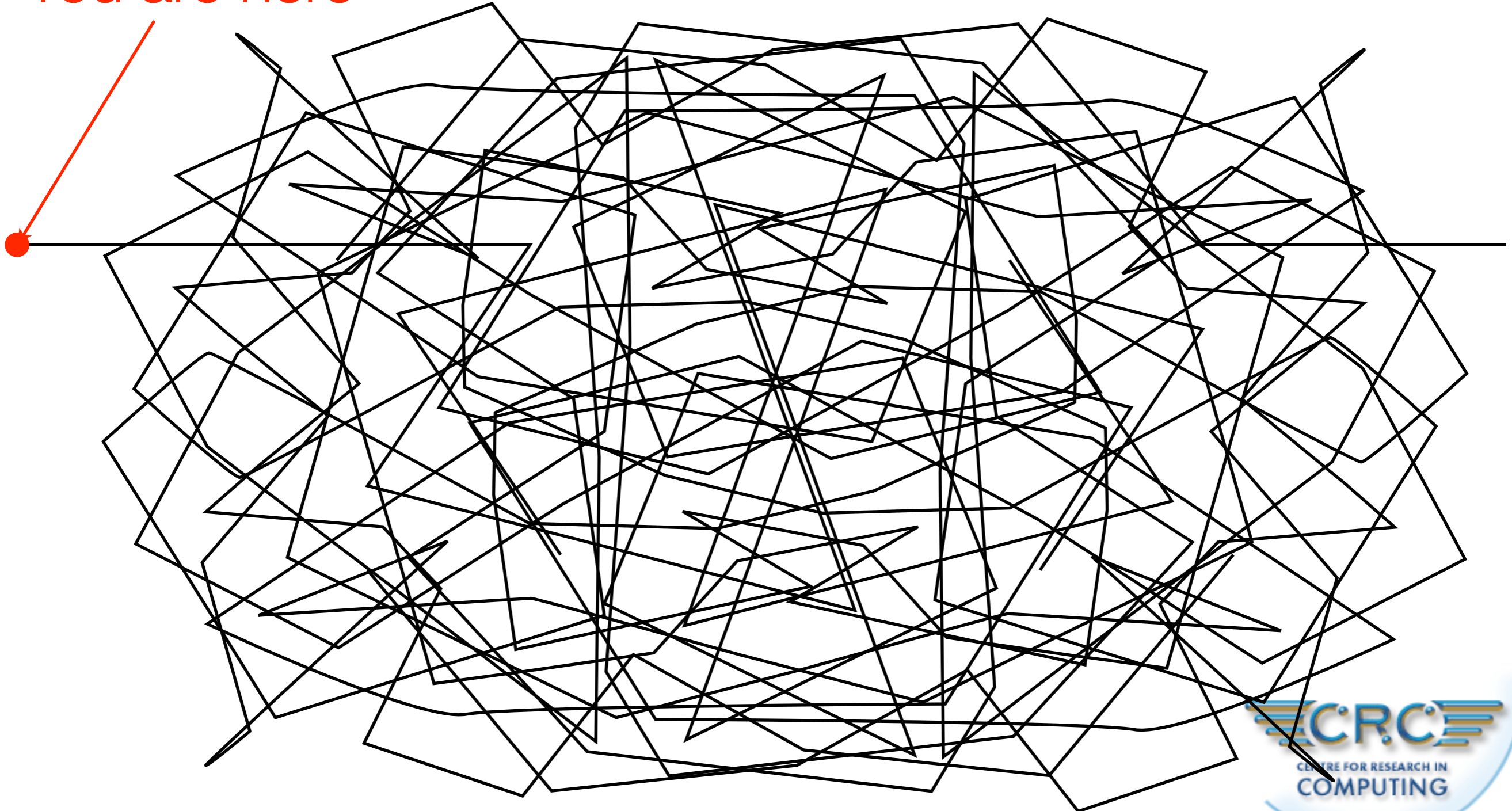
## The problem...

<i>Sluice Gate</i>	installed three years ago
<i>Water</i>	flows in the field, controlled by the Sluice Gate
<i>Manual and Automatic Control</i>	The Farmer should be able to raise, lower and stop the Sluice Gate. The Sluice Gate should allow water to flow in the field for 10 minutes every 3 hours
<i>Farmer</i>	operates the Sluice Gate in manual mode
<b>Controller</b>	<b>to be found</b>



# The real picture ...

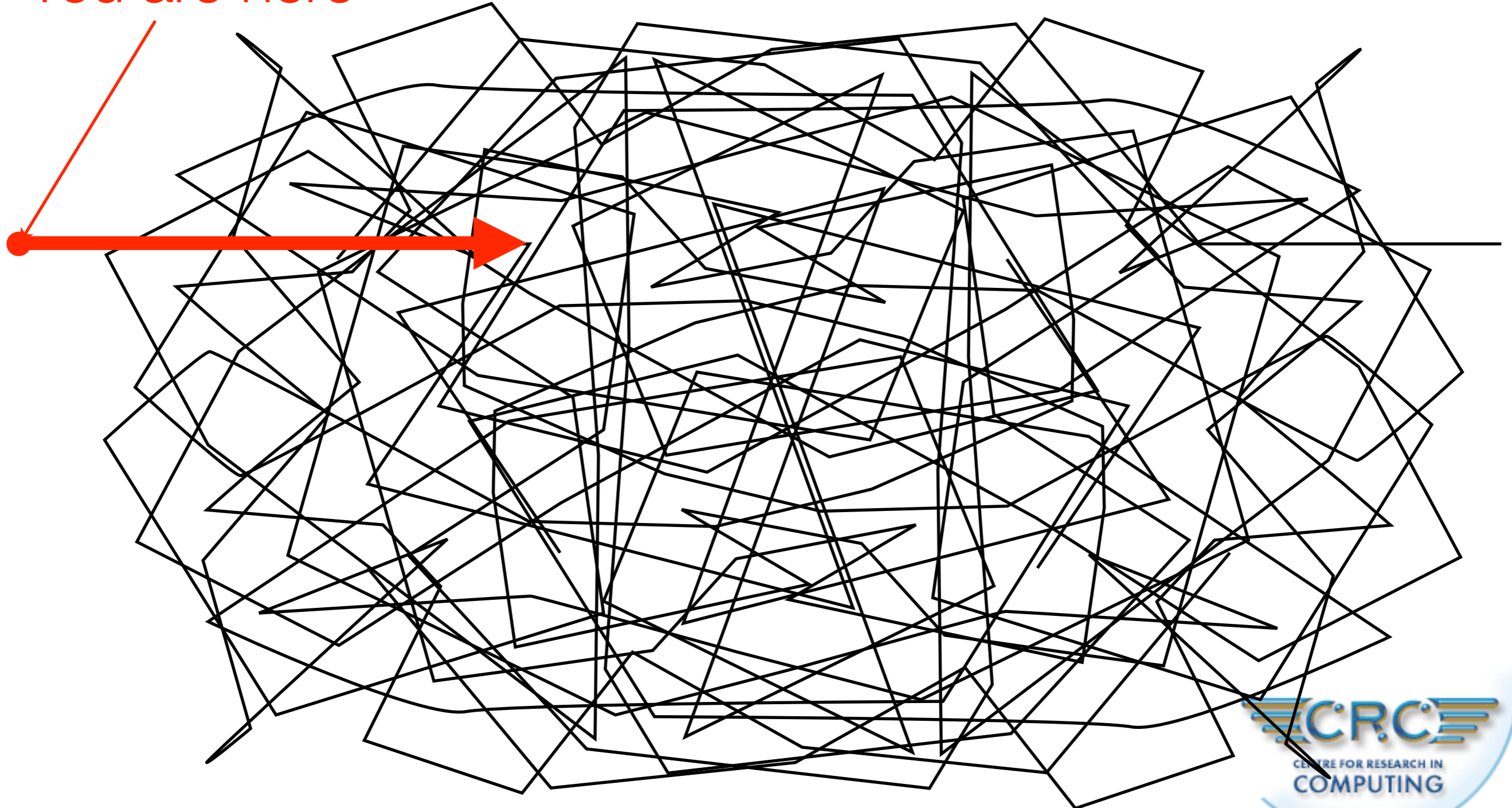
You are here





# The real picture ...

You are here





# Transforming problems ...

---

- ✻ Restatement to make it more workable
  - ✻ interpretation
- ✻ Decide solution structure
  - ✻ developing code
  - ✻ architecture
- ✻ Giving different perspectives on the problem
  - ✻ logical, physical
  - ✻ different stakeholders

[HR05] Jon G. Hall and L. Rapanotti. A framework for software problem analysis. Technical Report 2005/05, Department of Computing, The Open University, 2005.



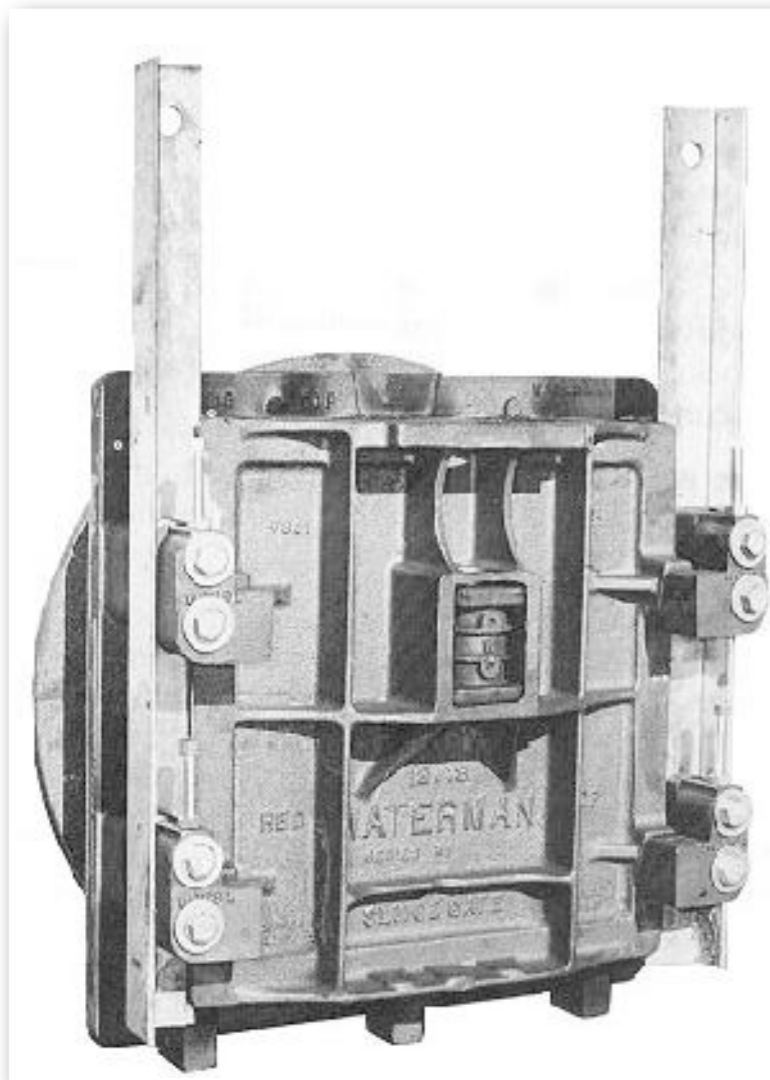


# Interpretation: using experience ...

*“installed three years ago”*

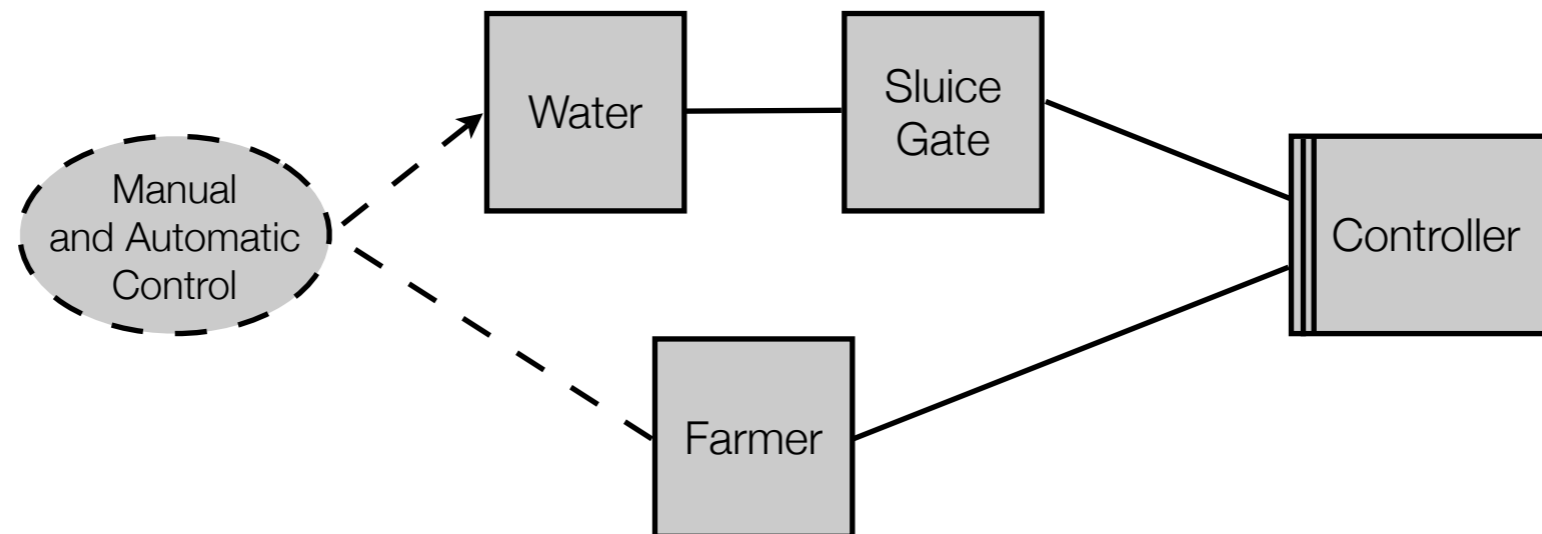
*“SGVert model was sold four years ago and was the most popular sluice gate”*

*The gate is the SGVert model*



*SGVert model*

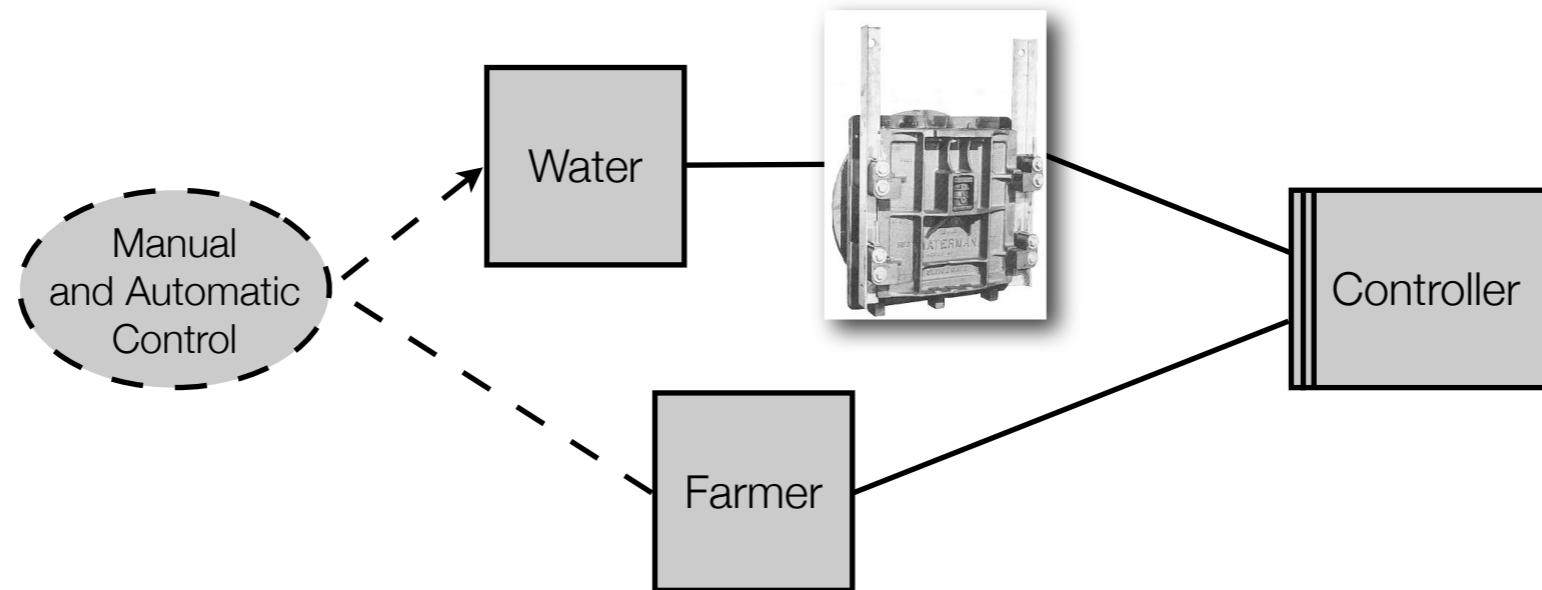
# Sluice gate problem ...



## The problem...

<i>Sluice Gate</i>	installed three years ago
<i>Water</i>	flows in the field, controlled by the Sluice Gate
<i>Manual and Automatic Control</i>	The Farmer should be able to raise, lower and stop the Sluice Gate. The Sluice Gate should allow water to flow in the field for 10 minutes every 3 hours
<i>Farmer</i>	operates the Sluice Gate in manual mode
<b>Controller</b>	<b>to be found</b>

# Sluice gate problem transformed...



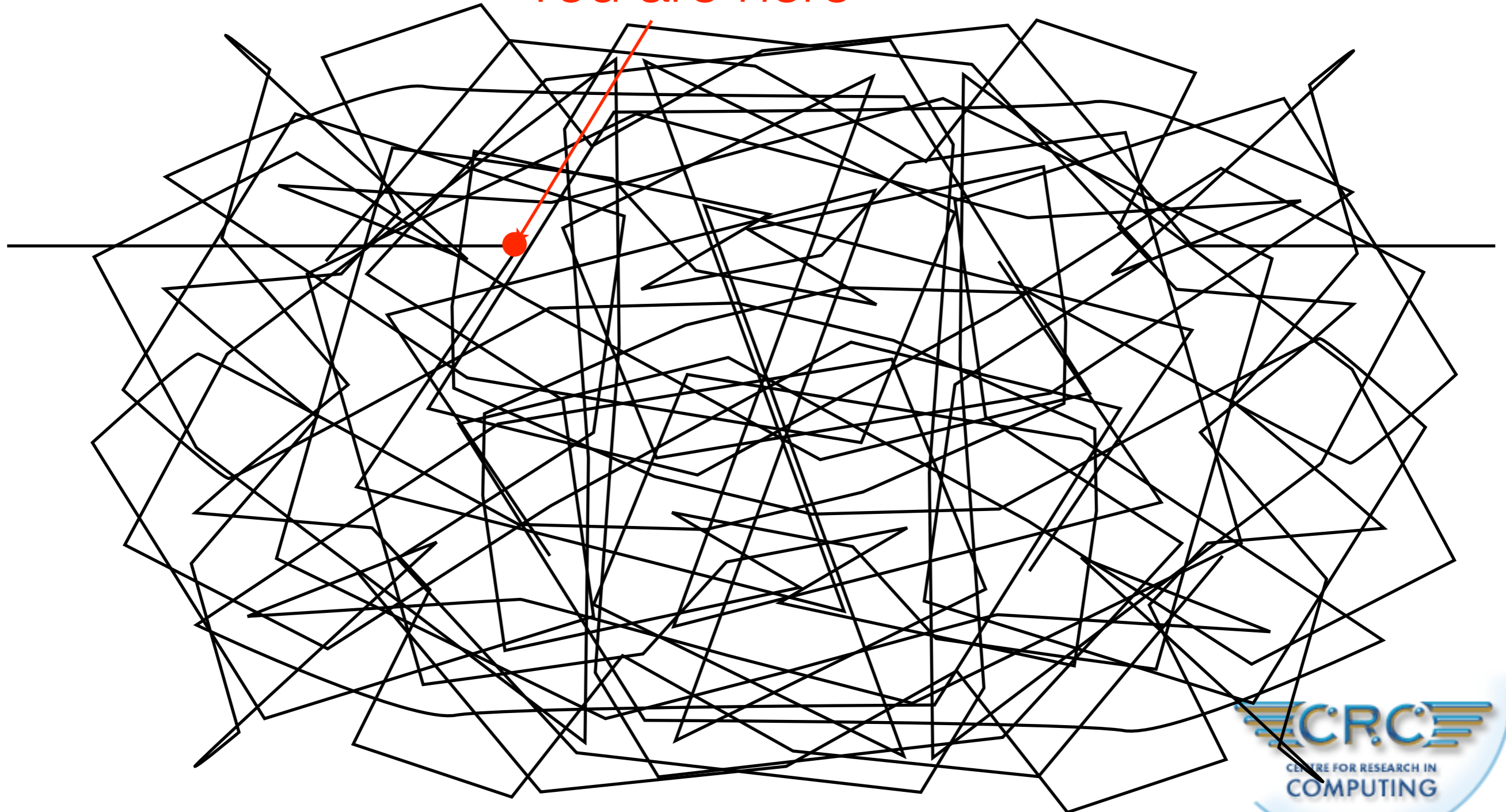
## the story unfolds...

<i>Sluice Gate</i>	SGVert model of Sluice Gate, with behaviour specified by the manual
<i>Water</i>	flows in the field, controlled by the Sluice Gate
<i>Manual and Automatic Control</i>	The Farmer should be able to raise, lower and stop the Sluice Gate. The Sluice Gate should allow water to flow in the field for 10 minutes every 3 hours
<i>Farmer</i>	operates the Sluice Gate in manual mode
<b>Controller</b>	<b>to be found</b>

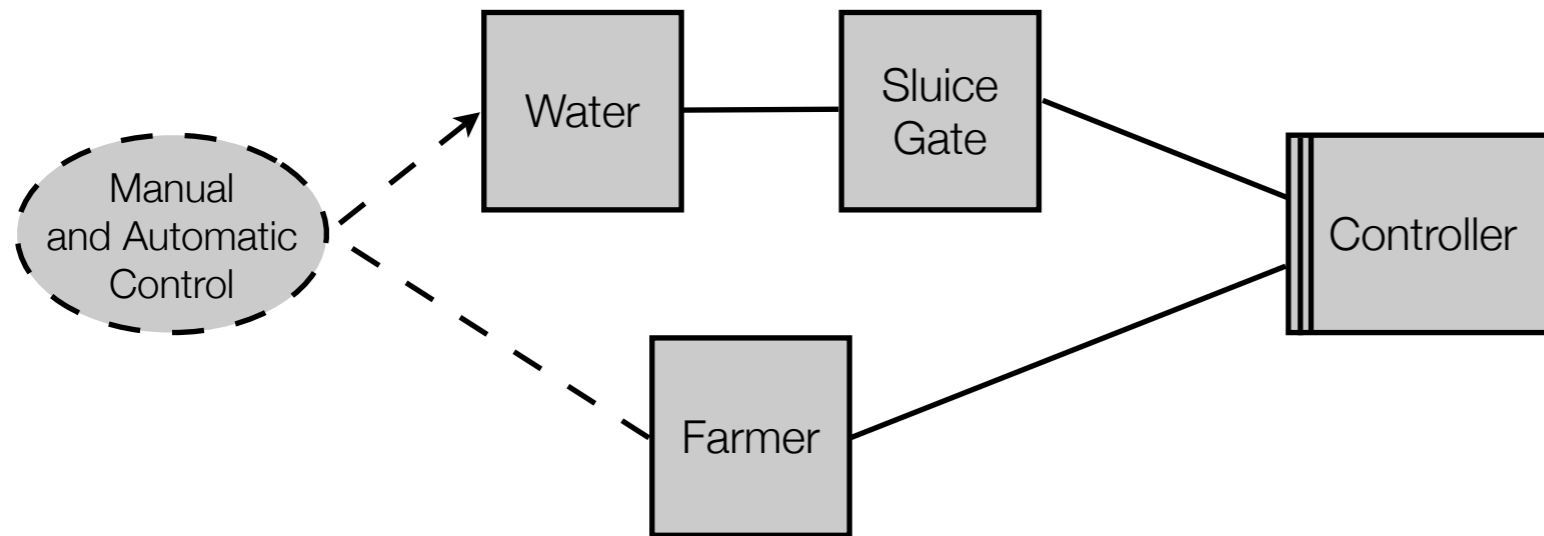


# The real picture ...

You are here



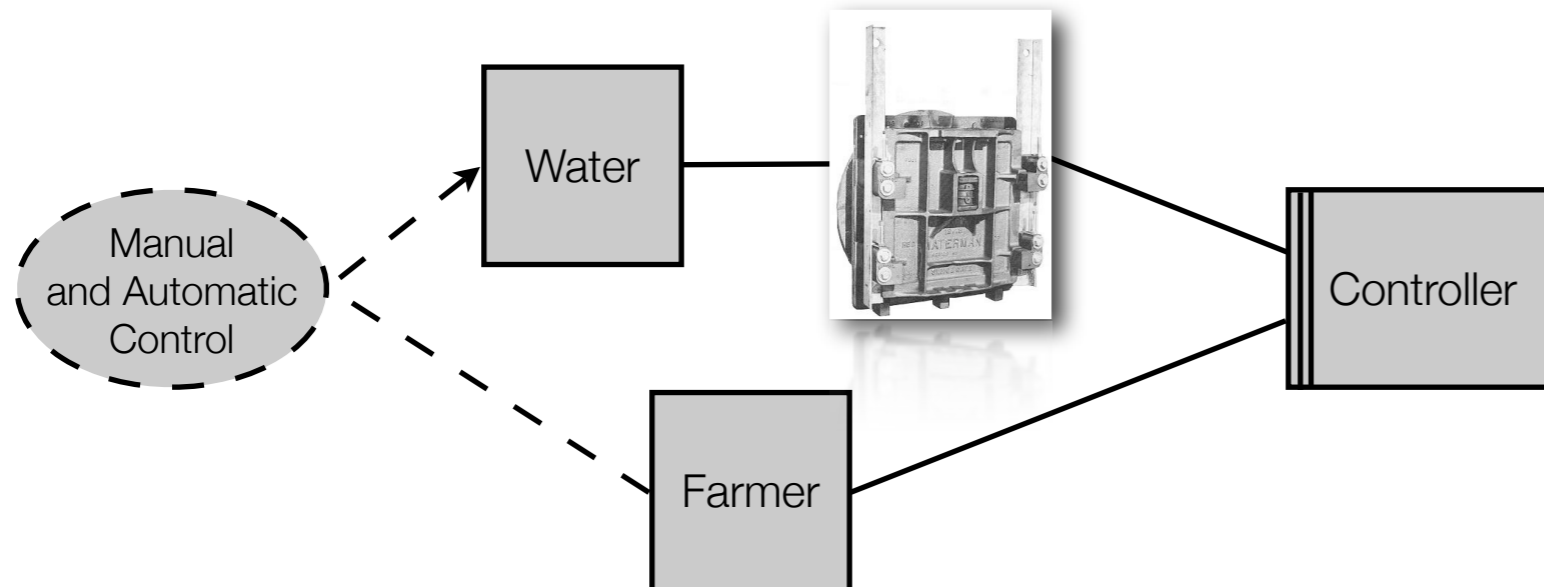
# The transformation



To solve this...

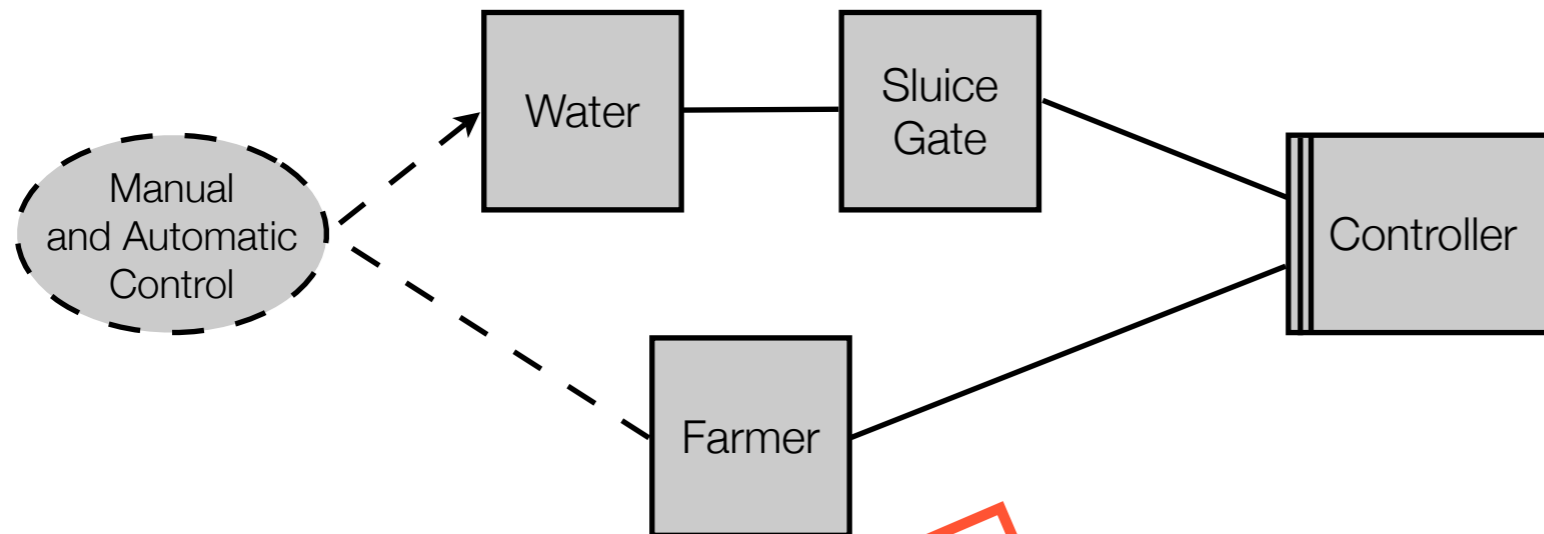


...transform the problem...

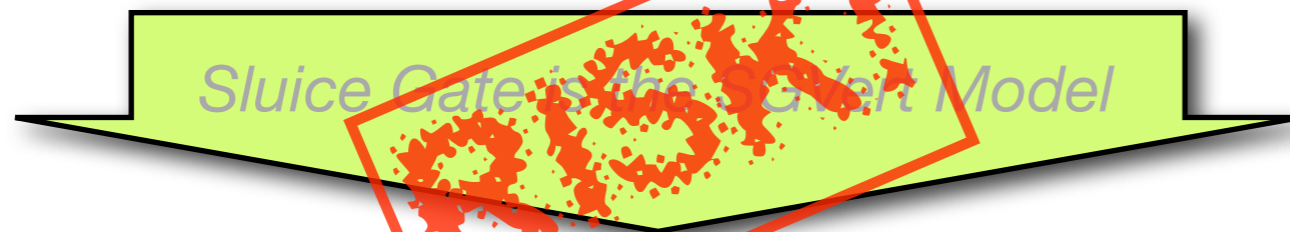


... and solve ...

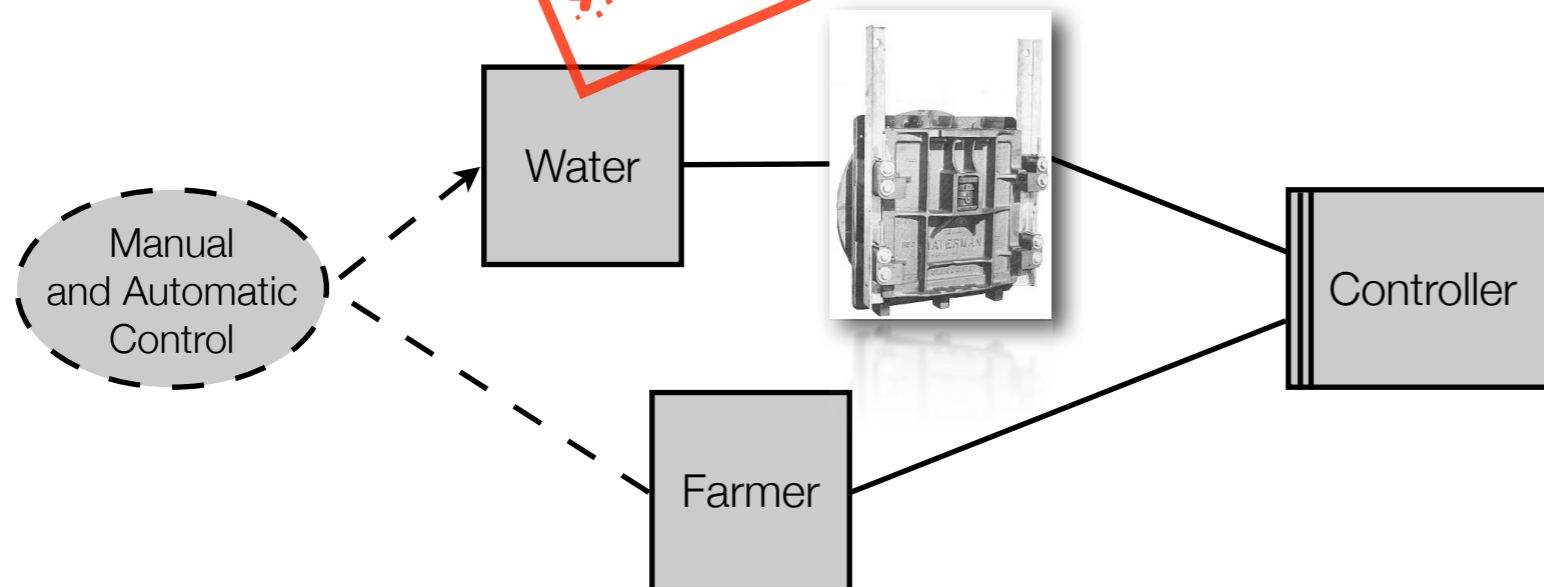
# The transformation



To solve this...



...transform the problem...



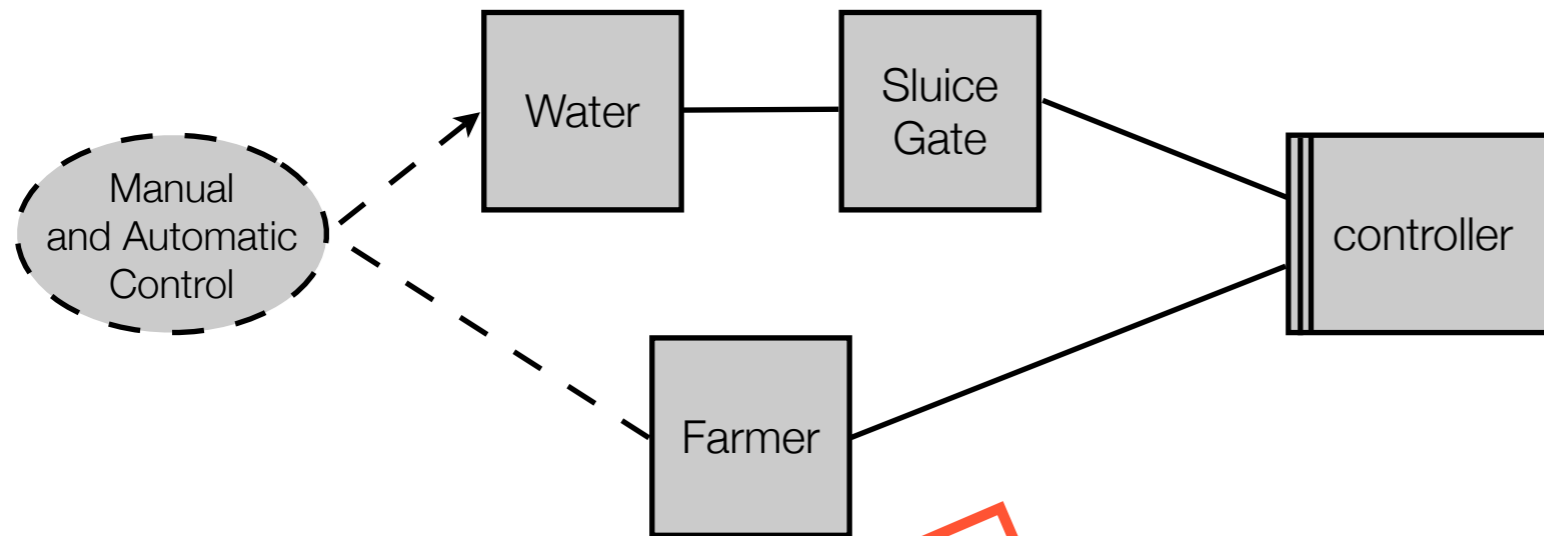
... and solve ...

# Are there risks in the transformation ...

---

- ✻ What if the farmer was wrong about the date of installation?
- ✻ What if the sluice gate was five years old when it was installed?
- ✻ What if the farmer bought the Italian SGVert Model of sluice gate?
  
- ✻ What are the risks of a *wrong* assumption?
- ✻ What are the risks of *running ahead* of justification?
- ✻ Who should be responsible?
  
- ✻ What can we do to manage the risks?

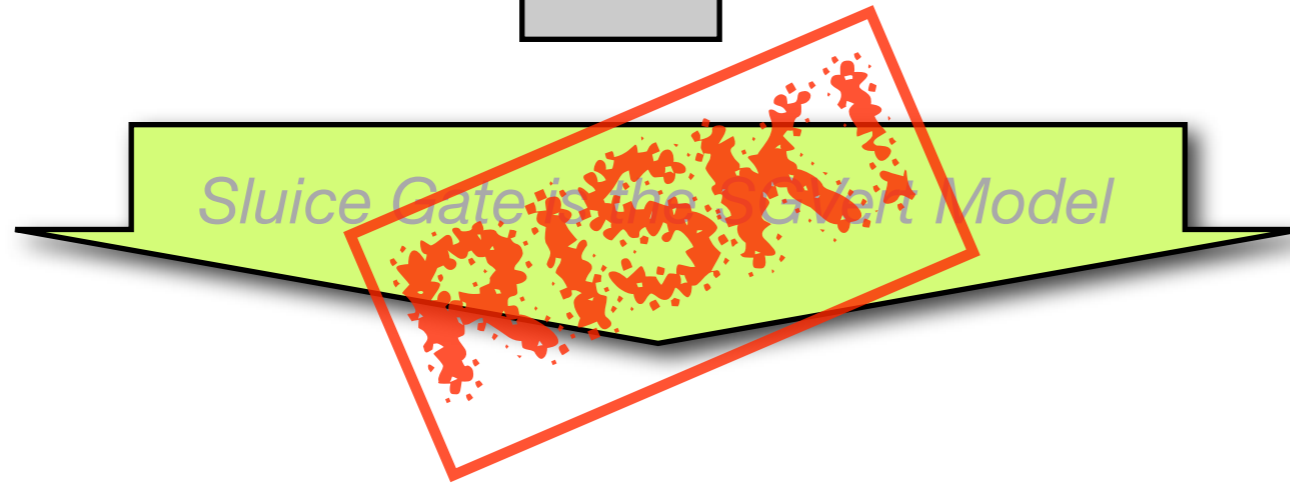
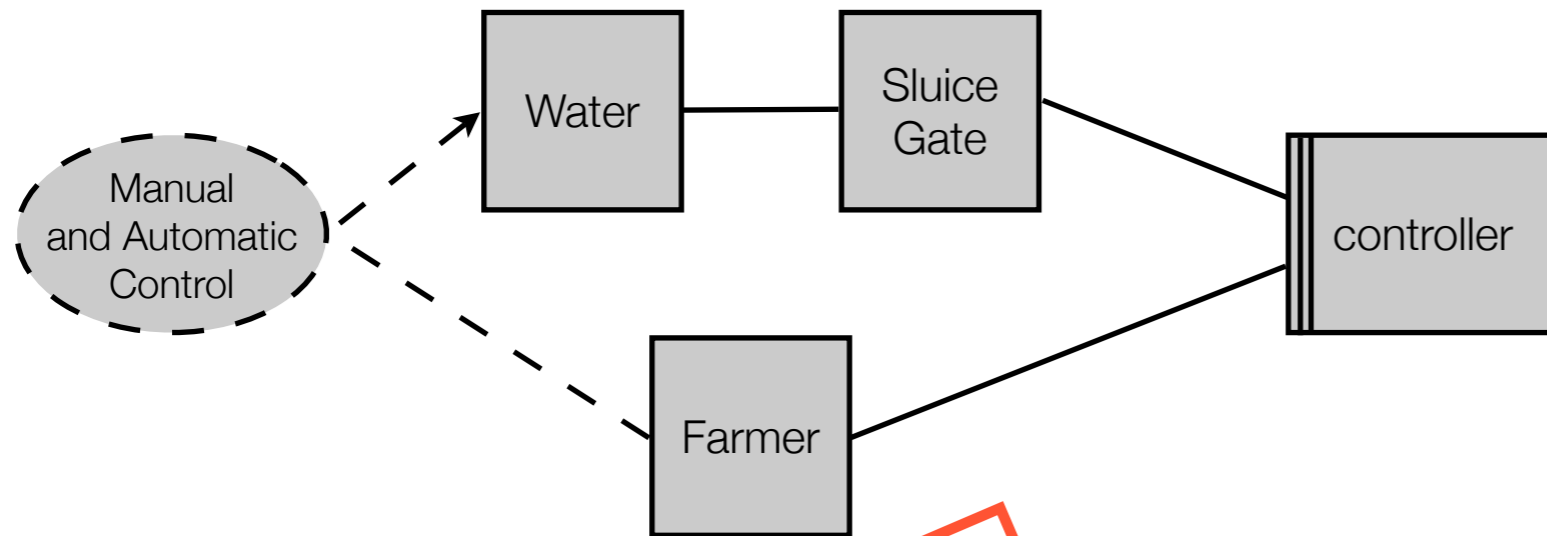
# Avoid the risk



*Sluice Gate is the ... Model*

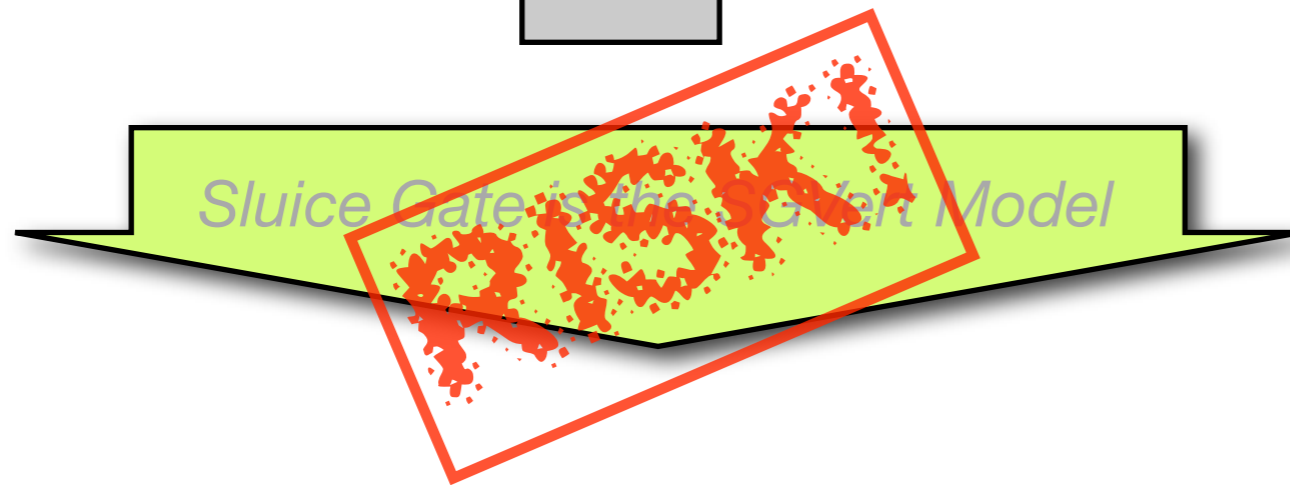
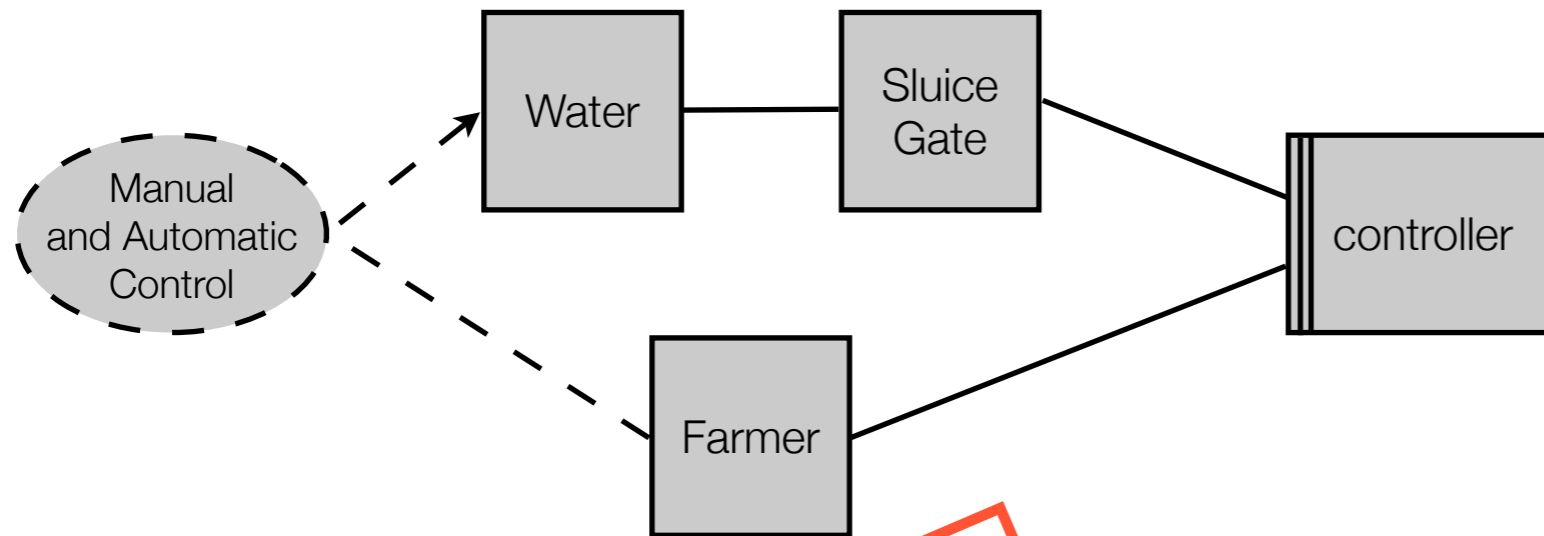


# Avoid the risk



inspect

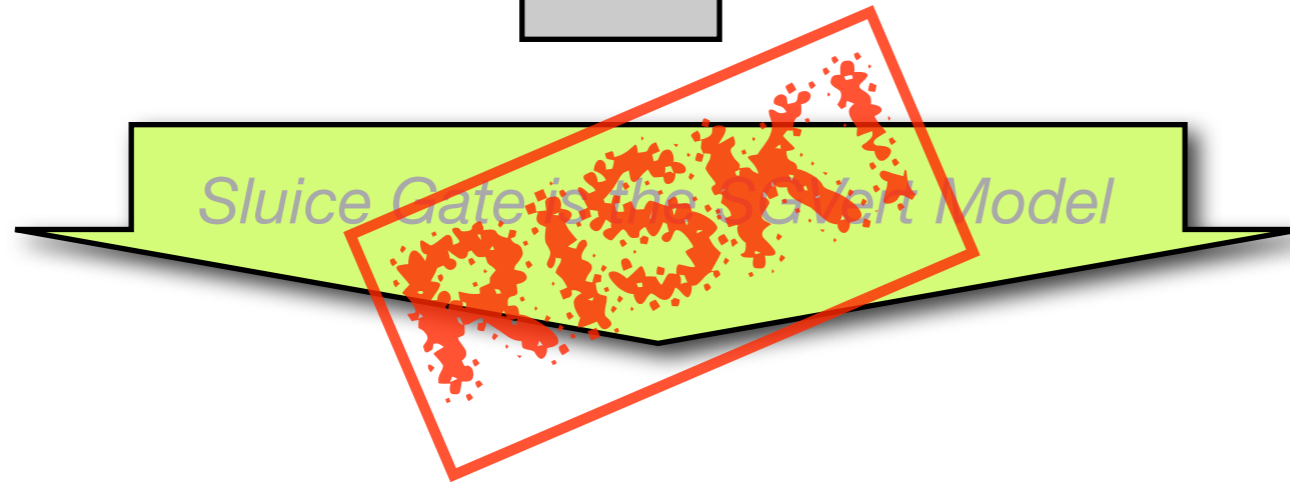
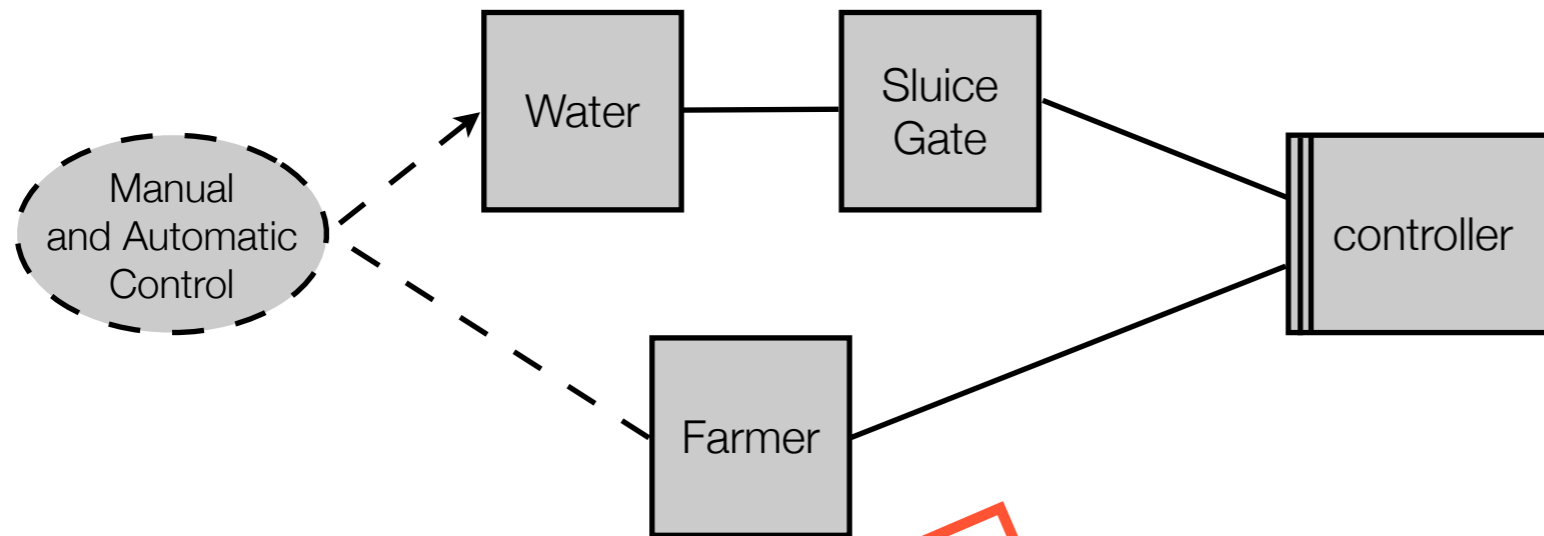
# Avoid the risk



inspect

☒ Record dead end, iterate

# Avoid the risk

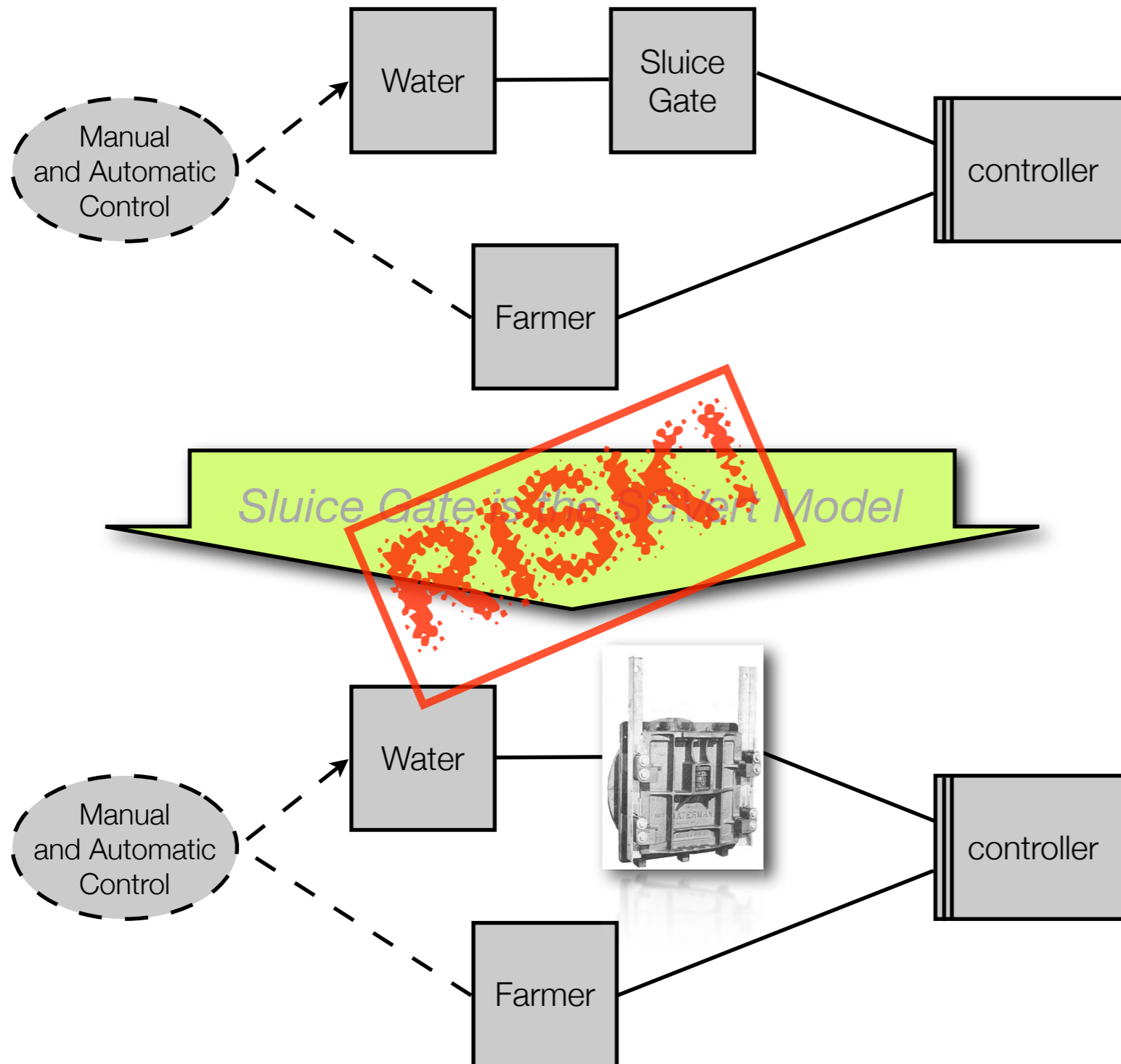


inspect

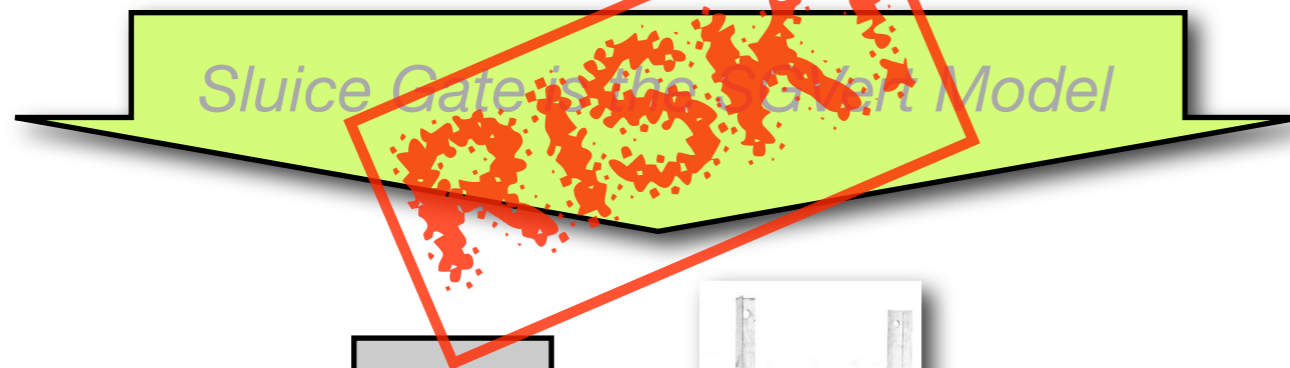
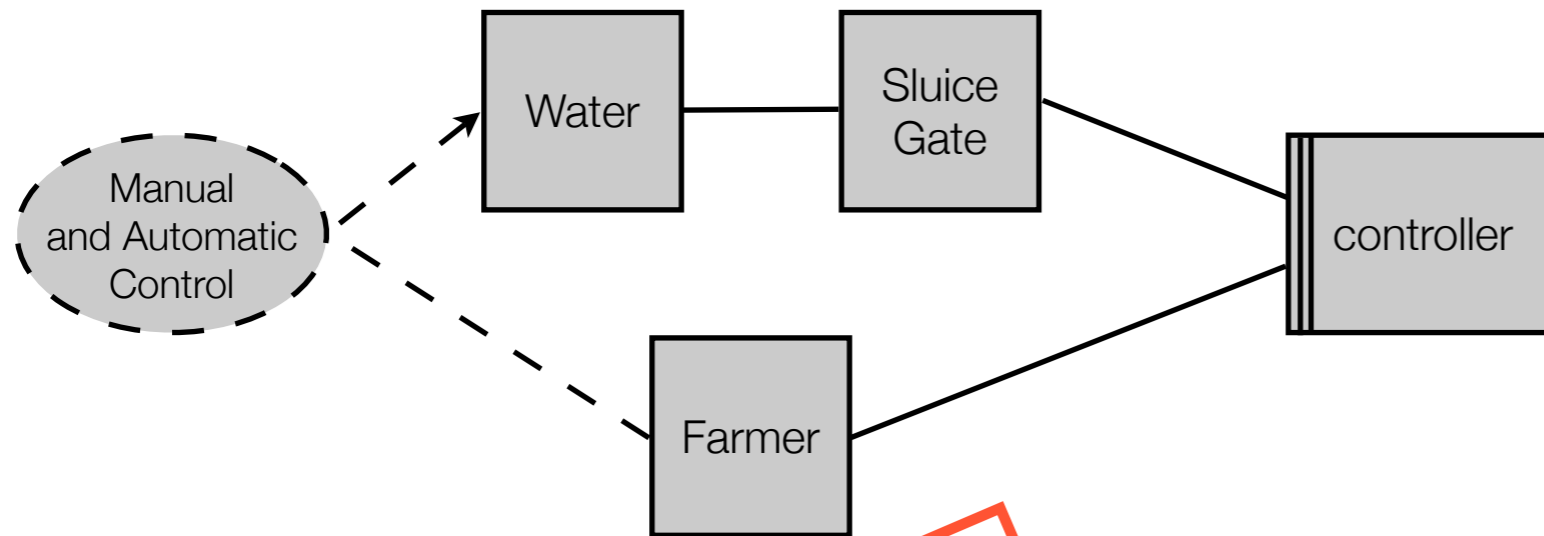
Record dead end, iterate

Continue development

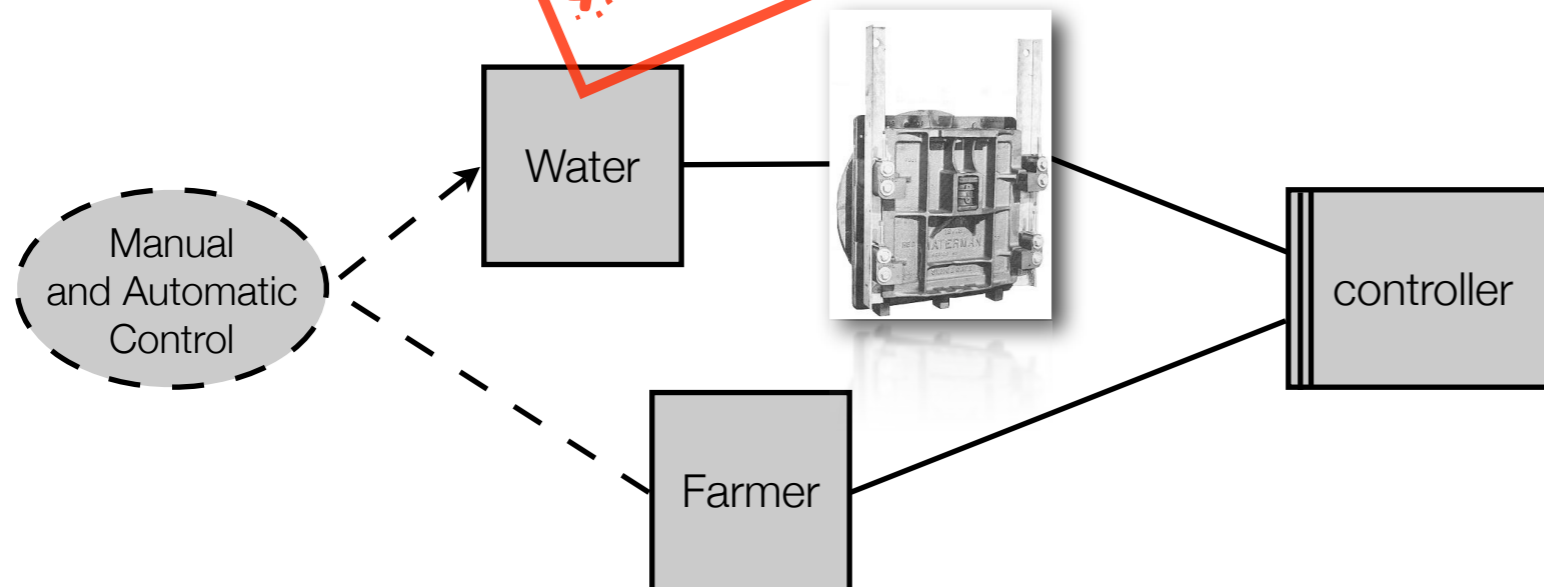
# Ignore the risk



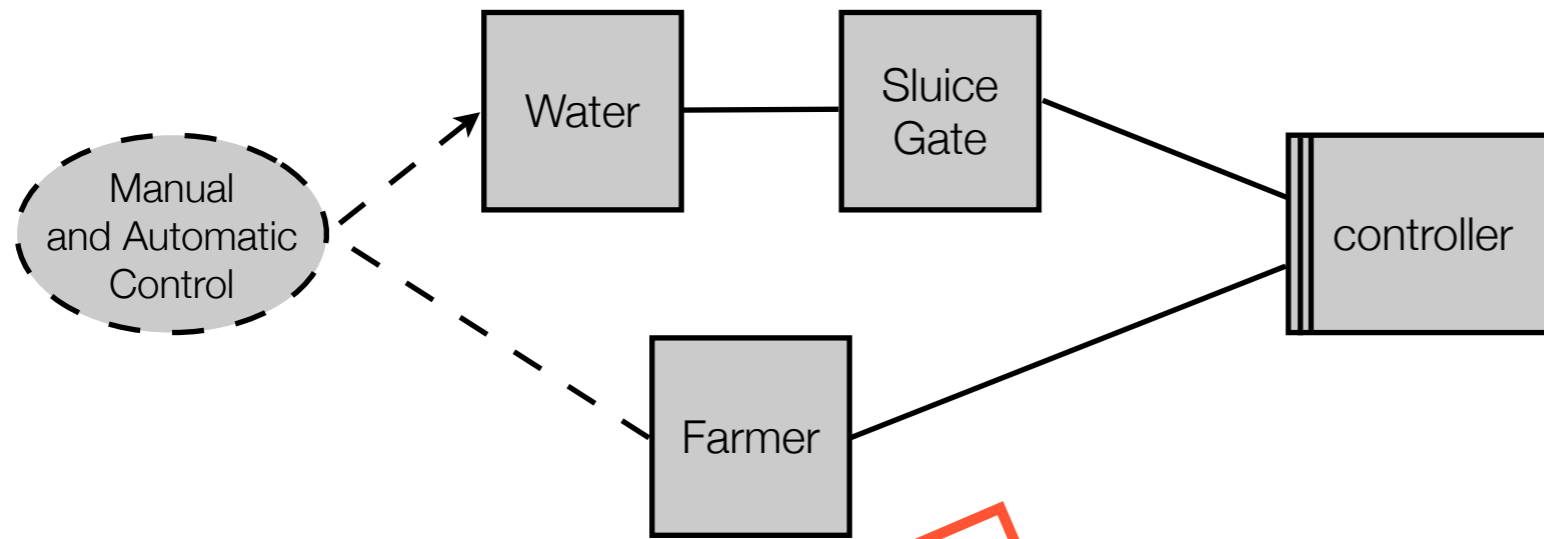
# Ignore the risk



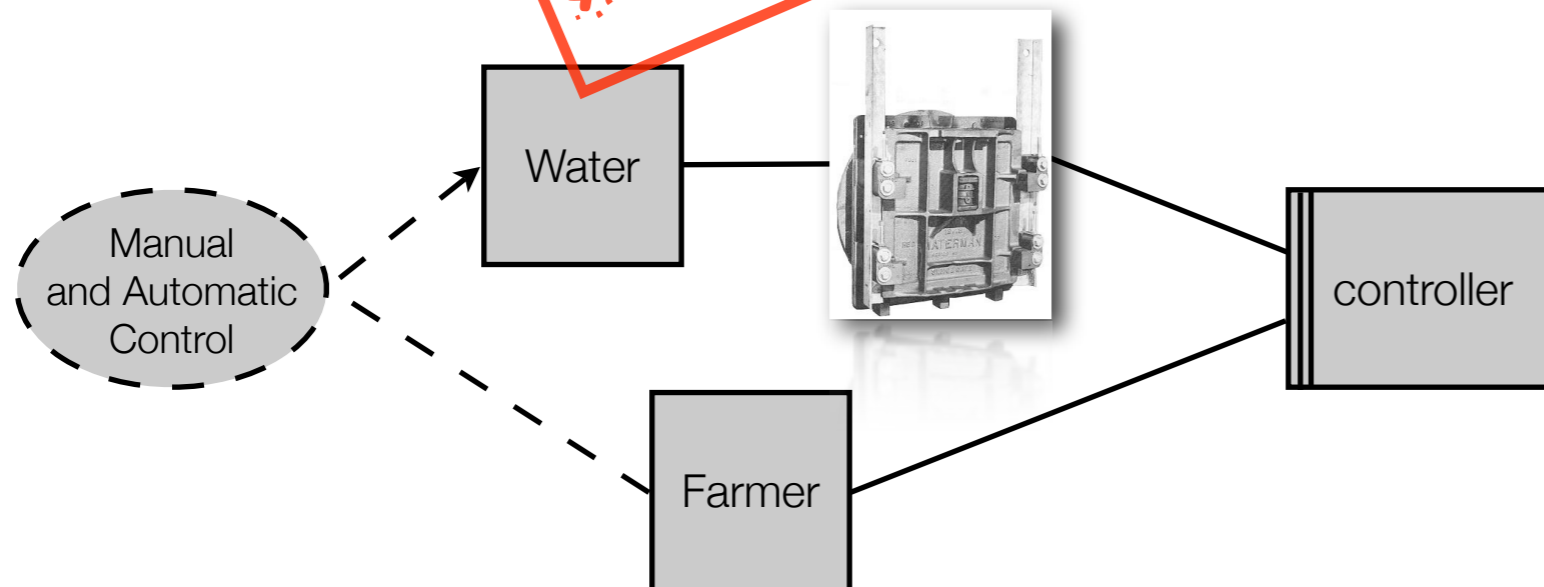
unconfirmed assumption



# Running ahead of the risk

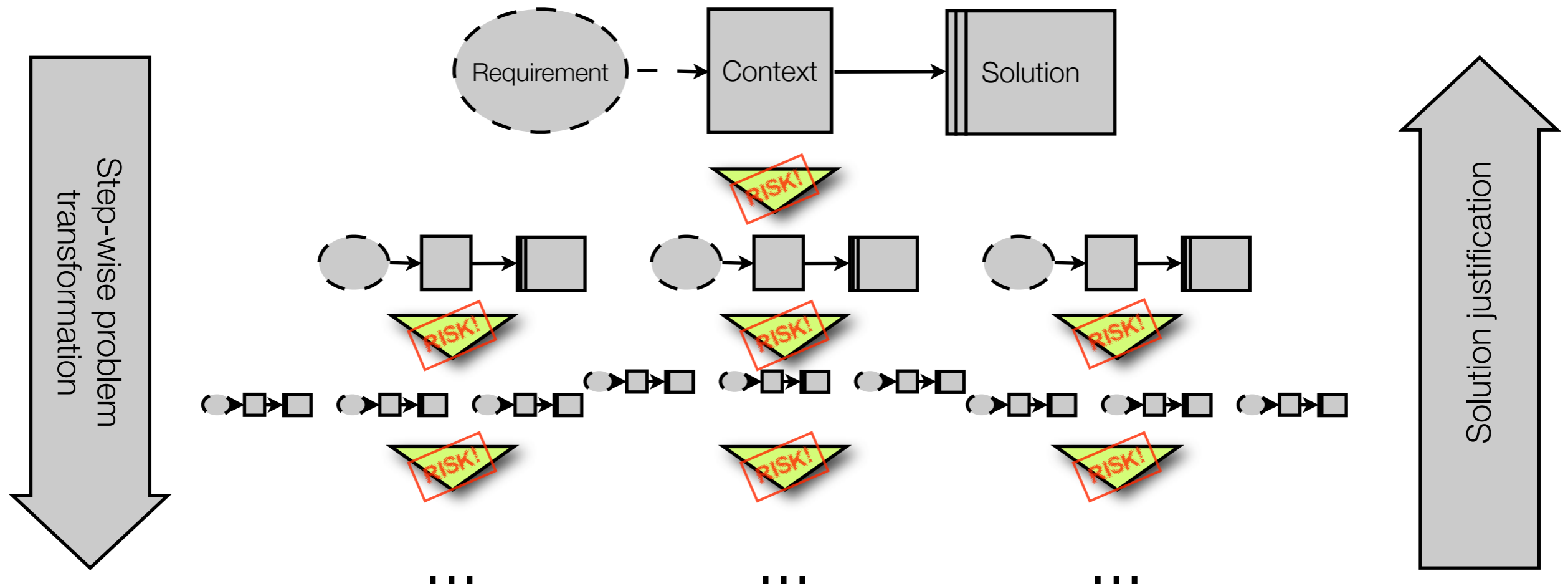


confirmation planned



# Adequacy

- ✿ Adequacy arguments justify the adequacy of a solution and establish a form of rich traceability between problems and solutions.





# Exploring the solution ...

---

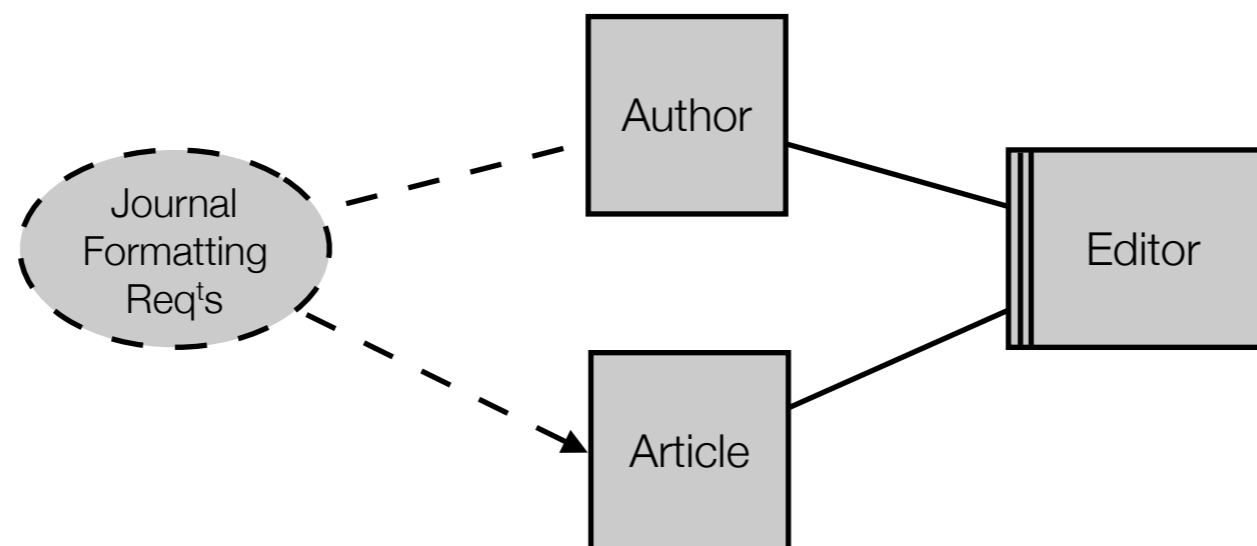
- ✱ Generally accepted that choice of solution structure can influence problem development
  - ✱ expertise often exists and is expressed in solution terms
  - ✱ reuse certainly happens in the solution space ...
    - ✱ ... and can make the most difficult problems trivial to solve
  - ✱ trade-offs affect problem development
  
- ✱ transform is *architectural expansion*





# A problem of style ...

- Journal editor needs an article editor that ensures the journal's format is used



<i>Author</i>	<i>presents ideas in an Article</i>
<i>Article</i>	<i>a document</i>
<i>Journal's formatting req's</i>	<i>the Article should be formatted in the Journal's style</i>
<b>Editor</b>	<b>to be found</b>



# A problem of style ... requirements

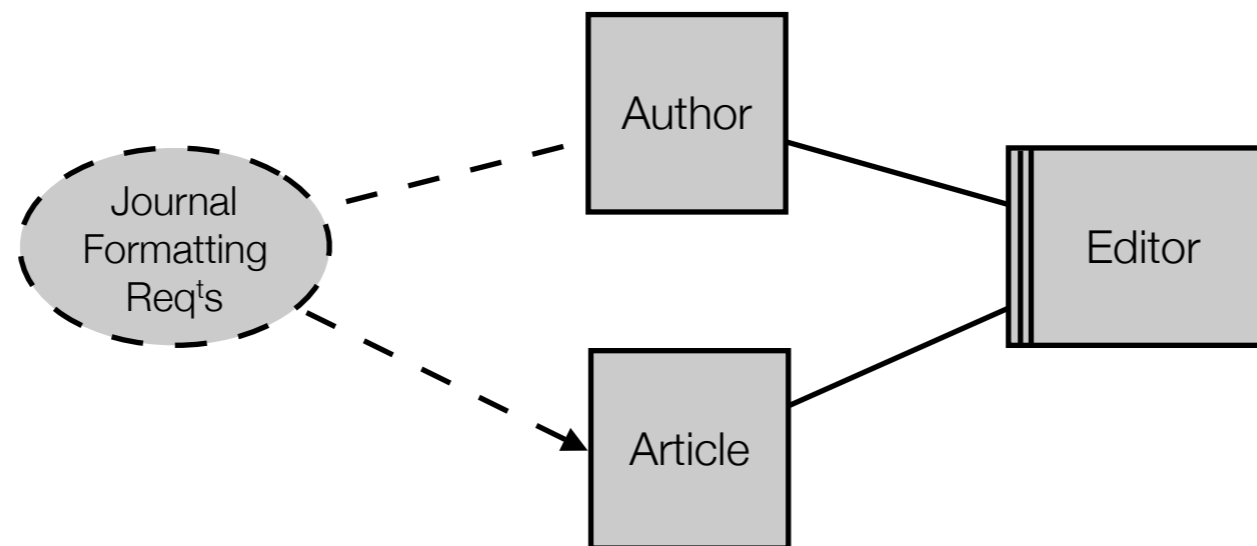
---

- cost-effective solution*
- needs excellent print quality*
- available for all platforms*
- good usability*
- no intellectual property rights of reuse*



# A problem of style ...

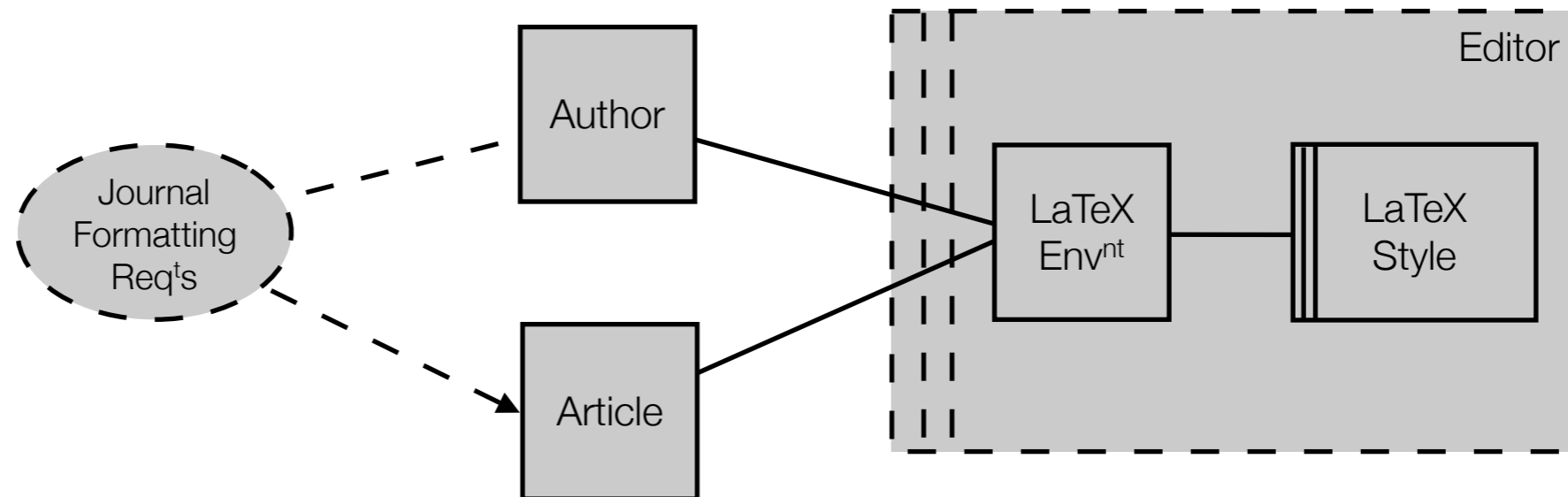
- ✱ Journal editor needs an article editor that ensures the journal's format is used
  - ✱ wants to consider a LaTeX environment as the *Editor* basis



<i>Author</i>	<i>presents ideas in an Article</i>
<i>Article</i>	<i>a document</i>
<i>Journal's formatting req's</i>	<i>the Article should be formatted in the Journal's style</i>
<b>Editor</b>	<b>to be found</b>

# A problem of style ...

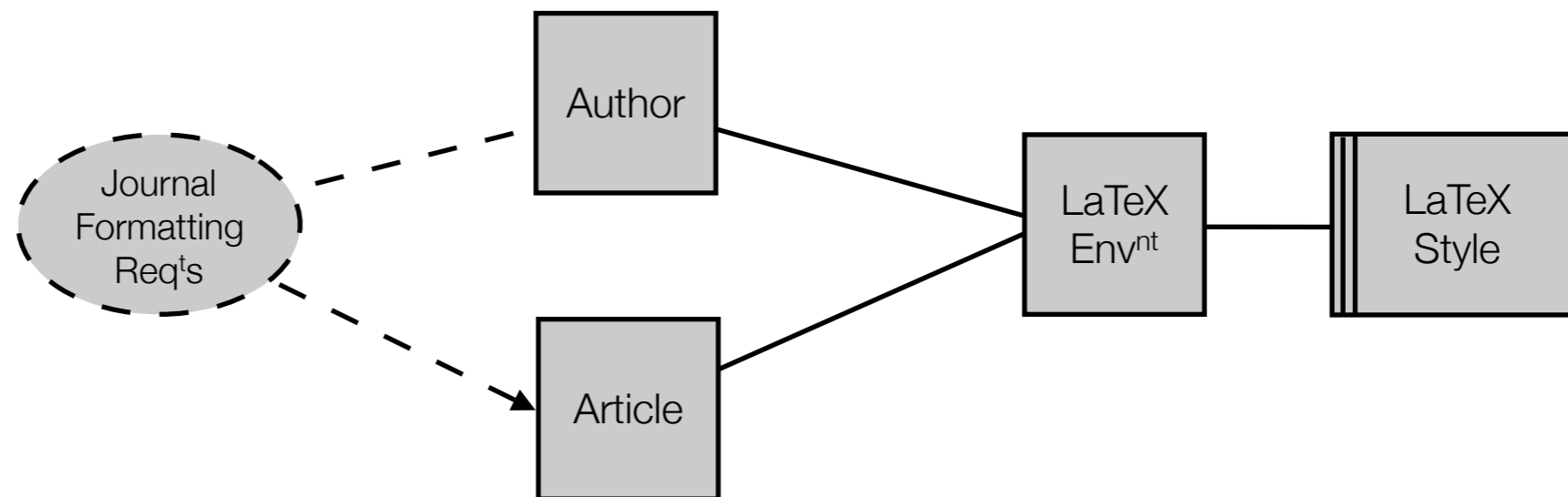
- Journal editor needs an article editor that ensures the journal's format is used
  - wants to consider a LaTeX environment as the *Editor* basis



<i>Author</i>	<i>presents ideas in an Article</i>
<i>Article</i>	<i>a document</i>
<i>Journal's formatting req's</i>	<i>the Article should be formatted in the Journal's style</i>
<i>Editor</i>	<i>LaTeX Environment (known) with <b>LaTeX Style (to be found)</b></i>

# A problem of style ...

- Journal editor needs an article editor that ensures the journal's format is used
  - wants to consider a LaTeX environment as the *Editor* basis



<i>Author</i>	<i>presents ideas in an Article</i>
<i>Article</i>	<i>a document</i>
<i>Journal's formatting req<sup>t</sup>s</i>	<i>the Article should be formatted in the Journal's style</i>
<i>LaTeX Env<sup>nt</sup></i>	<i>LaTeX Environment, LaTeX_Env</i>
<b>LaTeX Style</b>	<b>to be found</b>



# A problem of style ... trade-offs of using LaTeX

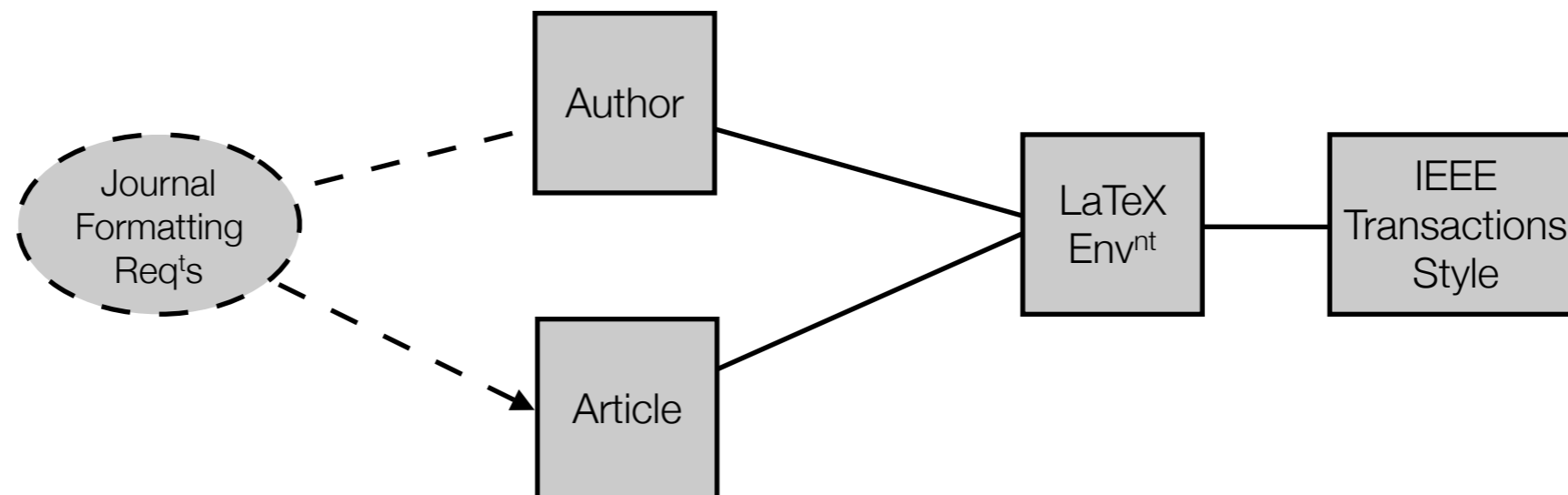
---

- cost-effective solution*
- adequate for the production of articles, excellent print quality*
  - *perhaps mathematical bias?*
- available for all platforms*
- issues of usability of the LaTeX 'programming' language*
- no intellectual property rights of reuse, LaTeX is public domain*
  - *is parametrisable using style files*



# A problem of style ...

- ✱ Journal editor needs an article editor that ensures the journal's format is used
  - ✱ wants to reuse a competing journal's style file



<i>Author</i>	<i>presents ideas in an Article</i>
<i>Article</i>	<i>a document</i>
<i>Journal's formatting req<sup>t</sup>s</i>	<i>the Article should be formatted in the Journal's style</i>
<i>LaTeX Env<sup>nt</sup></i>	<i>LaTeX Environment, LaTeX_Env</i>
<i>LaTeX Style</i>	<i>IEEE Transactions Style</i>



# Problem Frame Extensions







# Reaching the organisation

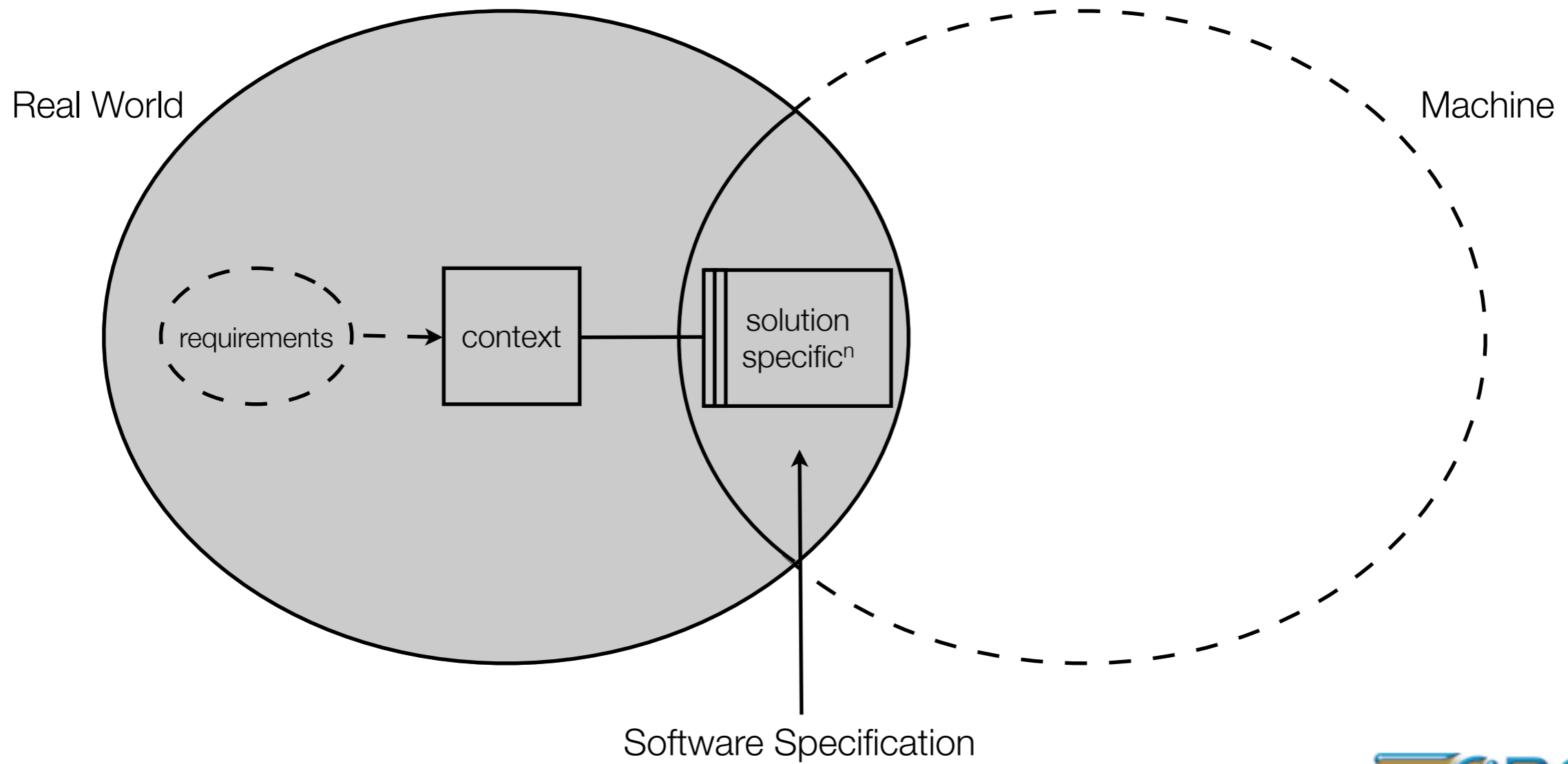
---

- ✻ Whilst retaining strong conceptual foundations, we can
  - ✻ move into the solution domain
  - ✻ include humans
  - ✻ include organisations

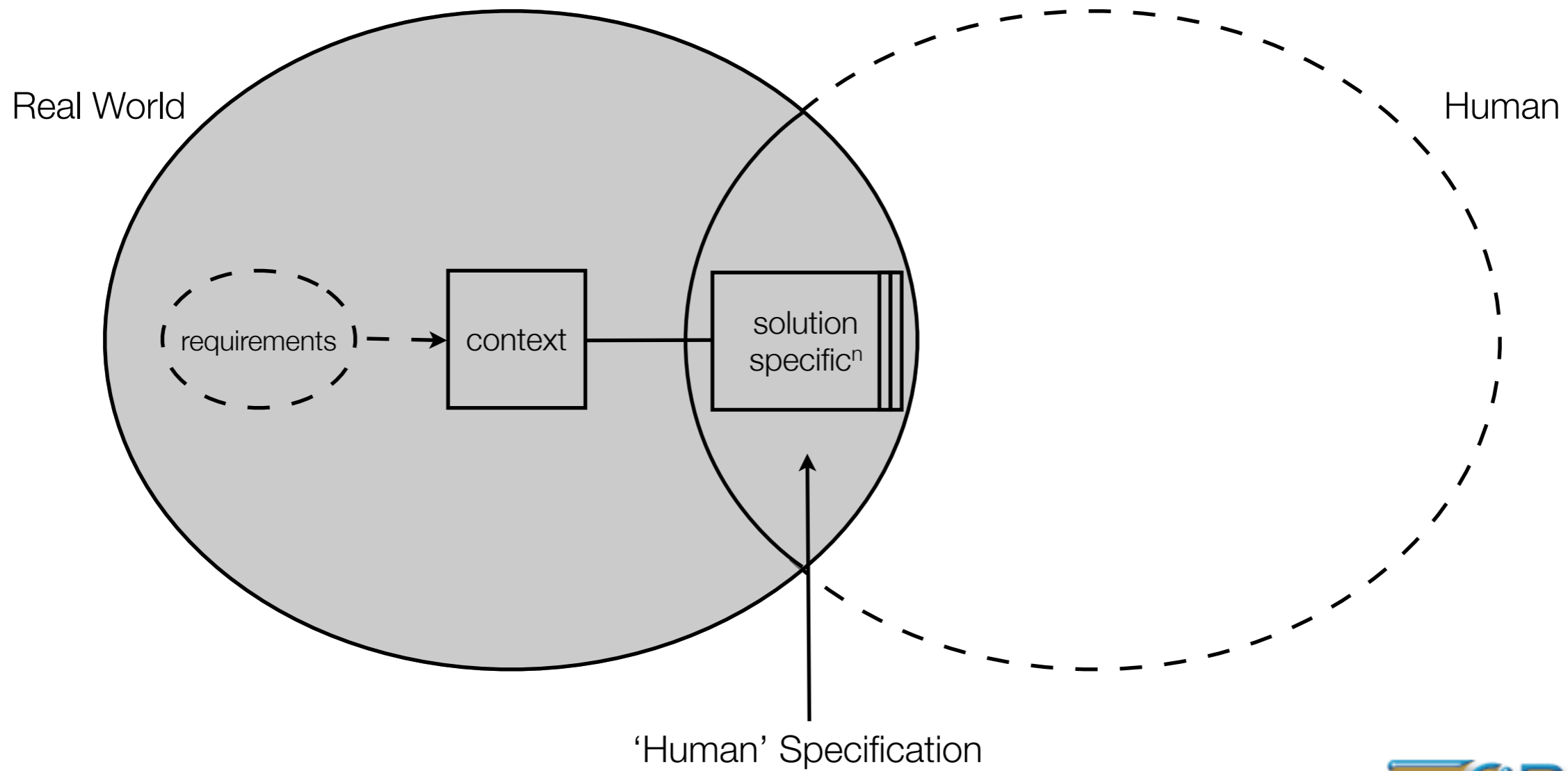




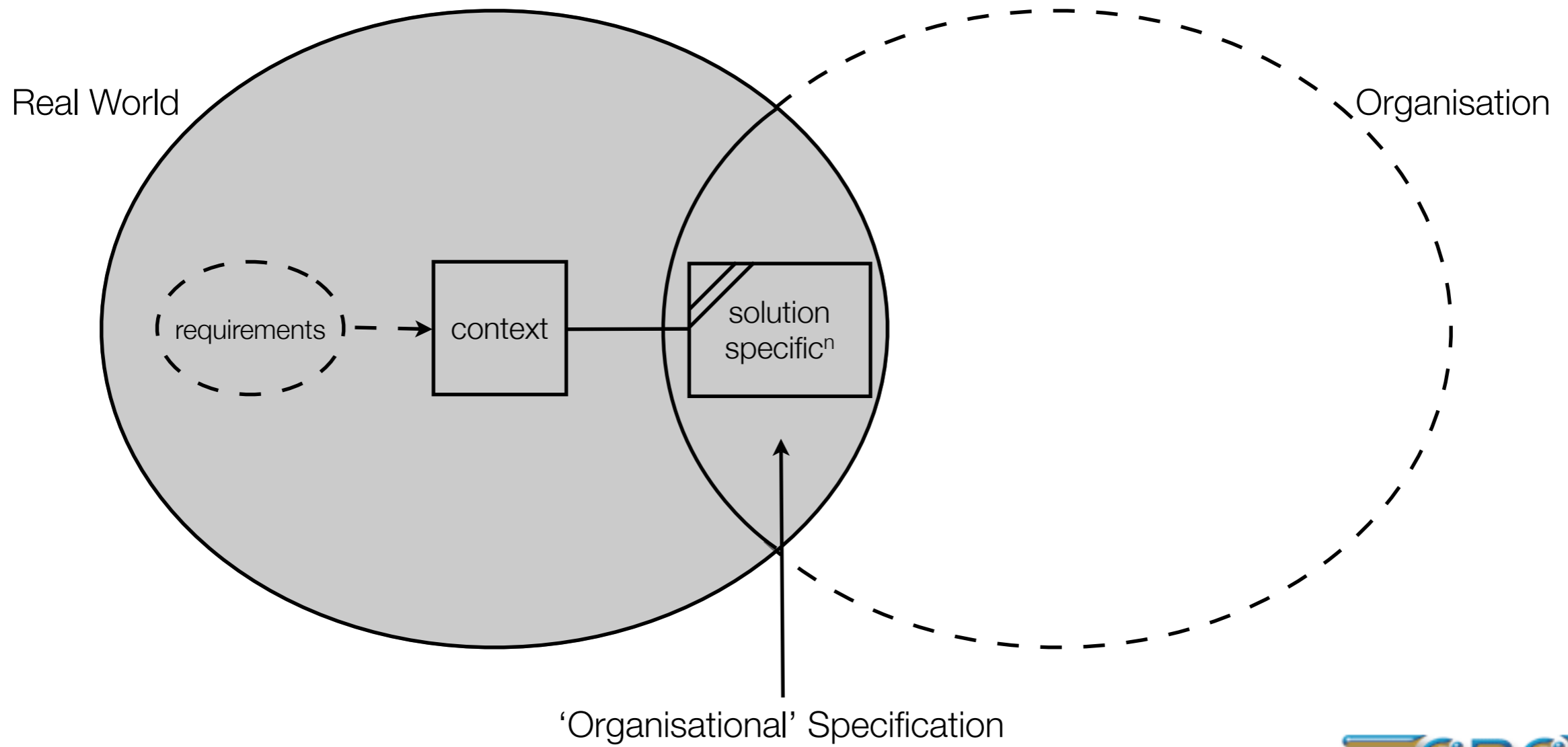
# Expanding applicability



# Expanding applicability



# Expanding applicability





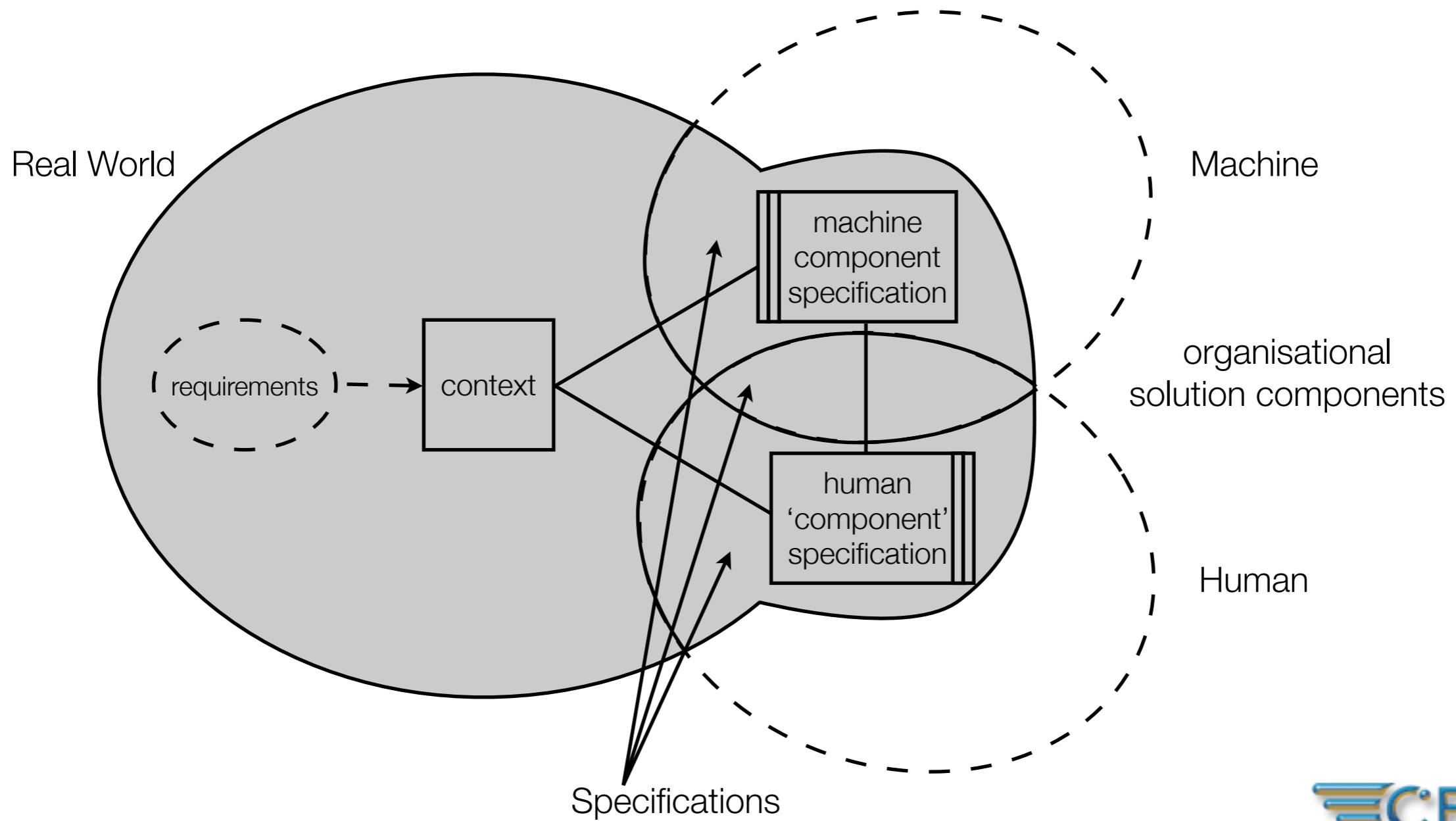
# And...

---

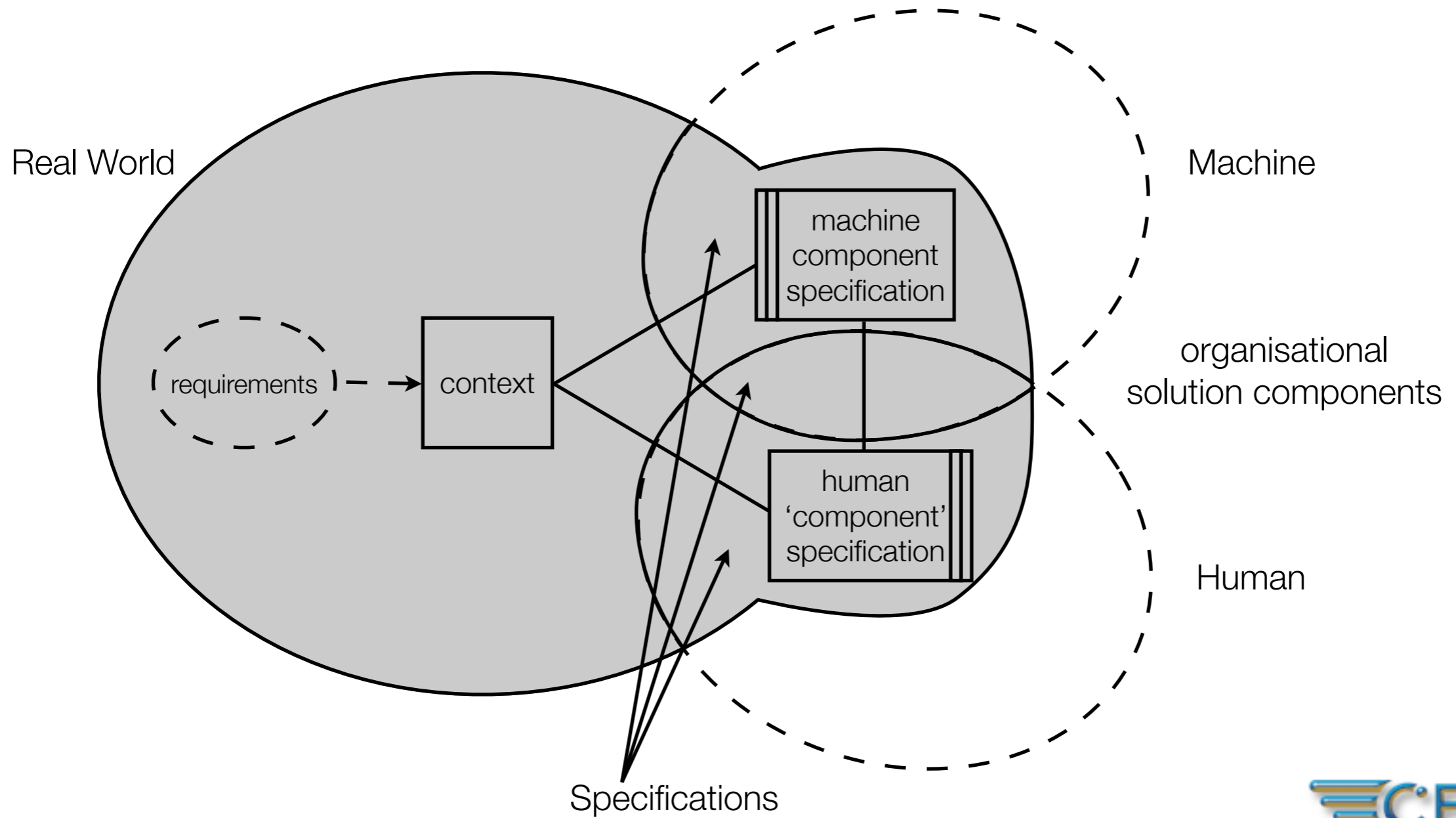
- ✻ Organisations consist of people of machines (working together in perfect harmony)
  - ✻ their 'architecture' is socio-technical



# Expanding applicability



# Expanding applicability



# Change Frames



- ✻ Work with John Brier
- ✻ Organisations face change from all directions
  - ✻ context: suppliers, customers, regulation, legislation, *etc*
  - ✻ *improving* technologies
  - ✻ automation
  - ✻ business process changes
- ✻ We want Change Frames to be able to replay the changes in response to environment changes
- ✻ We want to reflect this technology back as Software Change Frames

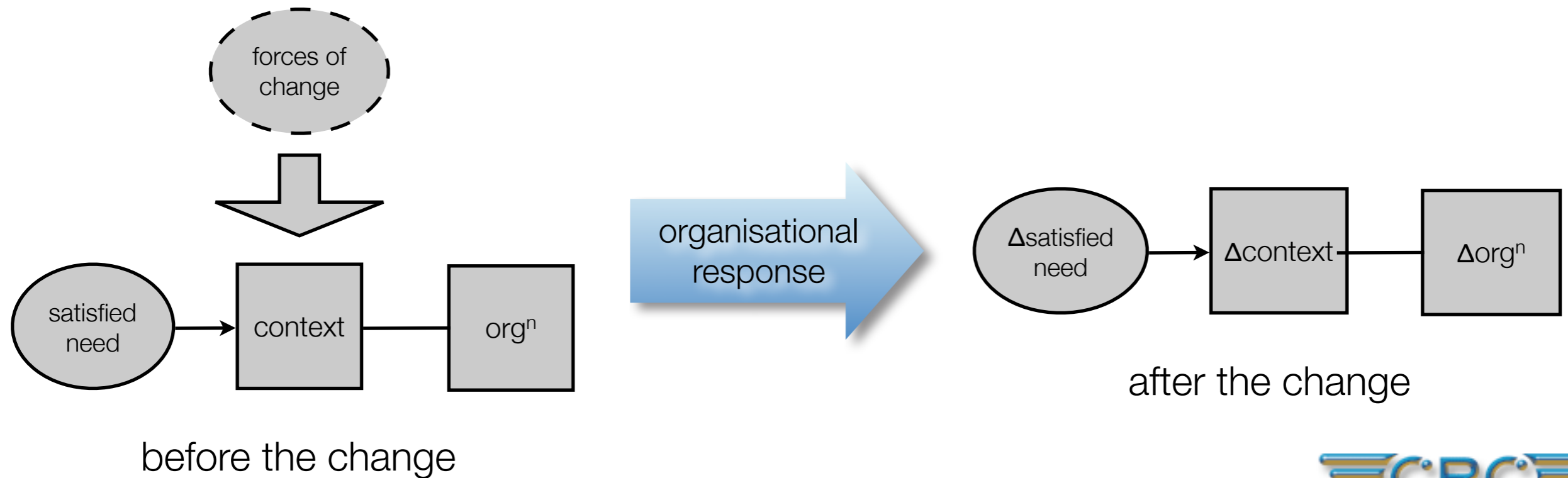




# Change Frames



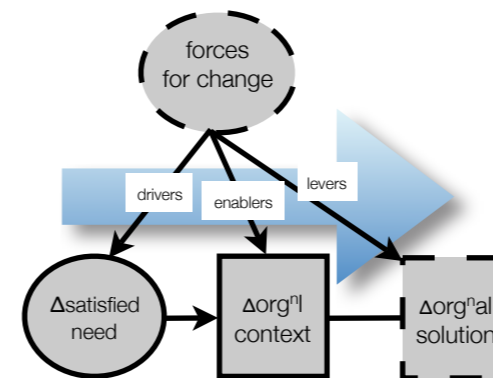
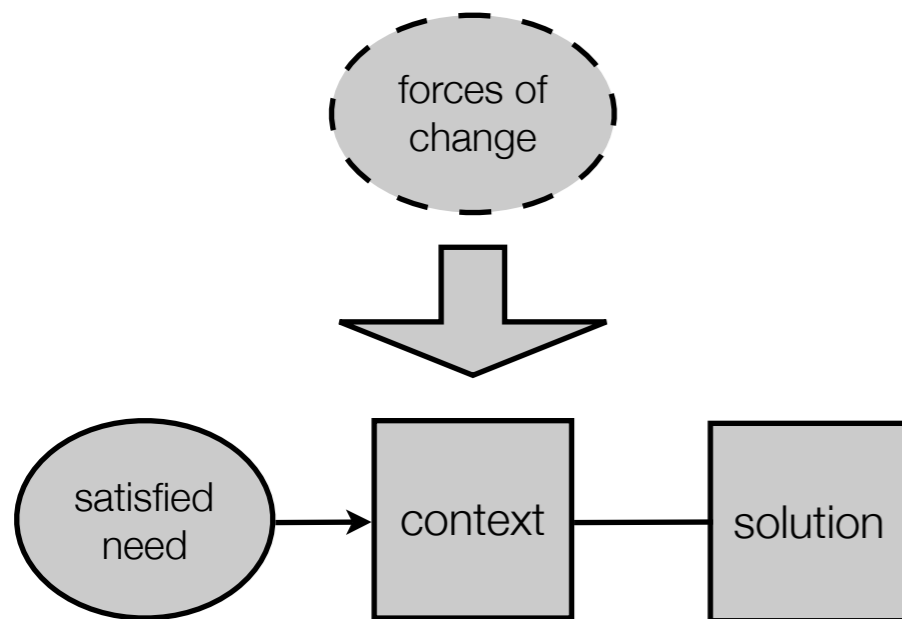
- ✱ Change Frames index change using:
  - ✱ forces for change: environmental drivers and enablers
    - ✱ changing context, changing satisfied need
  - ✱ response of the organisation: internal levers for change



# Change Frames



- Change Frames replay organisational response as a 'change script'



change pattern matches

1. begin by ...
2. then...
3. then...

'change script'

different organisation facing similar change situation



# Compliance requirements engineering



- ✻ Work with Shahbaz Ali
- ✻ Response to regulation, legislation
  - ✻ documentation of standards
  - ✻ risk assessment, control





# Conclusions

---

- ✱ Problem orientation is a step-wise problem solving framework
  - ✱ transformations correspond to sensible manipulations of problems
  - ✱ sense is defined step-wise (e.g., wrt risk, safety, *etc*)
  - ✱ rich traceability
- ✱ Extends to humans, socio-technical systems and organisations
- ✱ Allows change to be engineered (rather than products)





# Next steps ...

---

- ✿ Working on a textual notation, suitable as a tool basis (Hall, Rapanotti & Jackson 2005)
- ✿ Problem oriented Eclipse plug-in (Hall & Rapanotti)
- ✿ Socio-technical systems (Hall & Rapanotti, 2005), Organisational systems
- ✿ Change in Organisations (Brier *et al.*, 2005,6)
- ✿ Safety-critical systems (Mannering *et al.*)
- ✿ Mission-critical systems/Compliance requirements engineering (Ali, 2006)
- ✿ Conceptual foundations (Hall, Rapanotti & Jackson, 2003,4,5); transformations (Li *et al.*, 2005,6)
- ✿ Design tactics (Hall & Rapanotti)

