A No-Code Approach to Decision Support and Knowledge Capturing

simon.vandevelde@kuleuven.be

https://simonvandevel.de

https://DTAI.cs.kuleuven.be

Declarative Languages and Artificial Intelligence

15 Faculty
2 (IOF) Research Manager
± 10 Postdoctoral Researchers
± 60 PhD Researchers
> 80 Alumni

DEPARTEMENT COMPUTER WETENSCHAPPEN

Faculty of Engineering Science

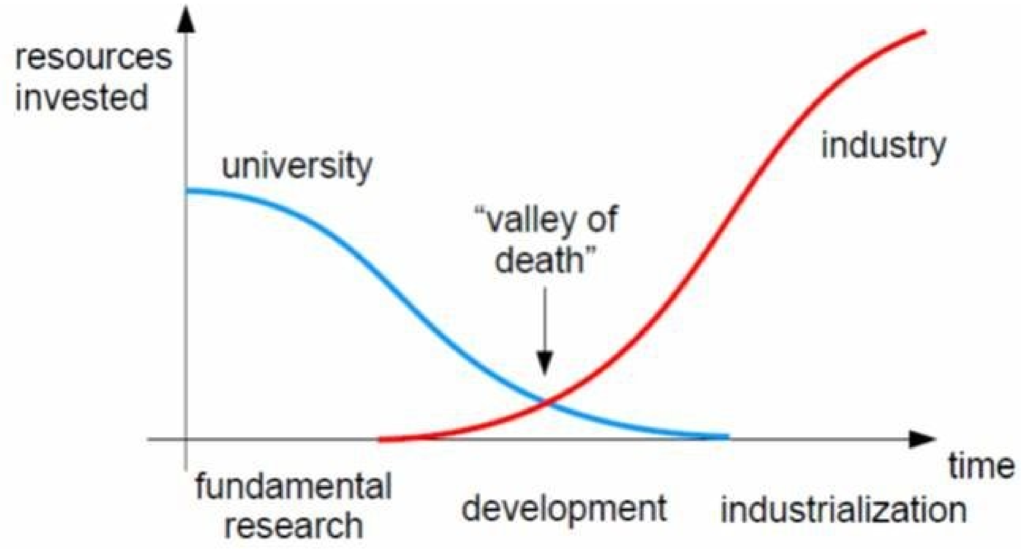
KU LEUVEN
FACULTY OF ENGINEERING TECHNOLOGY

Basic Research	Applications
Probabilistic Programming	Sports Analytics / Health
Predictive Learning and Clustering	Anomaly Detection
Automated Data Science	Robotics
Neuro-Symbolic	Engineering and Sensors
Privacy, Non-discrimination and Ethics	Creativity
Constraints	Texts and Web
Functional Programming	Games
Knowledge Base Systems	

http://dtai.cs.kuleuven.be/stories

Research topic

- Applications of (knowledge-based) AI technology
- Decision support
  - Ecosystem of tools
  - Concrete use cases with industrial partners



Contents

- Introduction: Knowledge-based AI methods in context
- Applications: Decision support
- Hands-on DMN
- Limitations of DMN
    - Interactive consultant
    - cDMN
- A deeper look
    - At inference tasks
    - At cDMN
- A realistic example
- Conclusions: use cases

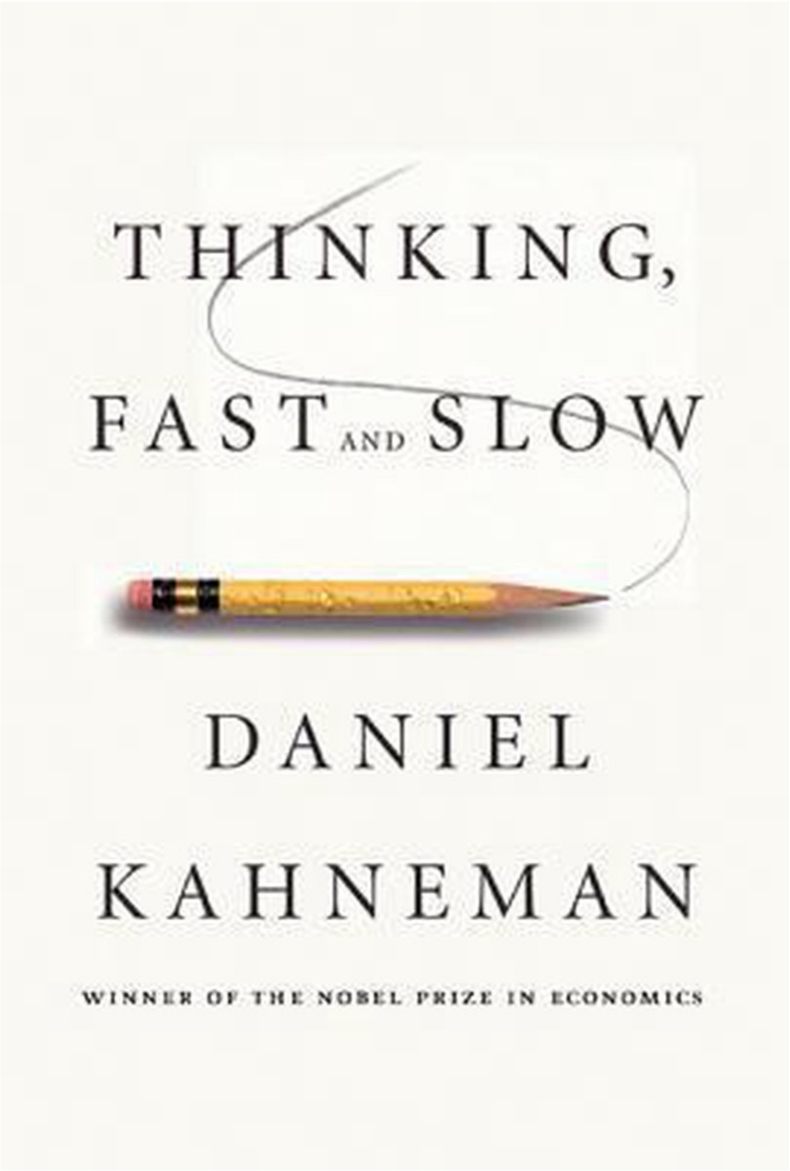
Introduction:

Knowledge-based AI methods

Artificial Intelligence

- Build systems that exhibit intelligent behaviour

Different kinds: small experiment



Don’t read this text

2875 + 1223 = ?

System 1

“Thinking fast”

- Automatic, effortless
- Implicit patterns
- Intuition from (lots of) experience
- Black box

System 2

“Thinking slow”

- Conscious reasoning
- Explicit information
- Rational, logical
- Explainable

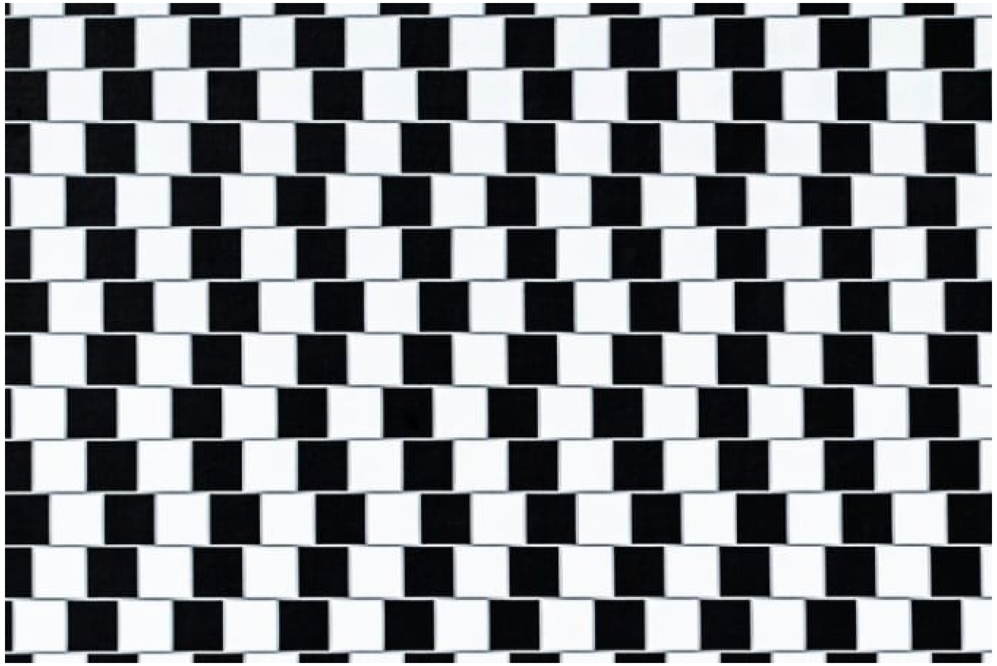
System 1

- Powerful:
  - Not exact matching
  - Learn implicit patterns

- But error-prone

- Requires a lot of data

Yuo cna porbalby raed tihs esaliy desptie teh msispeillgns.



System 2

👍

- Reliable
- Easily transferable

👎

- Requires explicit knowledge
- Black and white

Photo by Volodymyr Hryshchenko on Unsplash

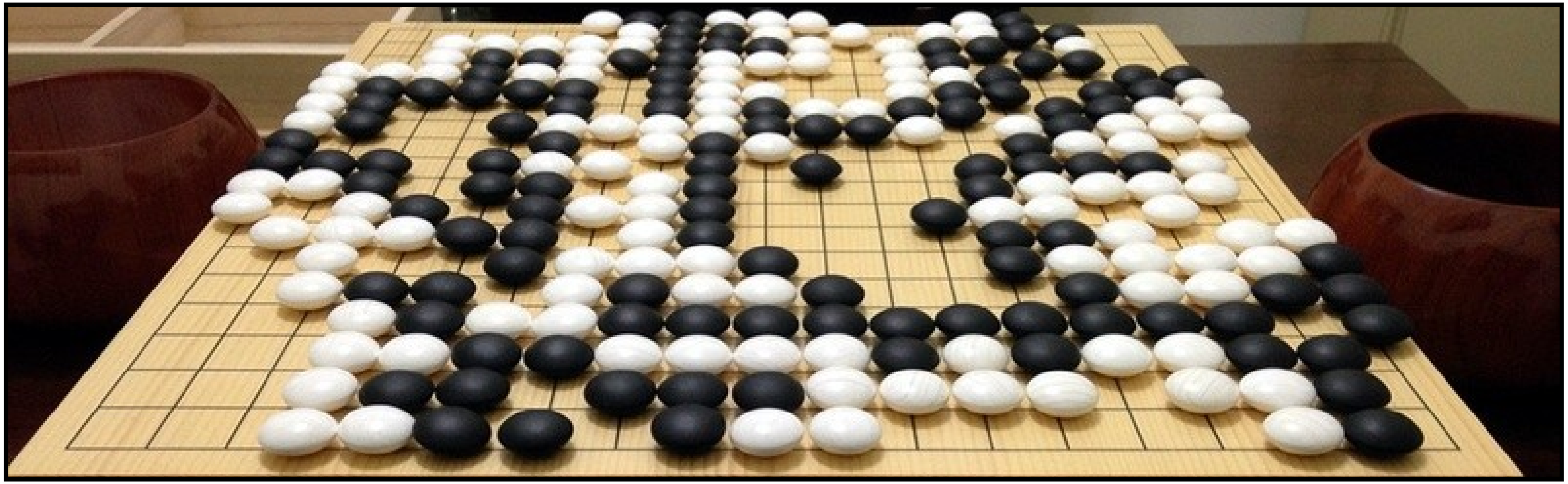
System 1

- Pick up statistical patterns
- Not always accurate

+

System 2

- Use explicit knowledge
- More reliable

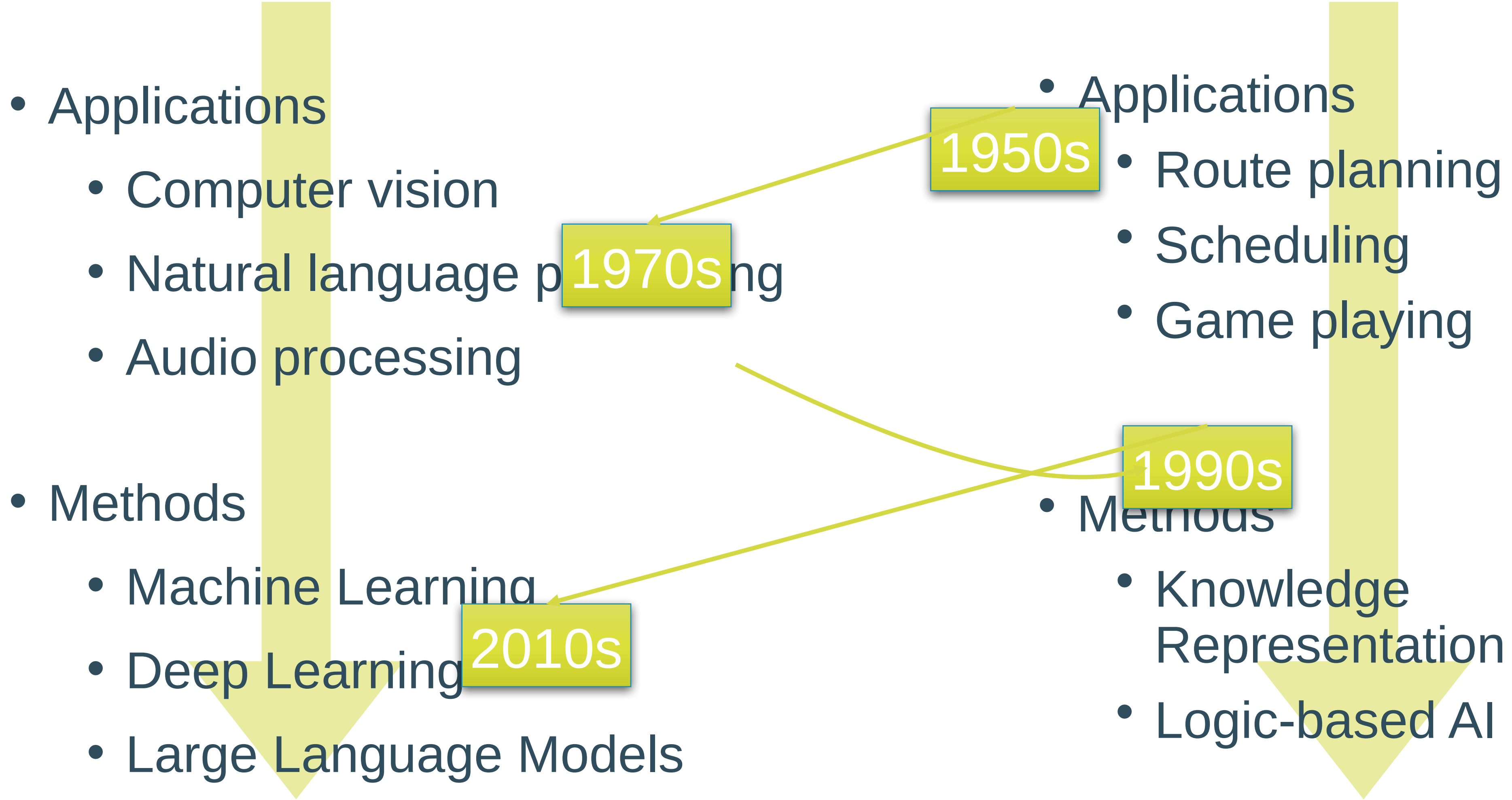


Artificial System 1

- Applications
  - Computer vision
  - Natural language processing
  - Audio processing
- Methods
  - Machine Learning
  - Deep Learning
  - Large Language Models

Artificial System 2

- Applications
  - Route planning
  - Scheduling
  - Game playing
- Methods
  - Knowledge Representation
  - Logic-based AI



Artificial System 1

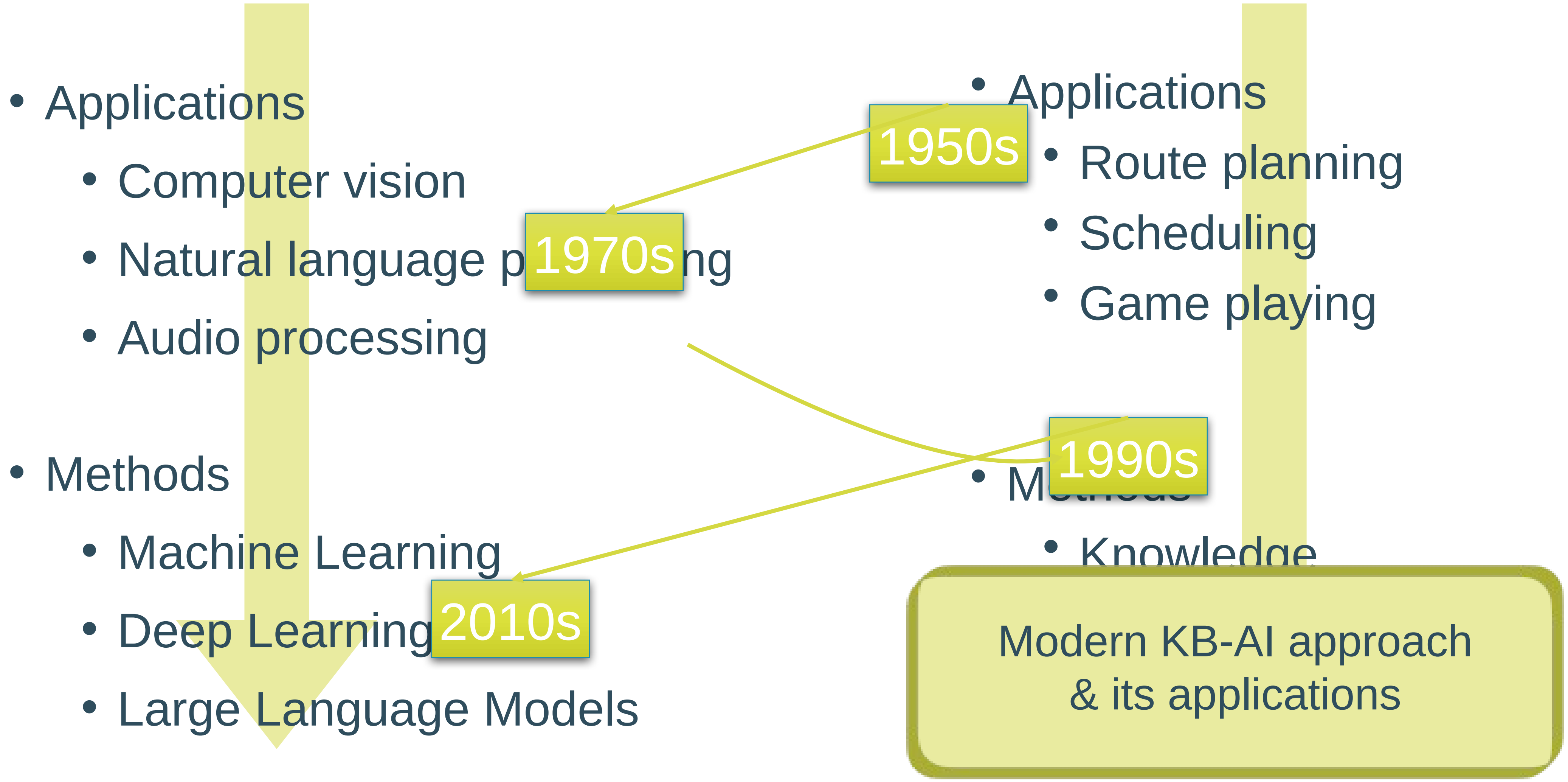
- Applications
  - Computer vision
  - Natural language processing
  - Audio processing
- Methods
  - Machine Learning
  - Deep Learning
  - Large Language Models

Artificial System 2

- Applications
  - Route planning
  - Scheduling
  - Game playing
- Methods
  - Knowledge Representation
  - Logic-based AI

Artificial System 1

Artificial System 2



- Applications
  - Computer vision
  - Natural language p 1970s ng
  - Audio processing
- Methods
  - Machine Learning
  - Deep Learning 2010s
  - Large Language Models

1950s

- Applications
  - Route planning
  - Scheduling
  - Game playing

1990s

- M
  - Knowledge

Modern KB-AI approach
& its applications

Applications:
Decision support

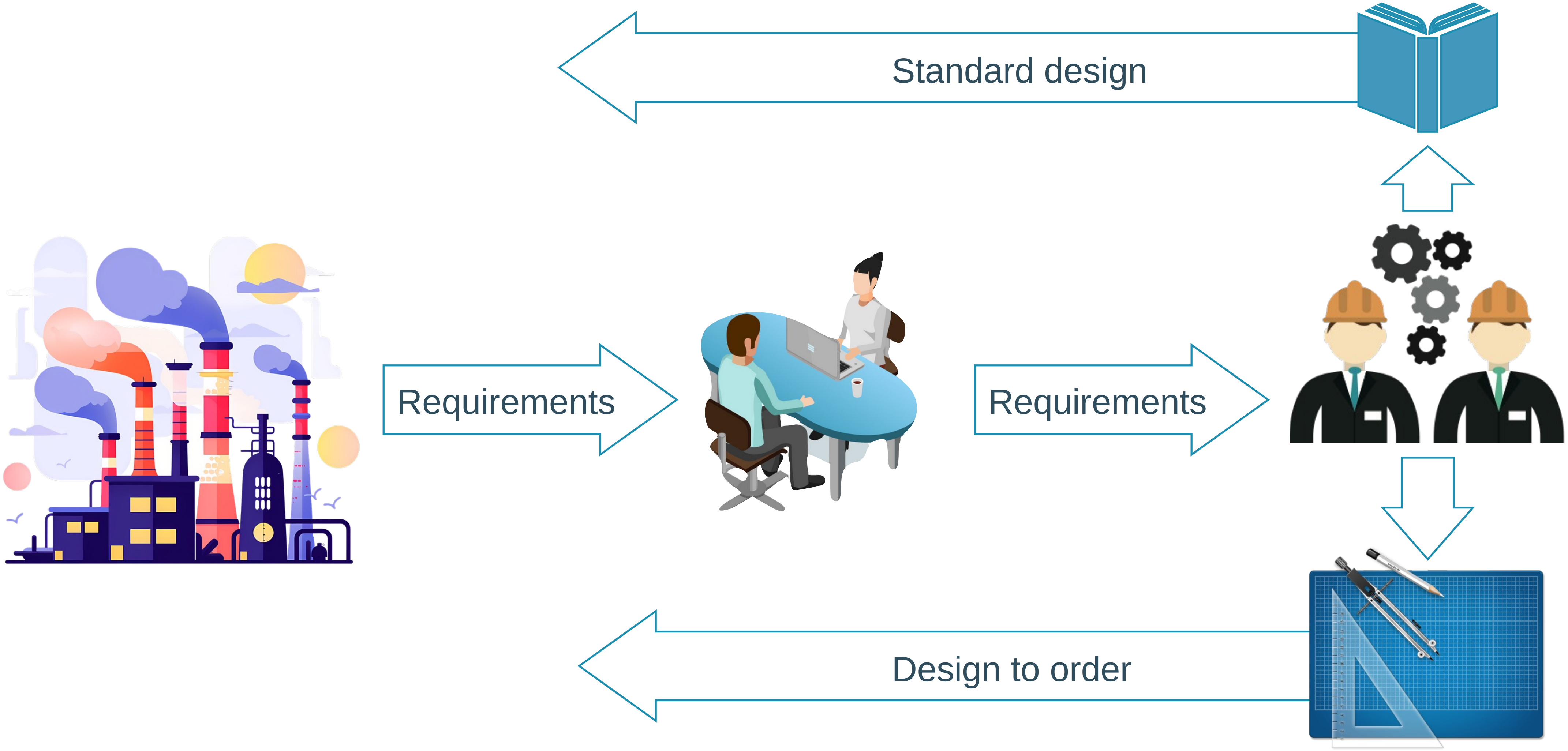
Concrete use cases

- Product selection *Joining & Materials lab (Flanders Make)*
- **Product design *Manufacturing multinational***
- Support for notary *Software company*
- Selection of financial products *Intelli-Select*
- …

Seal design for challenging applications

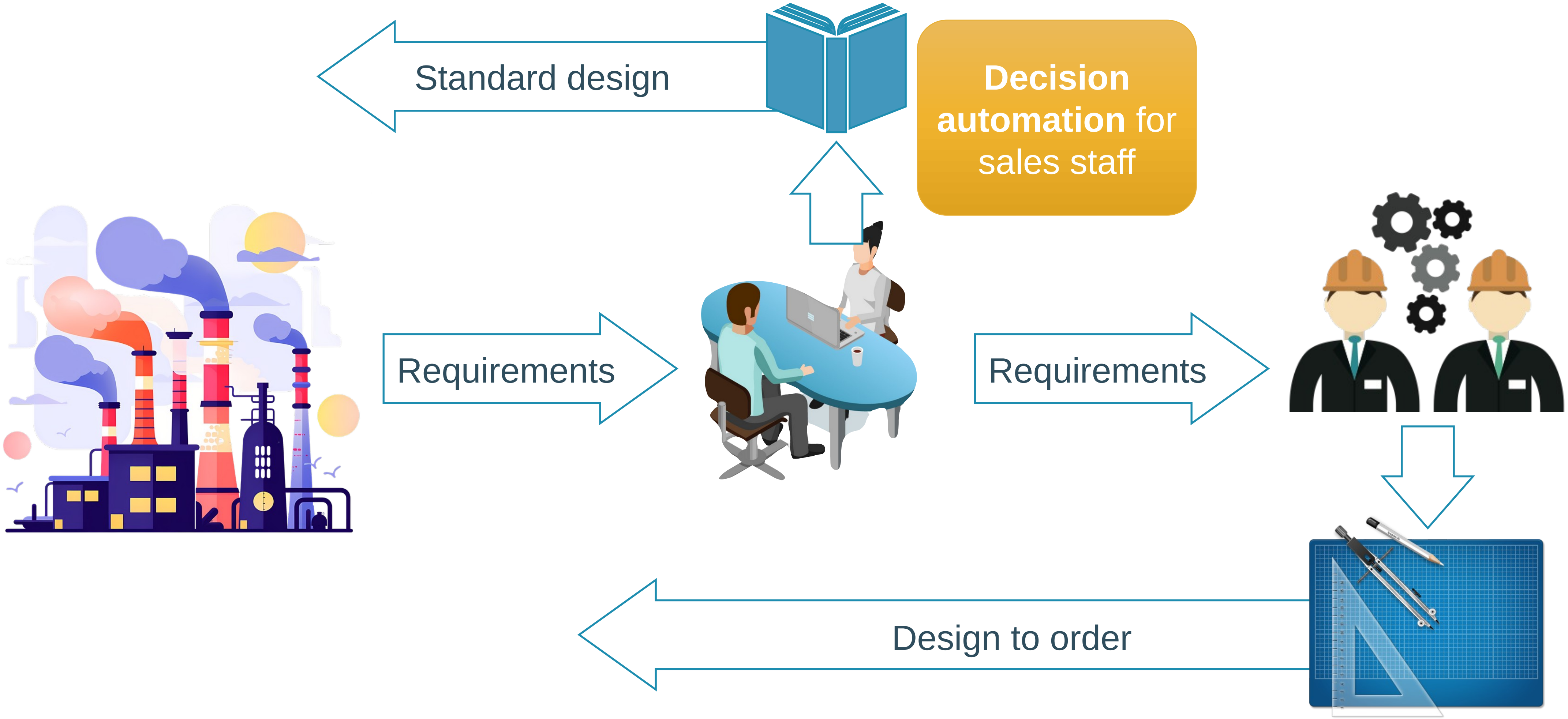
Reliability is key

Process (before)



jemima @ https://www.cleanpng.com

Goals



Standard design

Decision automation for sales staff

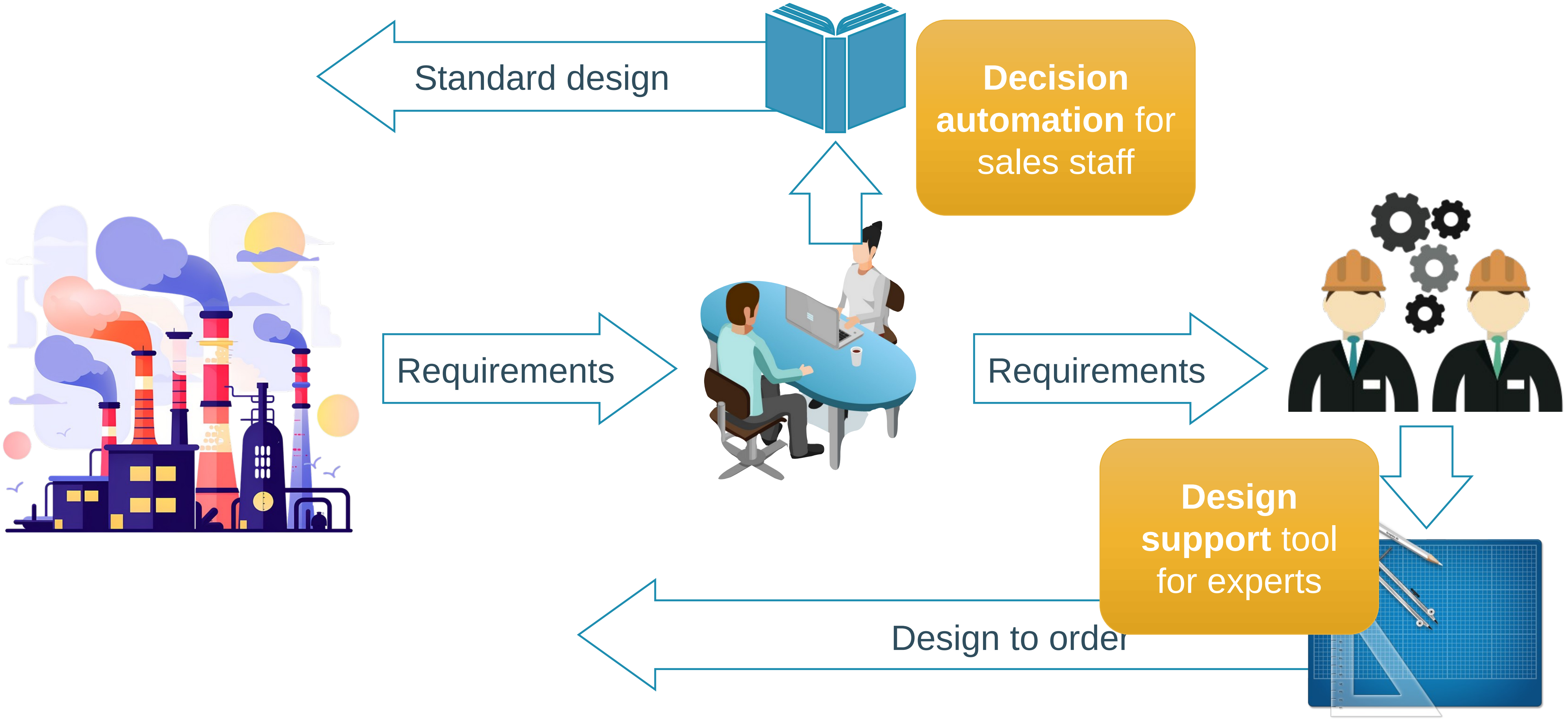
Requirements

Requirements

Design to order

jemima @ https://www.cleanpng.com

Goals



Standard design

Decision automation for sales staff

Requirements

Requirements

Design support tool for experts

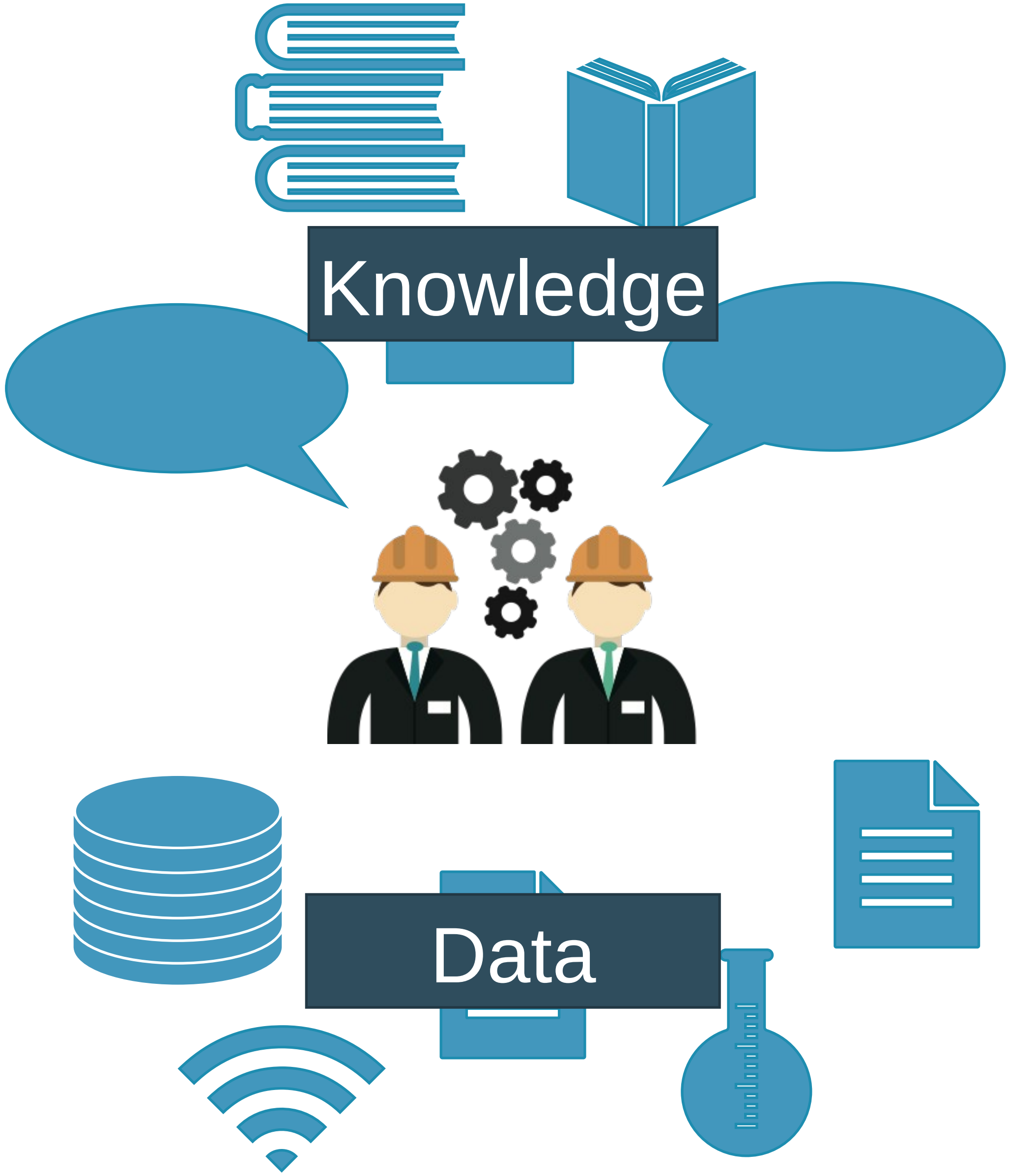
Design to order

jemima @ https://www.cleanpng.com

Resources

- Knowledge of engineers
  - Very valuable
  - Useful
  - Safeguarding

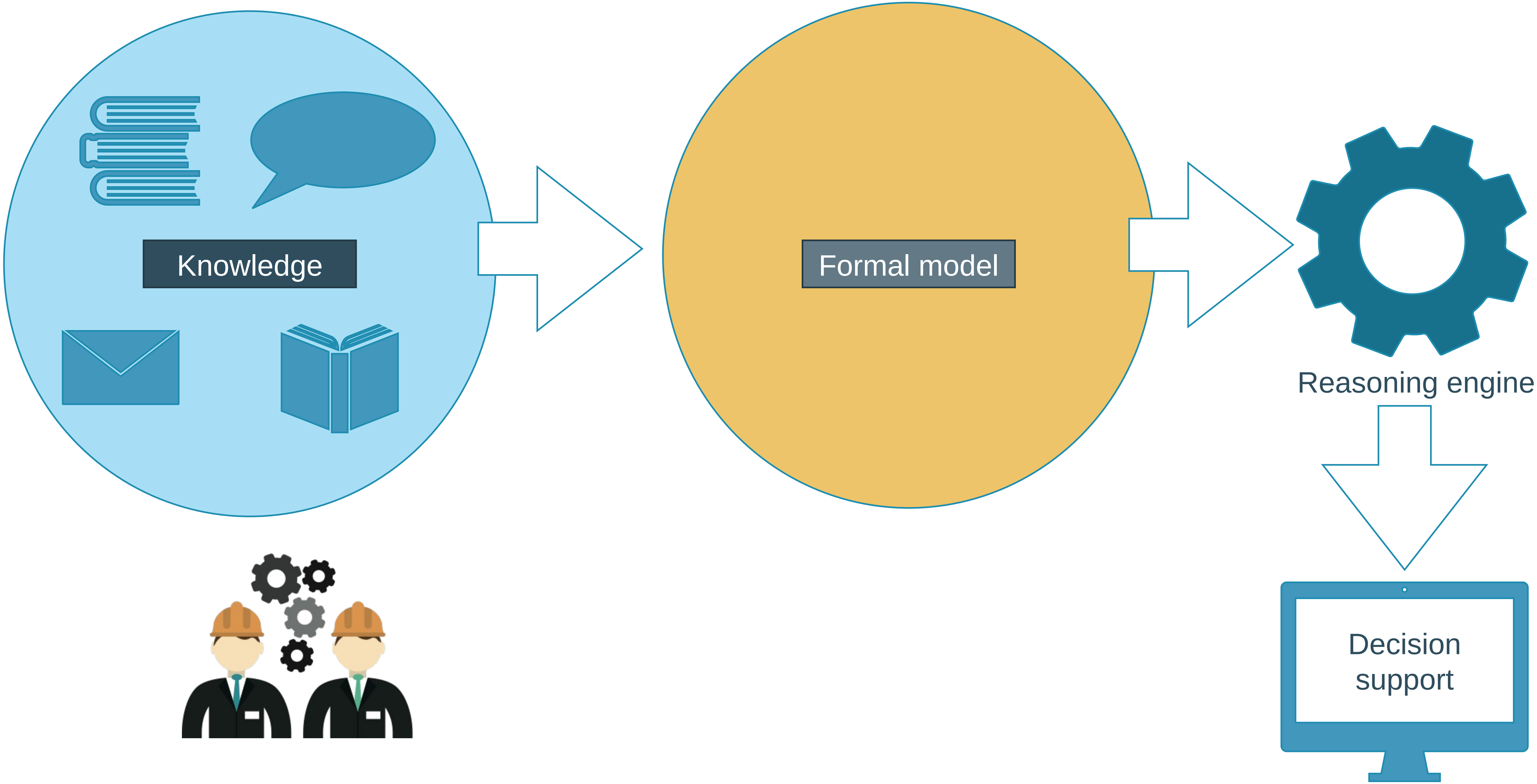
- Database of earlier designs
  - Unlabeled
  - Limited usefulness



Resources

- Knowledge of engineers
  - Very valuable
  - Useful
  - Safeguarding

- Database of earlier designs
  - Unlabeled
  - Limited usefulness

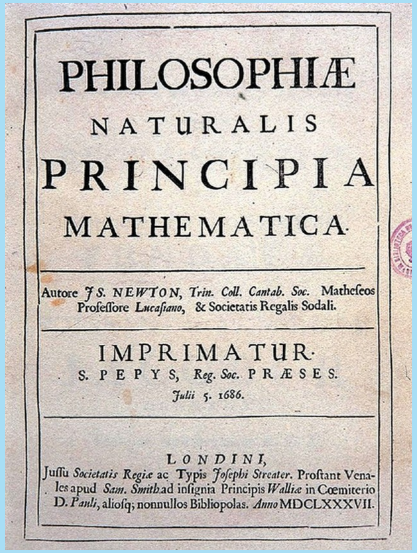


Knowledge

Formal model

Reasoning engine

Decision support

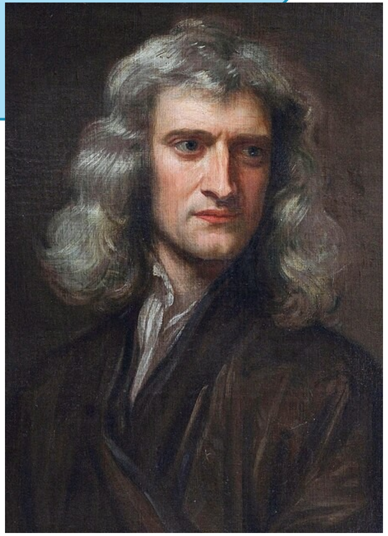


PHILOSOPHIÆ
NATURALIS
PRINCIPIA
MATHEMATICA.

Autore J S. NEWTON, *Trin. Coll. Cantab. Soc.* Matheſeos
Profeſſore *Lucaſiano*, & Societatis Regalis Sodali.

IMPRIMATUR.
S. PEPYS, *Reg. Soc.* PRÆSES.
Julii 5. 1686.

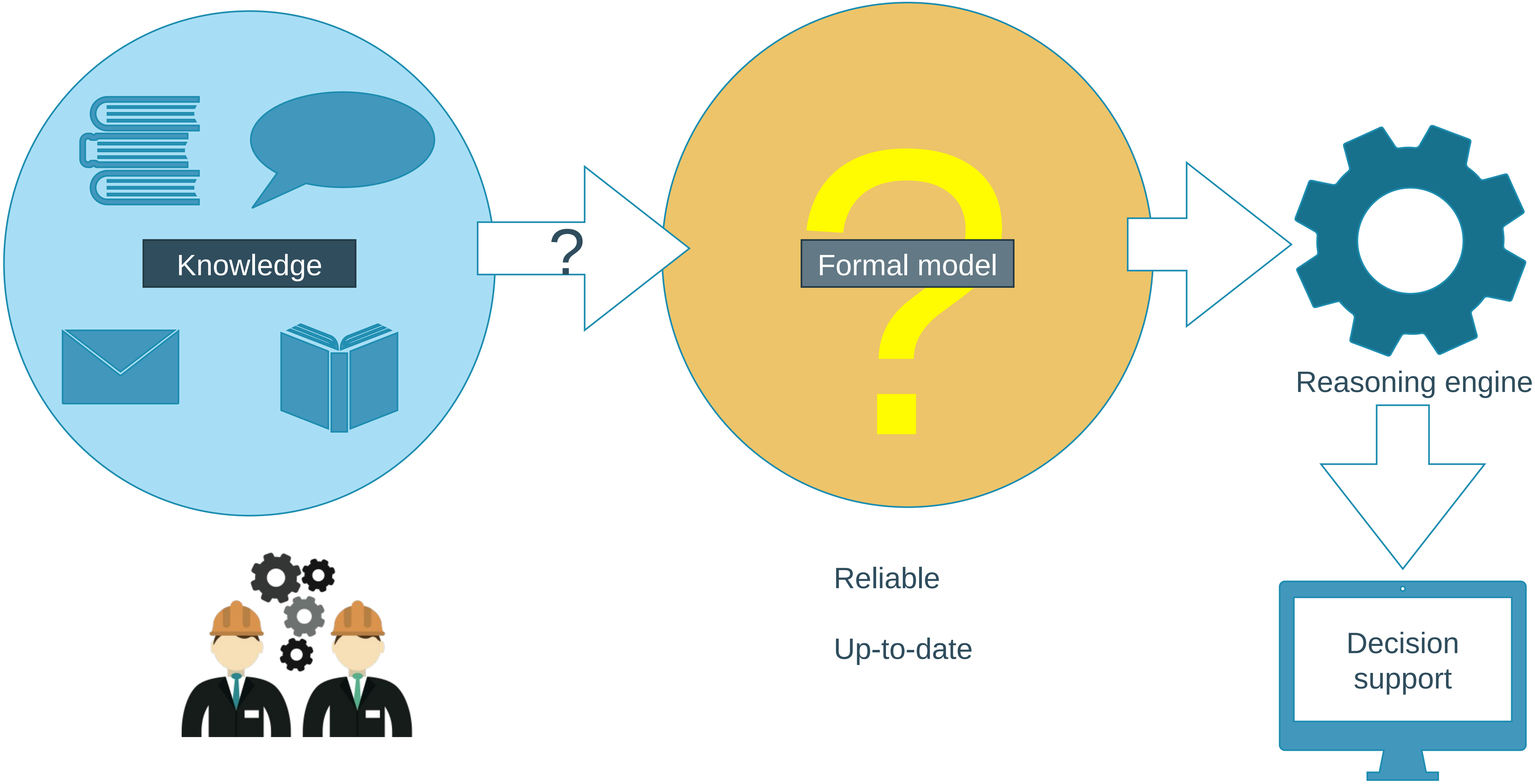
LONDINI,
Juſſu *Societatis Regiæ* ac Typis *Joſephi Streater*. Proſtant Venales apud *Sam. Smith* ad inſignia Principis *Walliæ* in Cœmiterio
D. *Pauli*, alioſq; nonnullos Bibliopolas. *Anno* MDCCLXXXVII.



F = m*a

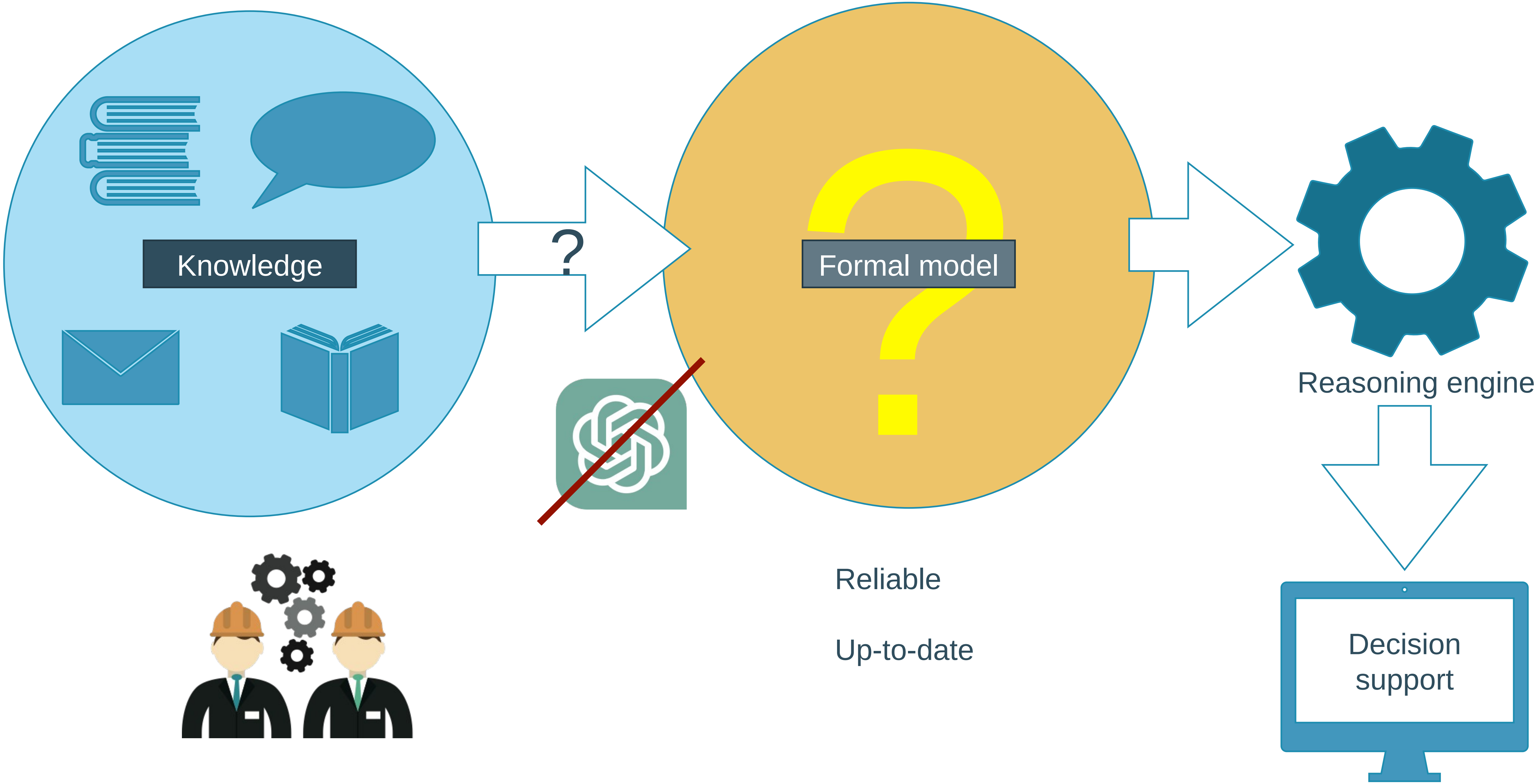
Mathematics engine

Decision support



Reliable

Up-to-date



Knowledge

?

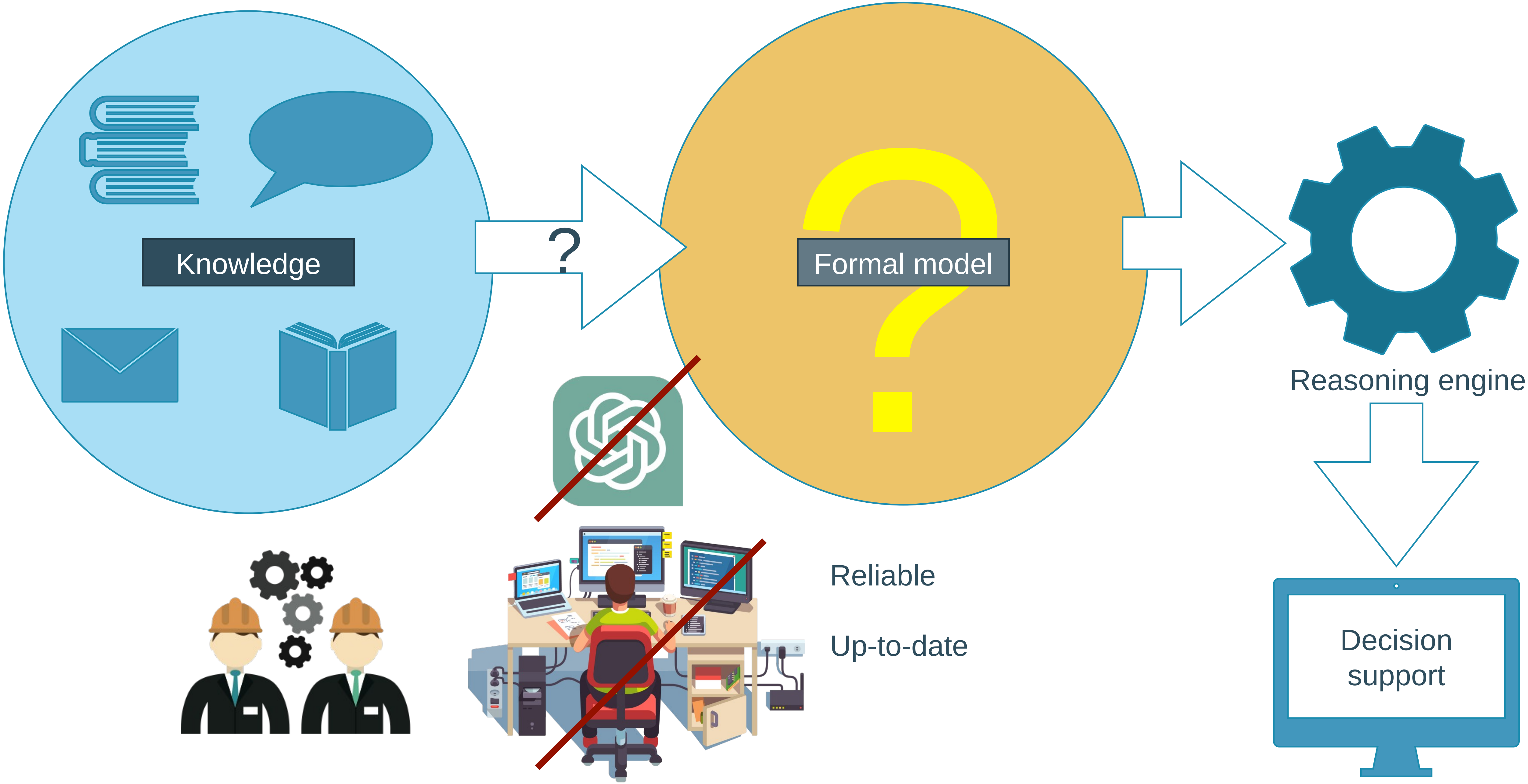
Formal model

Reasoning engine

Decision support

Reliable

Up-to-date



Knowledge

?

Formal model

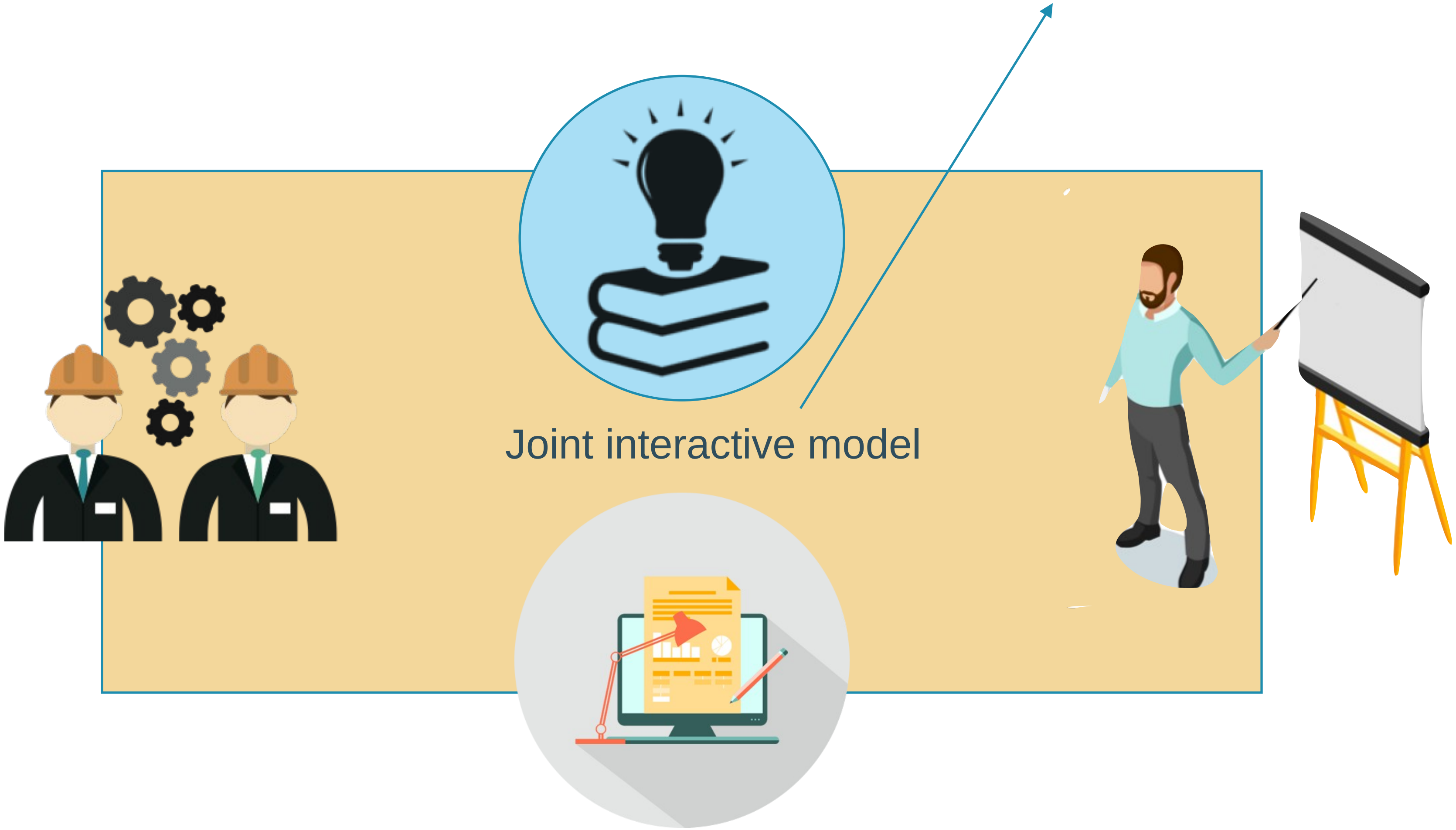
Reasoning engine

Decision support

Reliable

Up-to-date

What should this look like?



Joint interactive model

DMN

- Industry standard (OMG group)
- Readable / writable for domain experts
- Supported by different tools

MONITEUR BELGE | BELGISCH STAATSBLAD

Publication conforme aux articles 472 à 478 de la loi-programme du 24 décembre 2002, modifiés par les articles 4 à 8 de la loi portant des dispositions diverses du 20 juillet 2005 et les articles 117 et 118 de la loi du 5 mai 2019.

Le *Moniteur belge* peut être consulté à l'adresse :
www.moniteur.be

Direction du Moniteur belge, chaussée d'Anvers 53, 1000 Bruxelles - Directeur : Wilfried Verrezen

Numéro tél. gratuit : 0800-98 809

191e ANNEE

Publicatie overeenkomstig artikelen 472 tot 478 van de programmawet van 24 december 2002, gewijzigd door de artikelen 4 tot en met 8 van de wet houdende diverse bepalingen van 20 juli 2005 en artikelen 117 en 118 van de wet van 5 mei 2019.

Dit *Belgisch Staatsblad* kan geconsulteerd worden op :
www.staatsblad.be

Bestuur van het Belgisch Staatsblad, Antwerpsesteenweg 53, 1000 Brussel - Directeur : Wilfried Verrezen

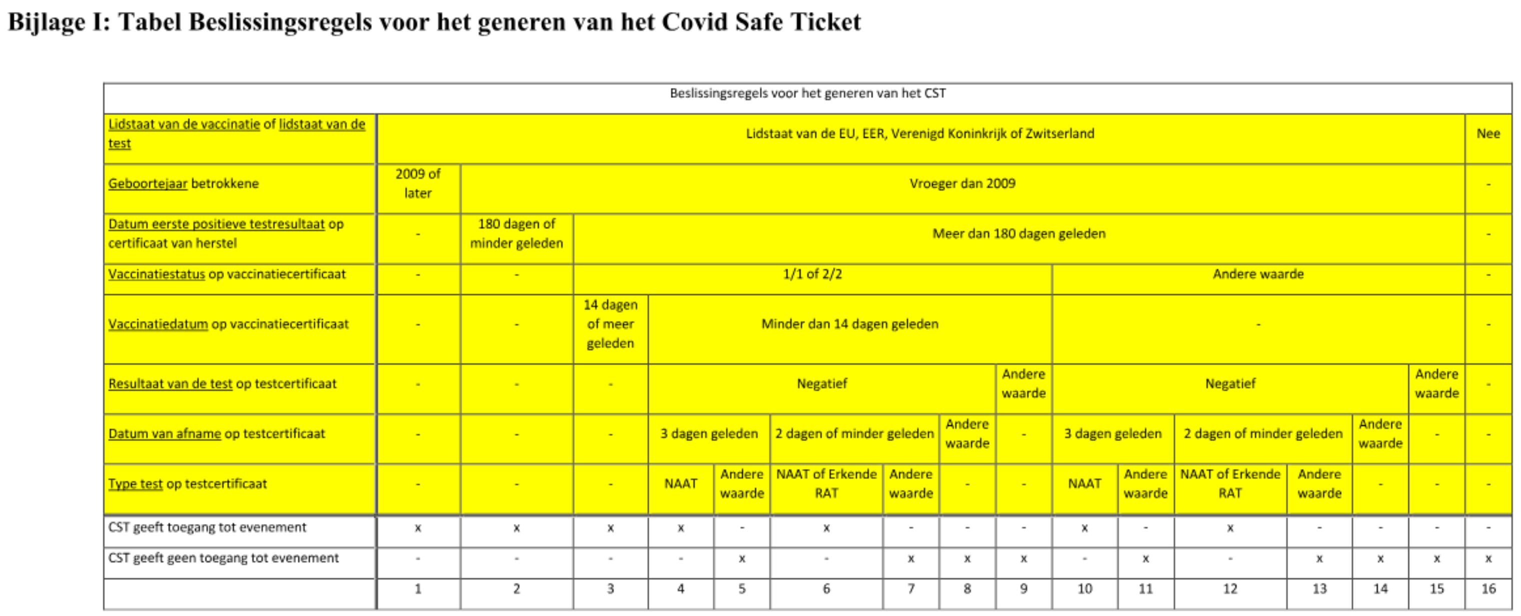
Gratis tel. nummer : 0800-98 809

191e JAARGANG

N. 204

VENDREDI 23 JUILLET 2021
TROISIEME EDITION

VRIJDAG 23 JULI 2021
DERDE EDITIE



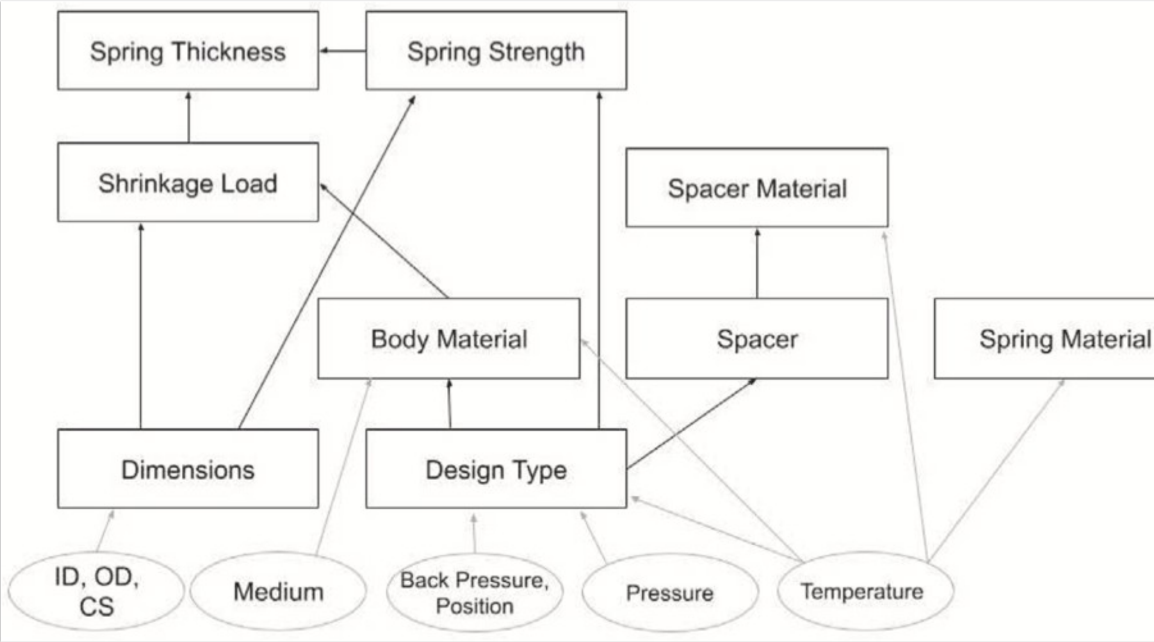
Bijlage I: Tabel Beslissingsregels voor het generen van het Covid Safe Ticket

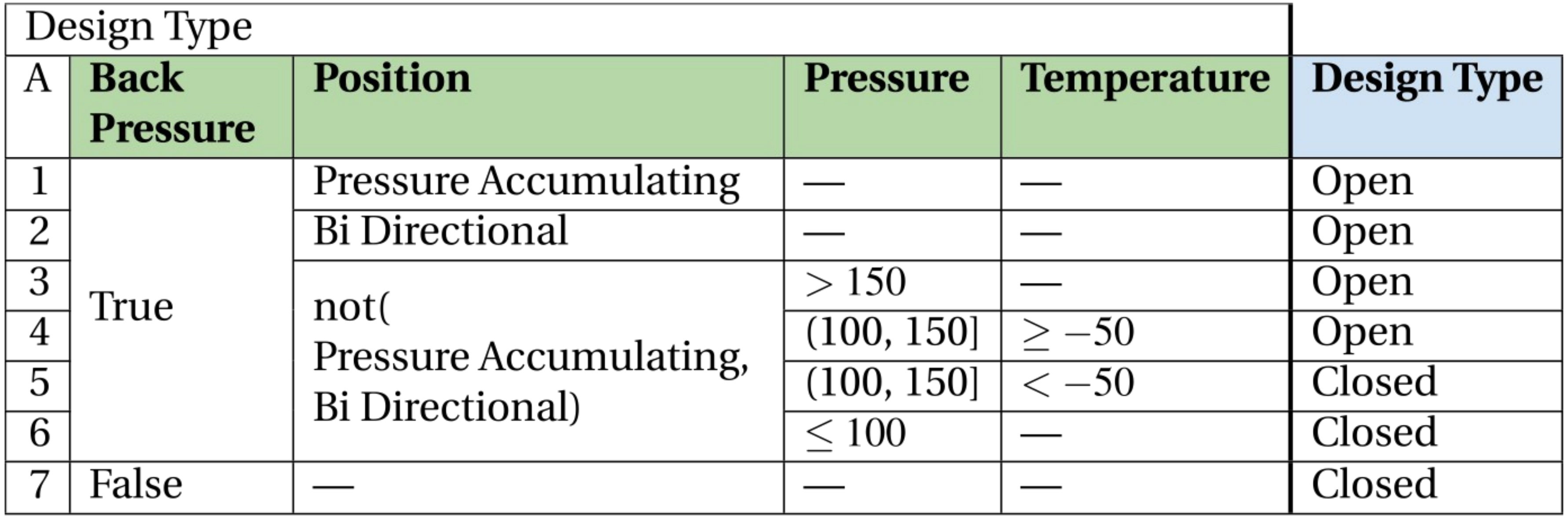
Beslissingsregels voor het generen van het CST																
Lidstaat van de vaccinatie of lidstaat van de test	Lidstaat van de EU, EER, Verenigd Koninkrijk of Zwitserland															Nee
Geboortejaar betrokkene	2009 of later	Vroeger dan 2009														-
Datum eerste positieve testresultaat op certificaat van herstel	-	180 dagen of minder geleden	Meer dan 180 dagen geleden													-
Vaccinatiestatus op vaccinatiecertificaat	-	-	1/1 of 2/2							Andere waarde						-
Vaccinatiedatum op vaccinatiecertificaat	-	-	14 dagen of meer geleden	Minder dan 14 dagen geleden						-						-
Resultaat van de test op testcertificaat	-	-	-	Negatief					Andere waarde	Negatief					Andere waarde	-
Datum van afname op testcertificaat	-	-	-	3 dagen geleden		2 dagen of minder geleden		Andere waarde	-	3 dagen geleden		2 dagen of minder geleden		Andere waarde	-	-
Type test op testcertificaat	-	-	-	NAAT	Andere waarde	NAAT of Erkende RAT	Andere waarde	-	-	NAAT	Andere waarde	NAAT of Erkende RAT	Andere waarde	-	-	-
CST geeft toegang tot evenement	x	x	x	x	-	x	-	-	-	x	-	x	-	-	-	-
CST geeft geen toegang tot evenement	-	-	-	-	x	-	x	x	x	-	x	-	x	x	x	x
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

KU LEUVEN

Approach

- Workshops with engineering team
  → Decision Requirements Diagram
- Team fills out and maintains tables

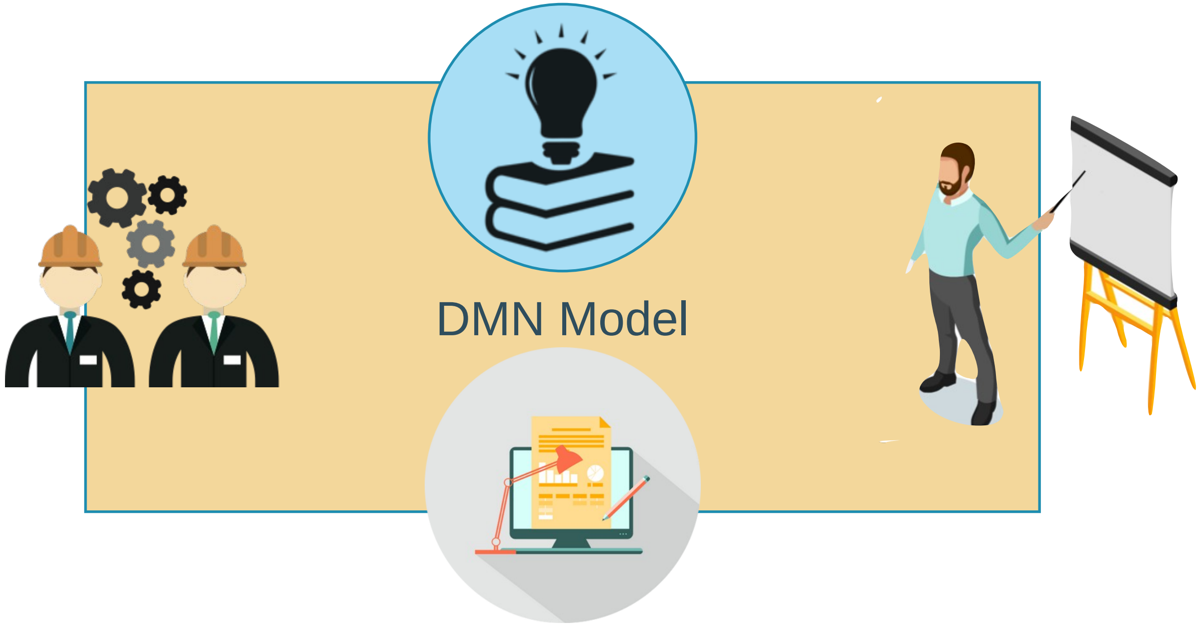




Design Type

A	**Back Pressure**	**Position**	**Pressure**	**Temperature**	**Design Type**
1	True	Pressure Accumulating	—	—	Open
2		Bi Directional	—	—	Open
3		not(Pressure Accumulating, Bi Directional)	> 150	—	Open
4			(100, 150]	≥ -50	Open
5			(100, 150]	< -50	Closed
6			≤ 100	—	Closed
7	False	—	—	—	Closed

Joint interactive modeling

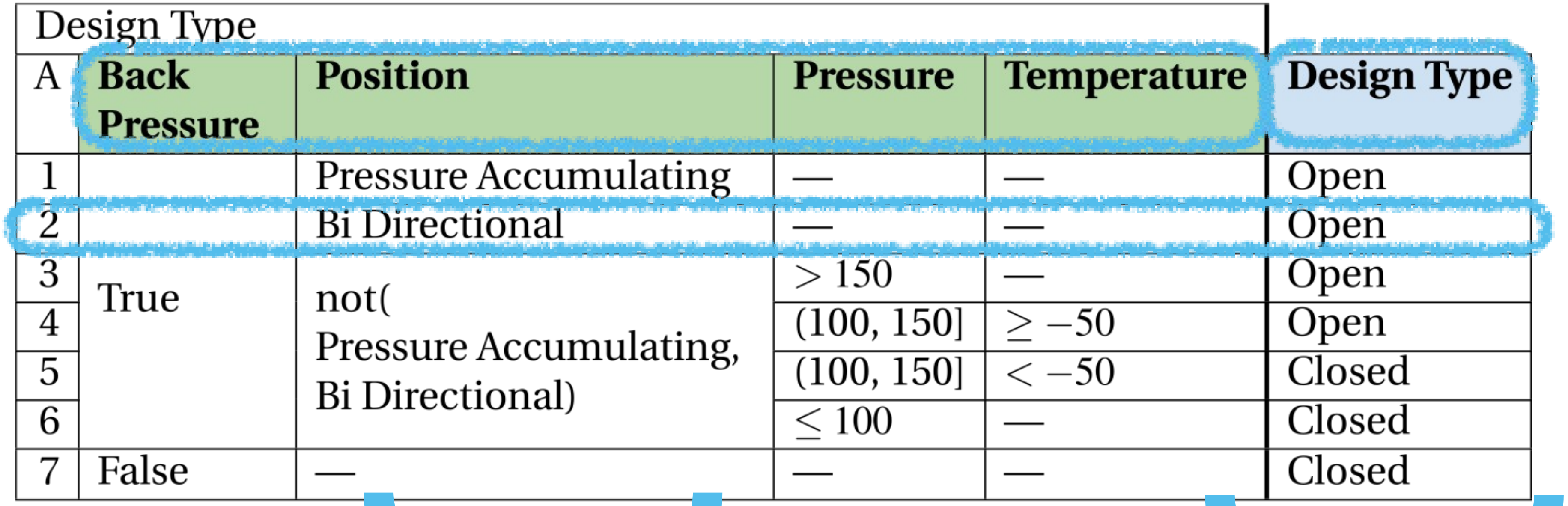


- Reliable: no misunderstandings
- Maintainable: no implementation delays
- Ownership stays with the experts

Hands-on DMN

Input variable(s)

Decision variable(s)



Design Type

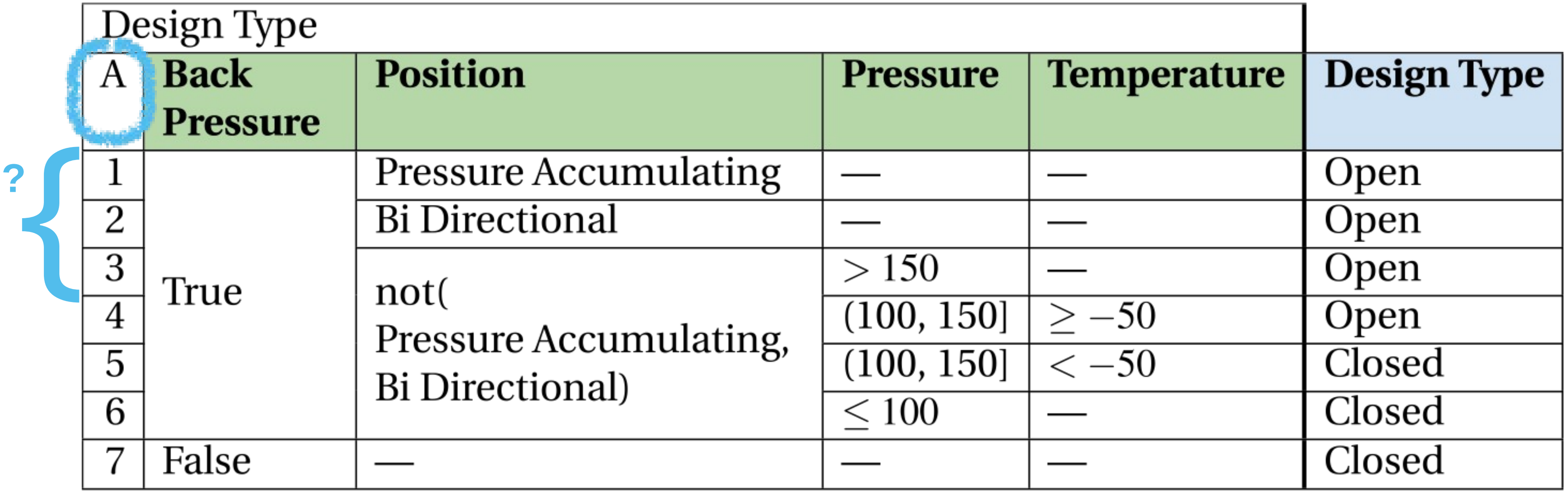
A	Back Pressure	Position	Pressure	Temperature	Design Type
1		Pressure Accumulating	—	—	Open
2		Bi Directional	—	—	Open
3	True	not(Pressure Accumulating, Bi Directional)	> 150	—	Open
4			$(100, 150]$	≥ -50	Open
5			$(100, 150]$	< -50	Closed
6			≤ 100	—	Closed
7	False	—	—	—	Closed

Case

Condition

- *value*
- not(*value*)
- $v_1, \ldots, v_n$
- [*i*, *j*]
- <= *n*
- –

Single value



Design Type

A	Back Pressure	Position	Pressure	Temperature	Design Type
1	True	Pressure Accumulating	—	—	Open
2		Bi Directional	—	—	Open
3		not(Pressure Accumulating, Bi Directional)	> 150	—	Open
4			$(100, 150]$	≥ -50	Open
5			$(100, 150]$	< -50	Closed
6			≤ 100	—	Closed
7	False	—	—	—	Closed

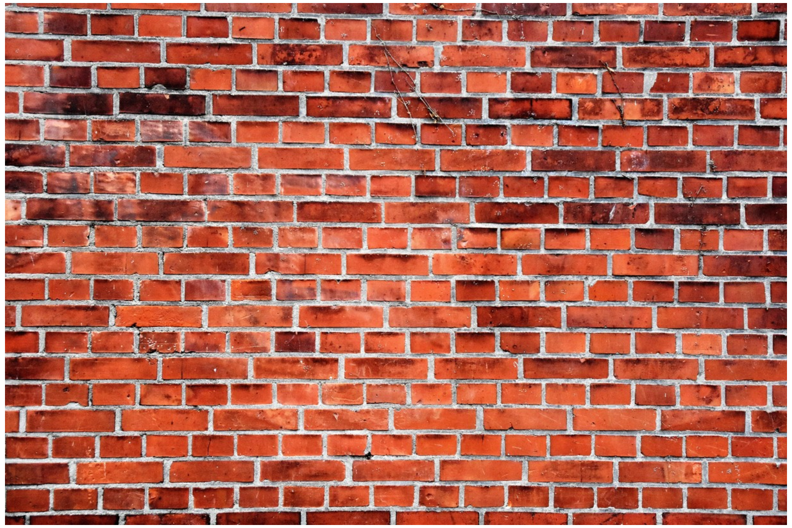
Overlap?

Hit policy

- **U**nique
- **F**irst
- **A**ny
- **C**ollect, e.g., **C+**

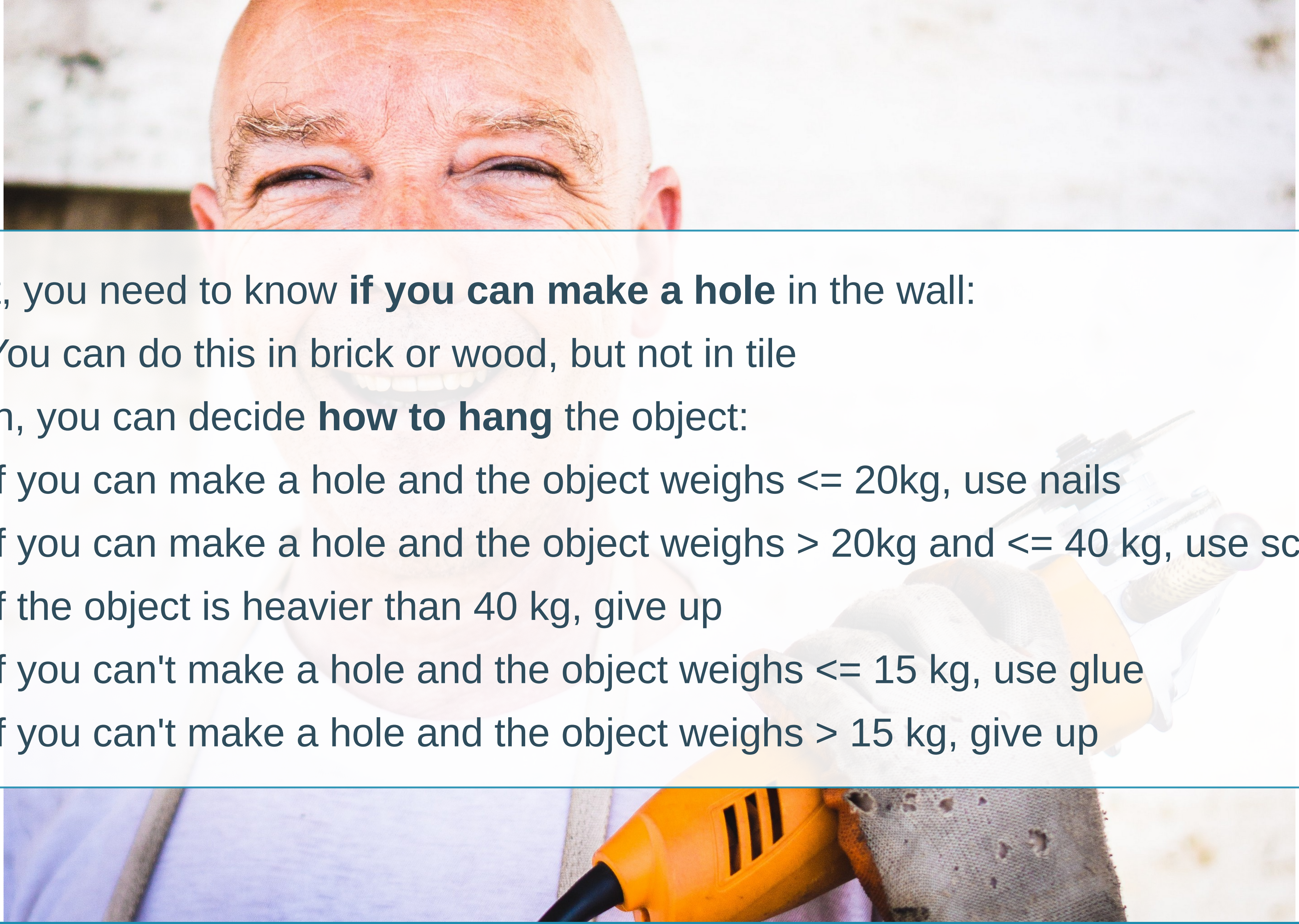
Running example

- Decision support for hanging object on our wall
  - Nail or screw or glue?



Photos by Chris Yates and Waldemar on Unsplash

Photo by David Siglin on Unsplash



- First, you need to know **if you can make a hole** in the wall:
  - You can do this in brick or wood, but not in tile
- Then, you can decide **how to hang** the object:
  - If you can make a hole and the object weighs <= 20kg, use nails
  - If you can make a hole and the object weighs > 20kg and <= 40 kg, use screws
  - If the object is heavier than 40 kg, give up
  - If you can't make a hole and the object weighs <= 15 kg, use glue
  - If you can't make a hole and the object weighs > 15 kg, give up

Photo by David Siglin on Unsplash

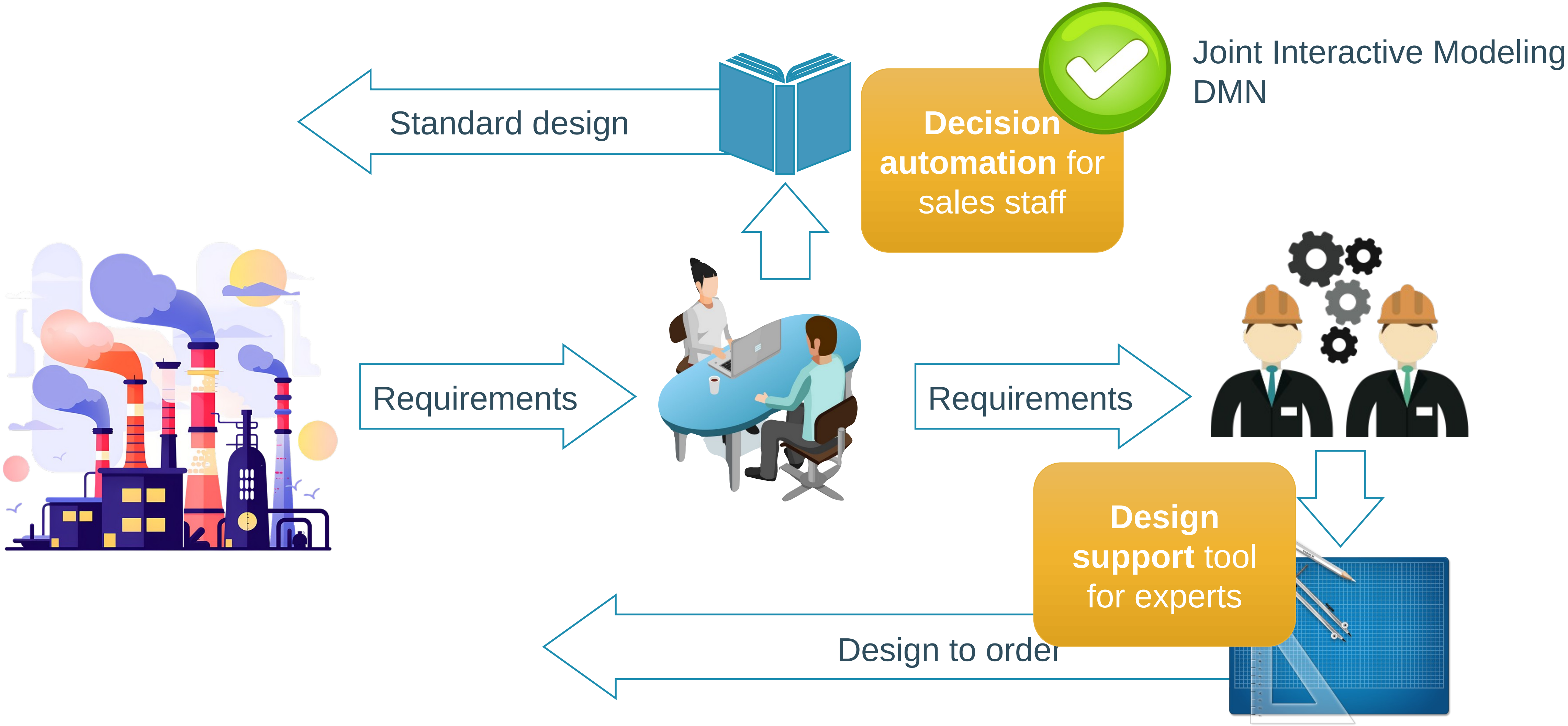
Commercial tool

https://demo.bpmn.io/dmn

slides.simonvandevelde.be/workshop-ac

Try it: Handyman in DMN

Goals



Joint Interactive Modeling
DMN

Decision automation for sales staff

Standard design

Requirements

Requirements

Design support tool for experts

Design to order

jemima @ https://www.cleanpng.com

Limitations

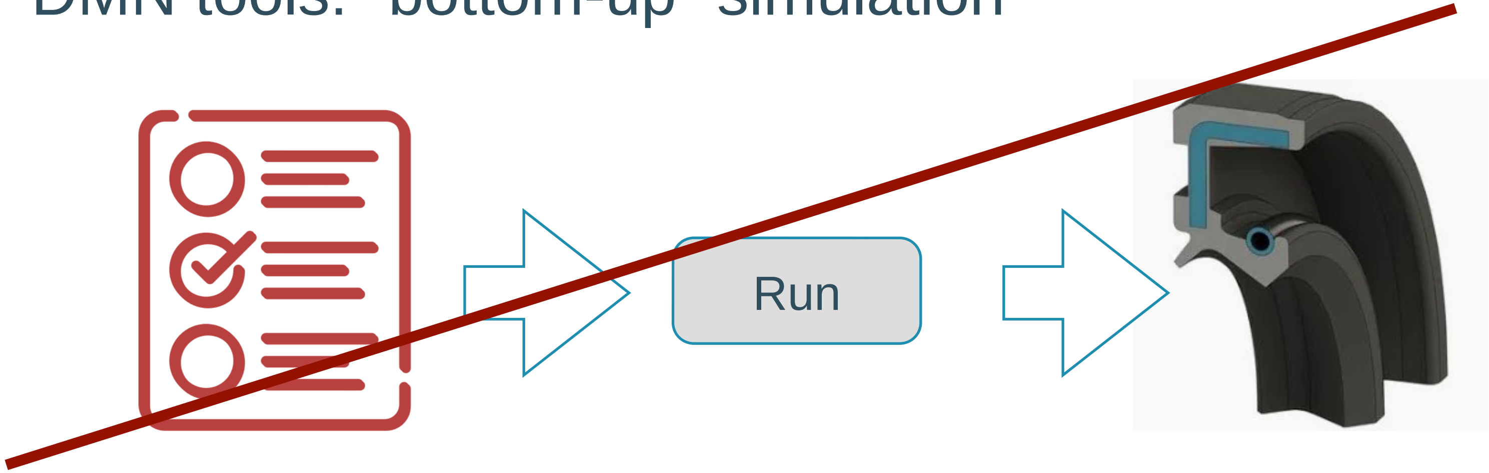
Limitations of...

- DMN tools: "bottom-up" simulation

- DMN language: deterministic decision processes

Limitations of...

- DMN tools: "bottom-up" simulation

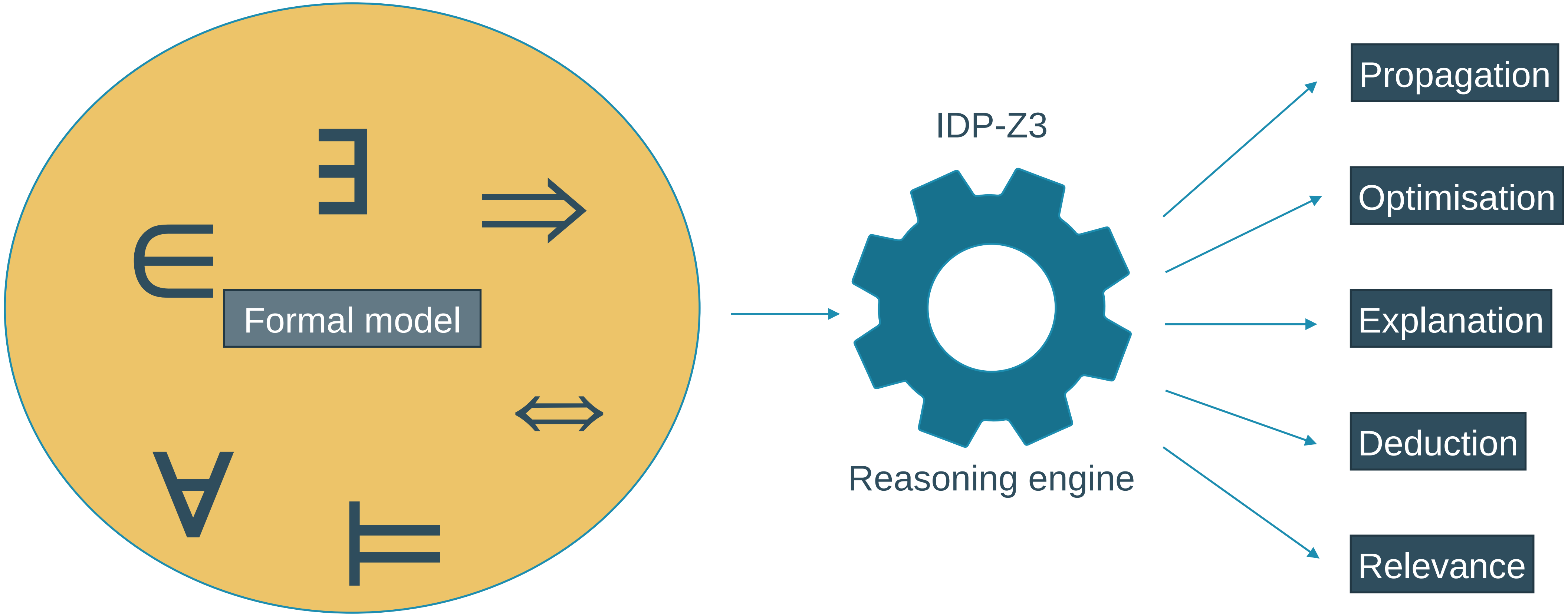


Expert users

- Responsibility
- More information
- Integrate into their proces

- DMN language: deterministic decision processes

Knowledge-base paradigm



Formal model

IDP-Z3

Reasoning engine

Propagation

Optimisation

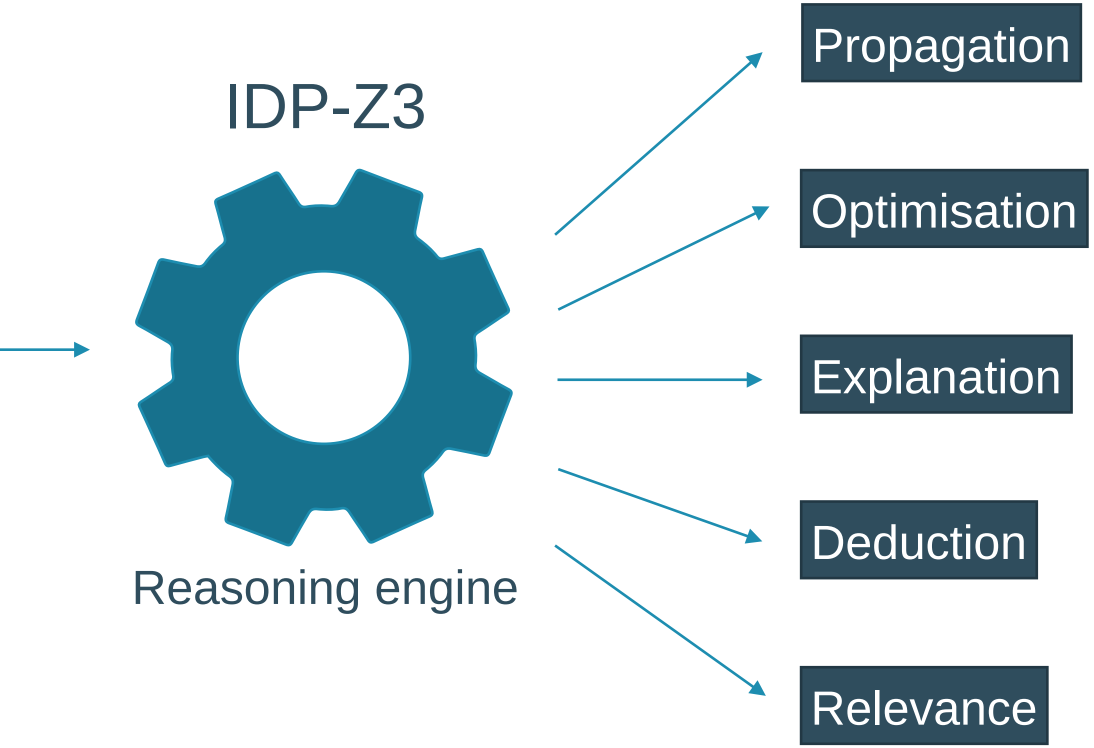
Explanation

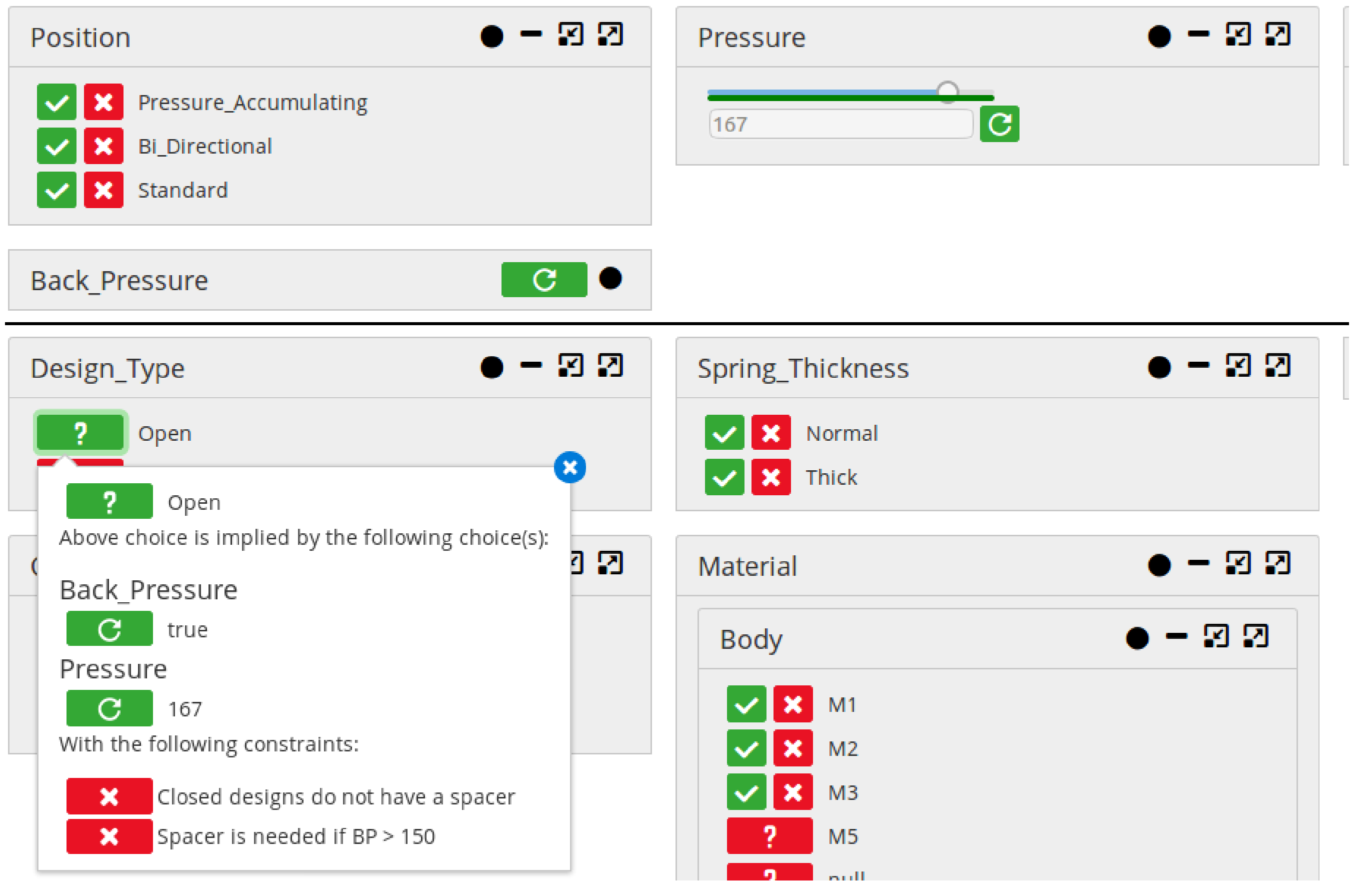
Deduction

Relevance

≠ single fixed work-flow

Interactive Consultant




Position ● ━ ⧉ ⧉

✓ ✗ Pressure_Accumulating
✓ ✗ Bi_Directional
✓ ✗ Standard

Pressure ● ━ ⧉ ⧉

167

Back_Pressure ●

Design_Type ● ━ ⧉ ⧉

? Open

? Open

Above choice is implied by the following choice(s):

Back_Pressure

true

Pressure

167

With the following constraints:

✗ Closed designs do not have a spacer

✗ Spacer is needed if BP > 150

Spring_Thickness ● ━ ⧉ ⧉

✓ ✗ Normal
✓ ✗ Thick

Material ● ━ ⧉ ⧉

Body ● ━ ⧉ ⧉

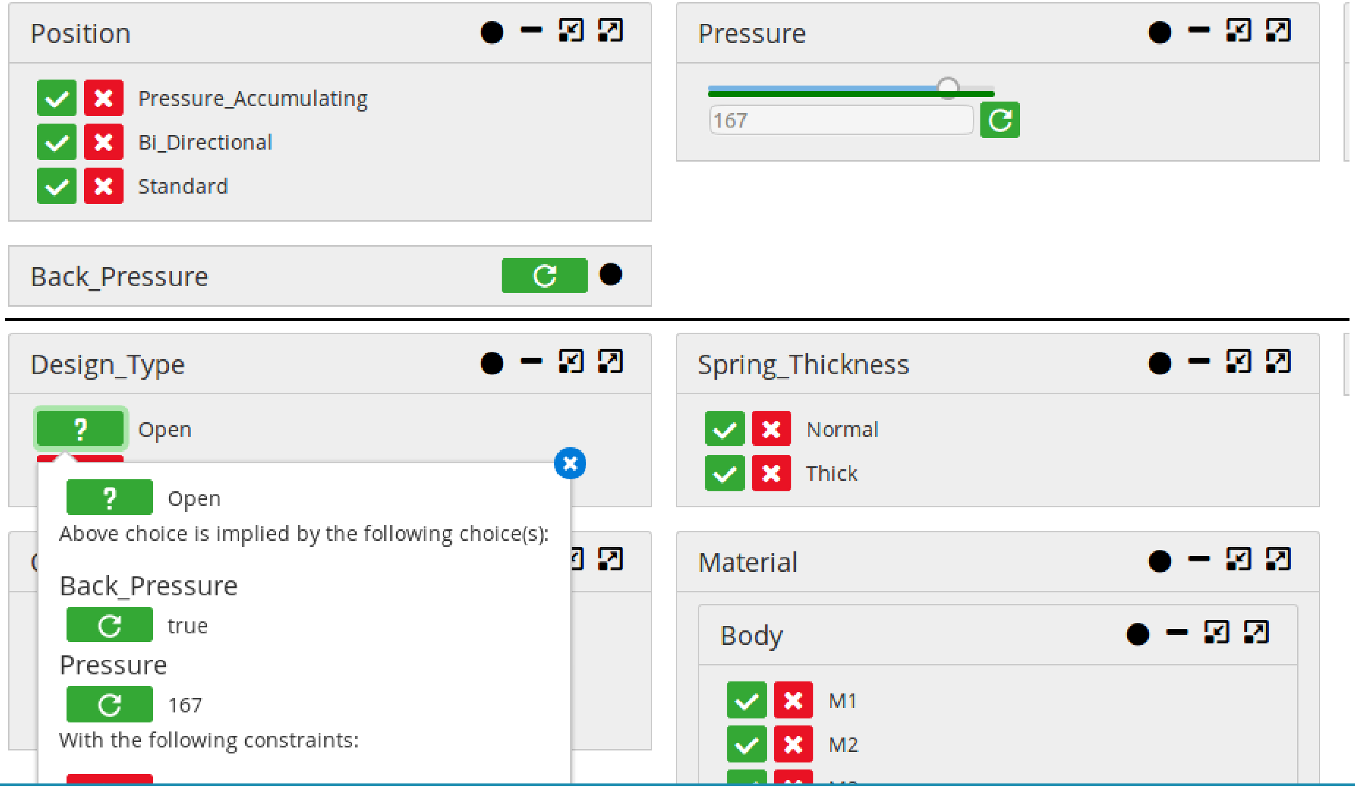
✓ ✗ M1
✓ ✗ M2
✓ ✗ M3
? M5
? null

Single KB

- Maintainable
- Flexible

- Which designs can withstand this pressure?
- What is the maximum temperature that this design can withstand?
- What is the cheapest design for these requirements?
- Why is this design not possible for these requirements?

[Pierre Carbonnelle, Marc Denecker, et al.]



Position

✓ ✗ Pressure_Accumulating

✓ ✗ Bi_Directional

✓ ✗ Standard

Pressure

167

Back_Pressure ●

Design_Type

? Open

? Open

Above choice is implied by the following choice(s):

Back_Pressure

true

Pressure

167

With the following constraints:

Spring_Thickness

✓ ✗ Normal

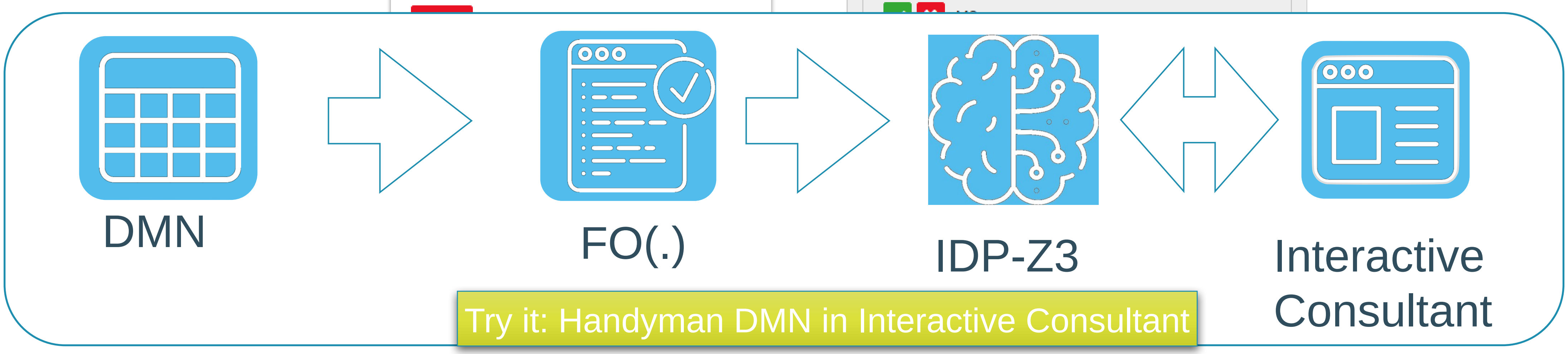
✓ ✗ Thick

Material

Body

✓ ✗ M1

✓ ✗ M2



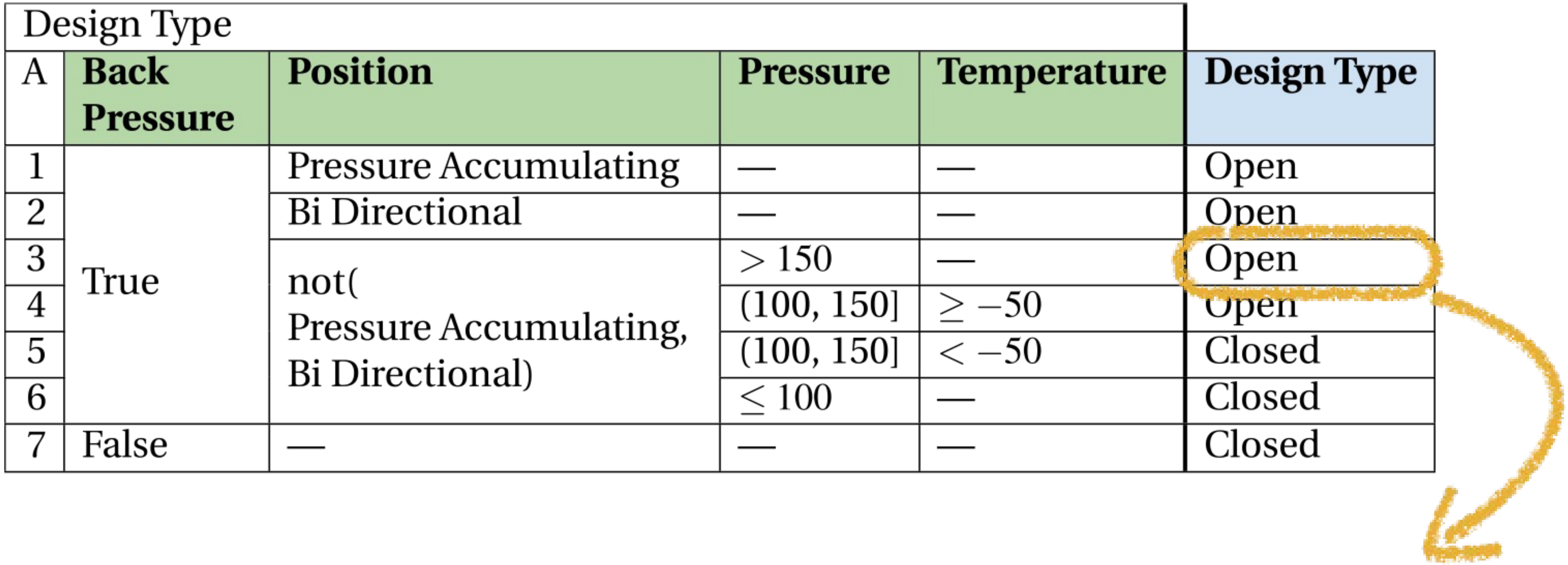
Try it: Handyman DMN in Interactive Consultant

Limitations of...

- DMN tools: "bottom-up" simulation

- DMN language: deterministic decision processes

Limitation: DMN language itself



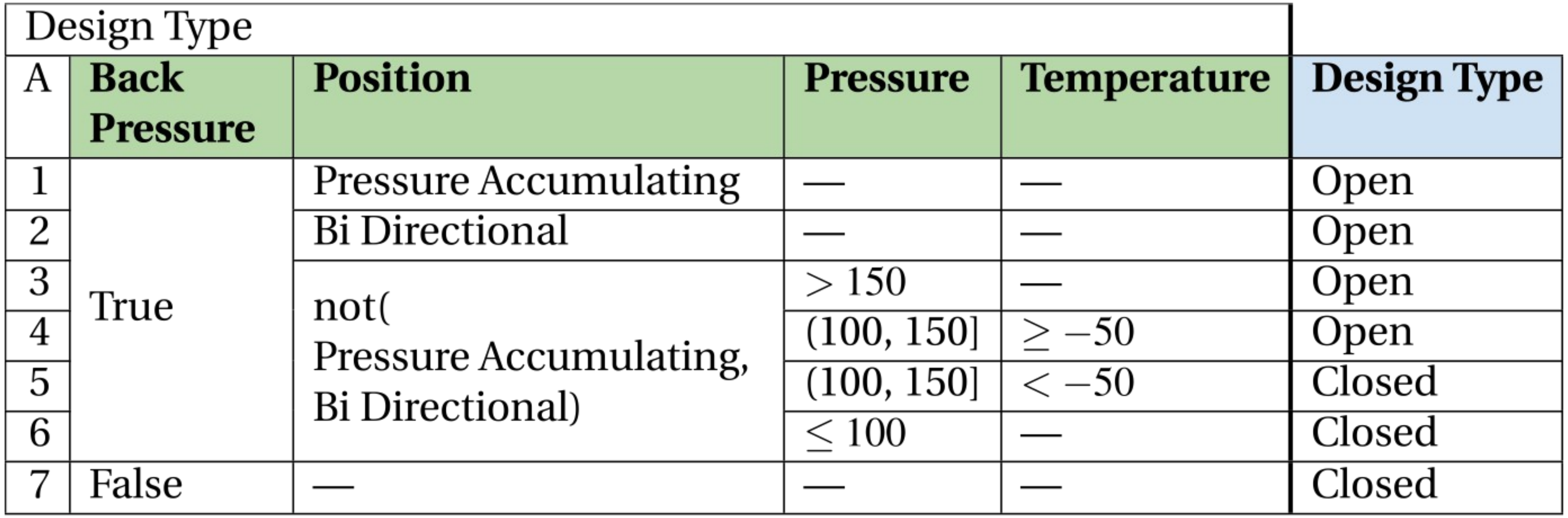
Design Type

A	Back Pressure	Position	Pressure	Temperature	Design Type
1	True	Pressure Accumulating	—	—	Open
2		Bi Directional	—	—	Open
3		not(Pressure Accumulating, Bi Directional)	> 150	—	Open
4			(100, 150]	≥ -50	Open
5			(100, 150]	< -50	Closed
6			≤ 100	—	Closed
7	False	—	—	—	Closed

What but not *why*

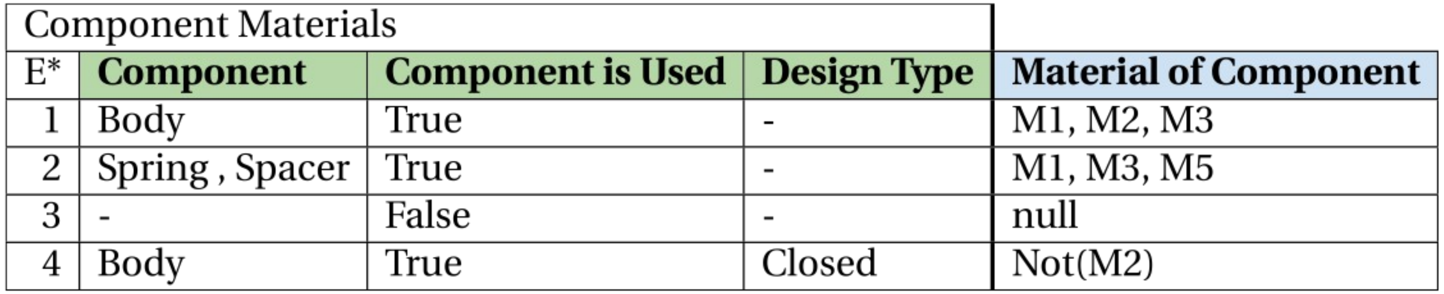
Constraints + preferences

For design support



Design Type					
A	**Back Pressure**	**Position**	**Pressure**	**Temperature**	**Design Type**
1	True	Pressure Accumulating	—	—	Open
2		Bi Directional	—	—	Open
3		not(Pressure Accumulating, Bi Directional)	> 150	—	Open
4			$(100, 150]$	≥ -50	Open
5			$(100, 150]$	< -50	Closed
6			≤ 100	—	Closed
7	False	—	—	—	Closed

From DMN ...



Component Materials				
E*	**Component**	**Component is Used**	**Design Type**	**Material of Component**
1	Body	True	-	M1, M2, M3
2	Spring , Spacer	True	-	M1, M3, M5
3	-	False	-	null
4	Body	True	Closed	Not(M2)

... to cDMN

- If you can make a hole and the object weighs (20,40] kg, use screws
- If you can make a hole and the object weighs <= 20kg, use nails
- If you can't make a hole and the object weighs <= 15 kg, use glue

Can I glue an object to a brick wall?

Can I use screws with object of 10kg?

Is there a more reliable option than nails to hang a 19kg object?

The price of screws just went down -- what changes?

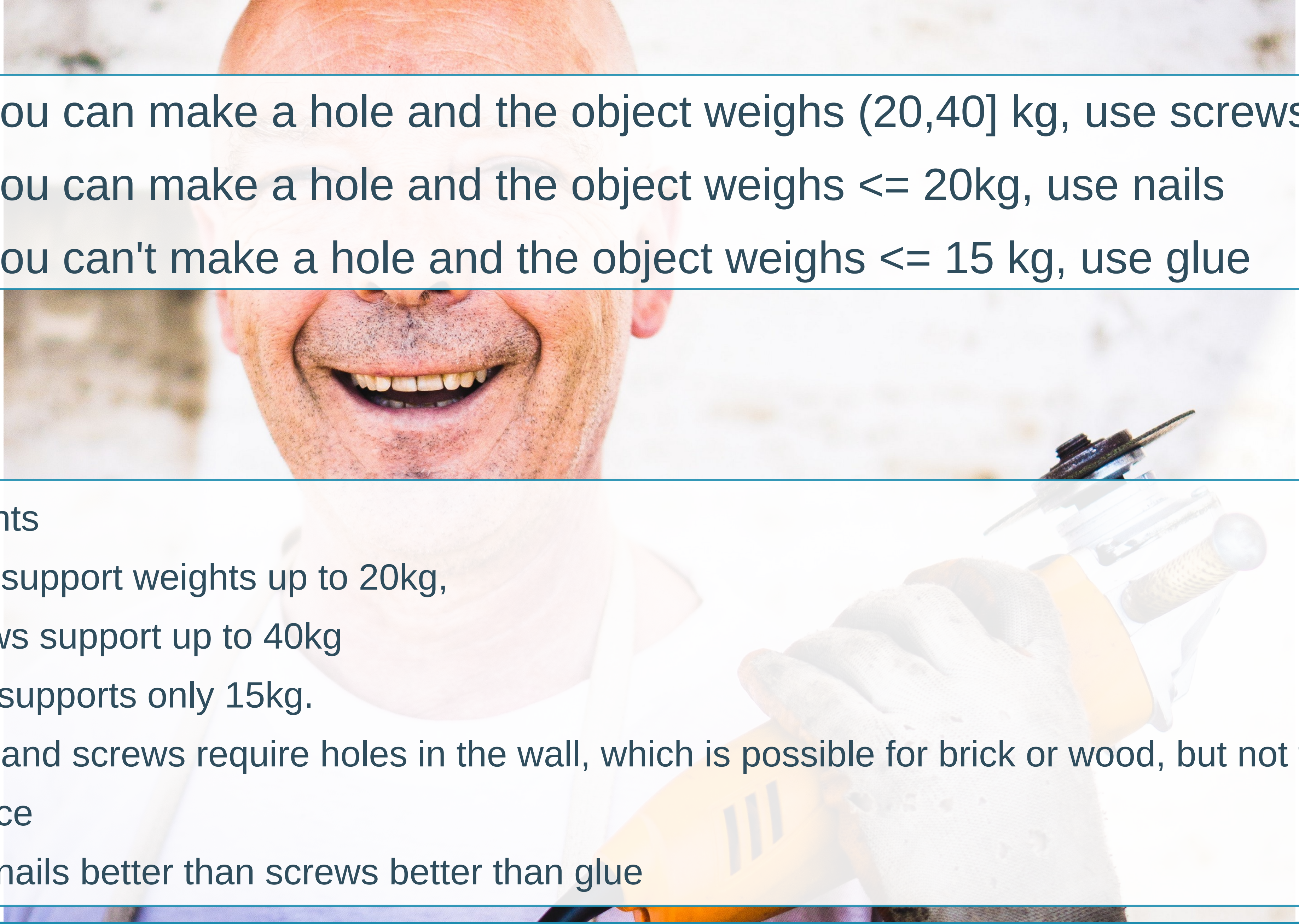
- If you can make a hole and the object weighs (20,40] kg, use screws
- If you can make a hole and the object weighs <= 20kg, use nails
- If you can't make a hole and the object weighs <= 15 kg, use glue

What

Constraints + preferences

- Constraints
  - Nails support weights up to 20kg,
  - Screws support up to 40kg
  - Glue supports only 15kg.
  - Nails and screws require holes in the wall, which is possible for brick or wood, but not for tiles.
- Preference
  - I like nails better than screws better than glue

Why



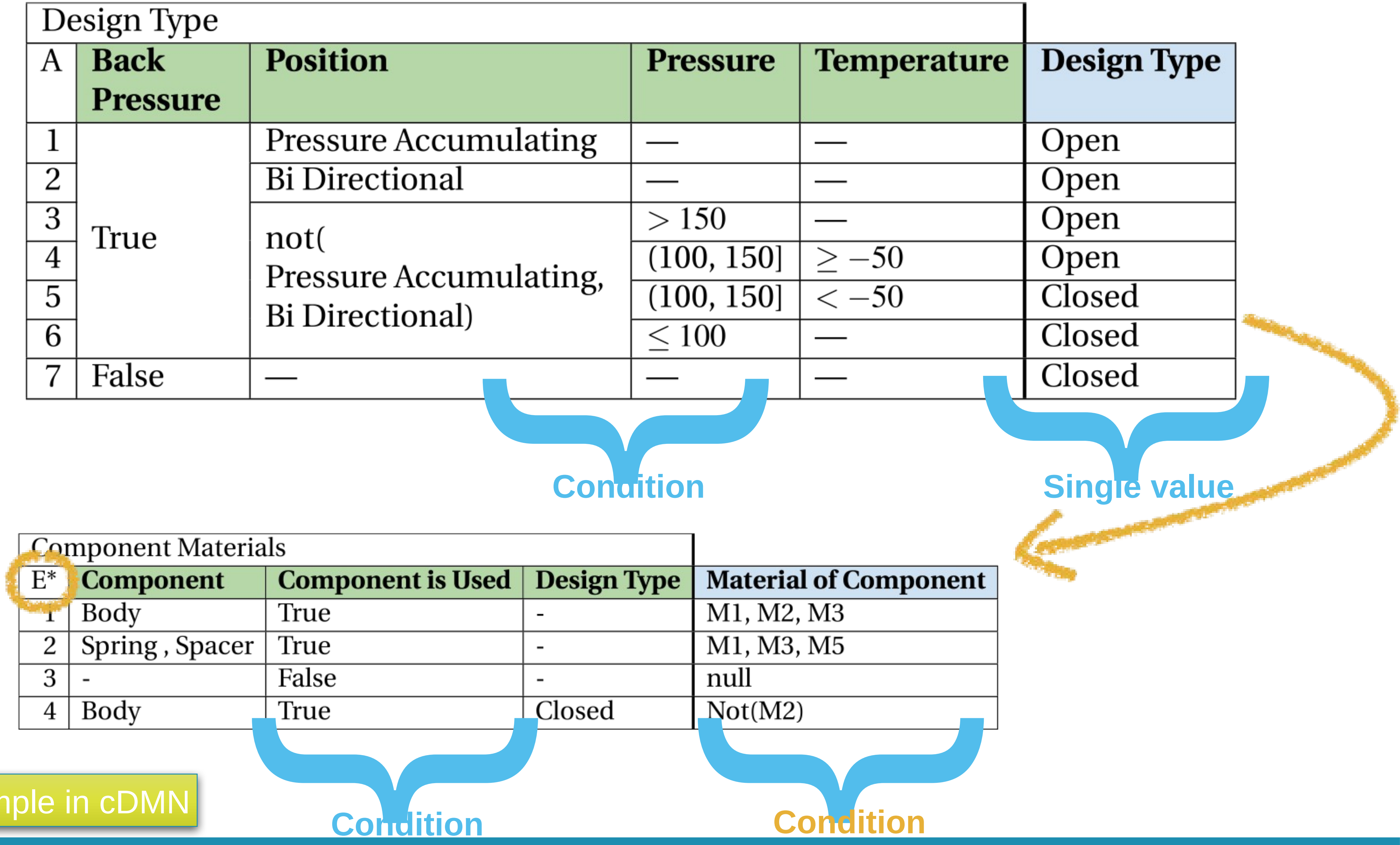
- If you can make a hole and the object weighs (20,40] kg, use screws
- If you can make a hole and the object weighs <= 20kg, use nails
- If you can't make a hole and the object weighs <= 15 kg, use glue

- Constraints
  - Nails support weights up to 20kg,
  - Screws support up to 40kg
  - Glue supports only 15kg.
  - Nails and screws require holes in the wall, which is possible for brick or wood, but not for tiles.
- Preference
  - I like nails better than screws better than glue

Hit policy:
U,F,A,C

Decision table

Constraint table



Design Type

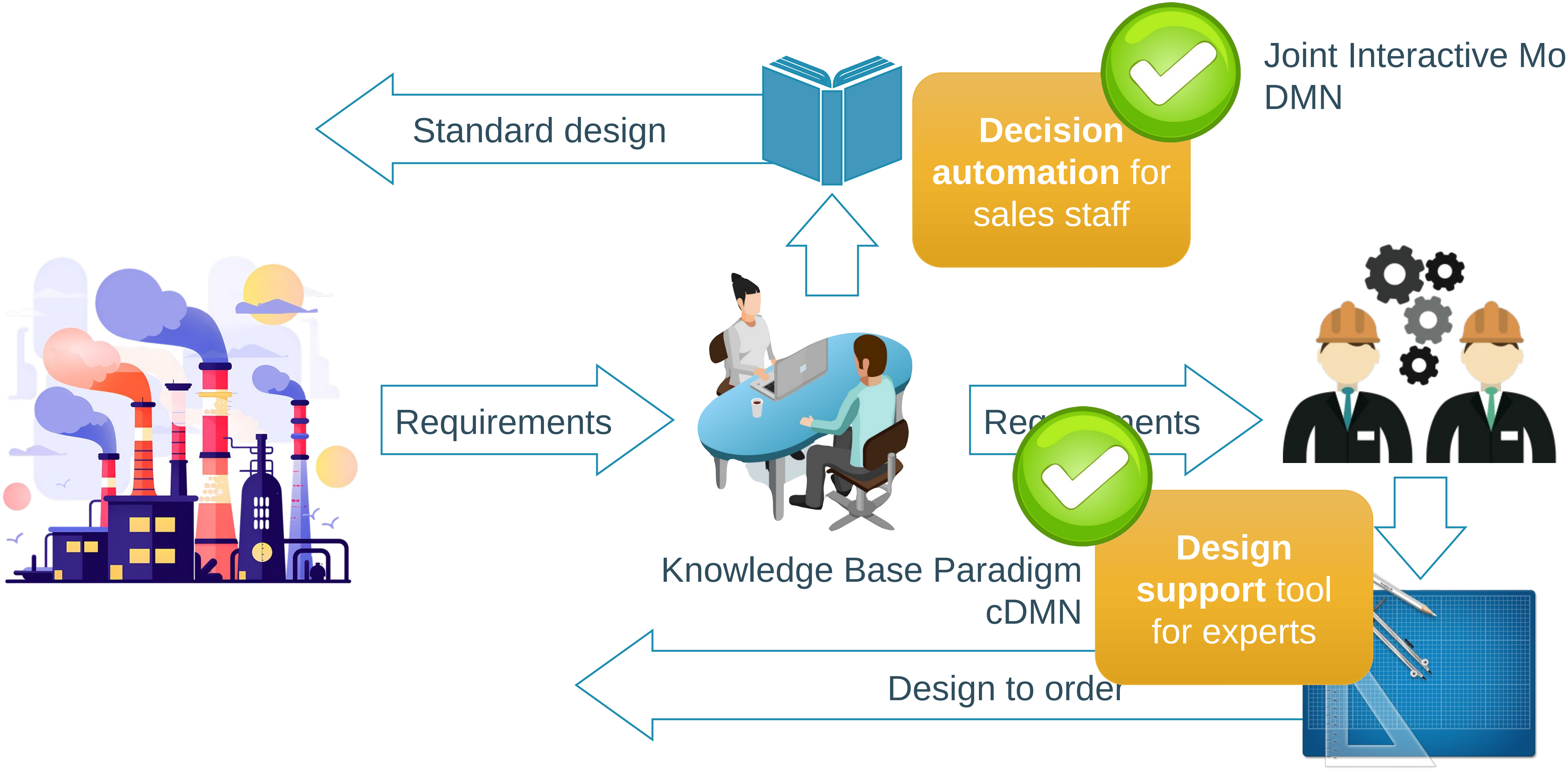
A	Back Pressure	Position	Pressure	Temperature	Design Type
1	True	Pressure Accumulating	—	—	Open
2		Bi Directional	—	—	Open
3		not(Pressure Accumulating, Bi Directional)	> 150	—	Open
4			$(100, 150]$	≥ -50	Open
5			$(100, 150]$	< -50	Closed
6			≤ 100	—	Closed
7	False	—	—	—	Closed

Component Materials

E*	Component	Component is Used	Design Type	Material of Component
1	Body	True	-	M1, M2, M3
2	Spring , Spacer	True	-	M1, M3, M5
3	-	False	-	null
4	Body	True	Closed	Not(M2)

Try it: Handyman example in cDMN

Goals



Standard design

Decision automation for sales staff

Joint Interactive Modeling
DMN

Requirements

Requirements

Knowledge Base Paradigm
cDMN

Design support tool for experts

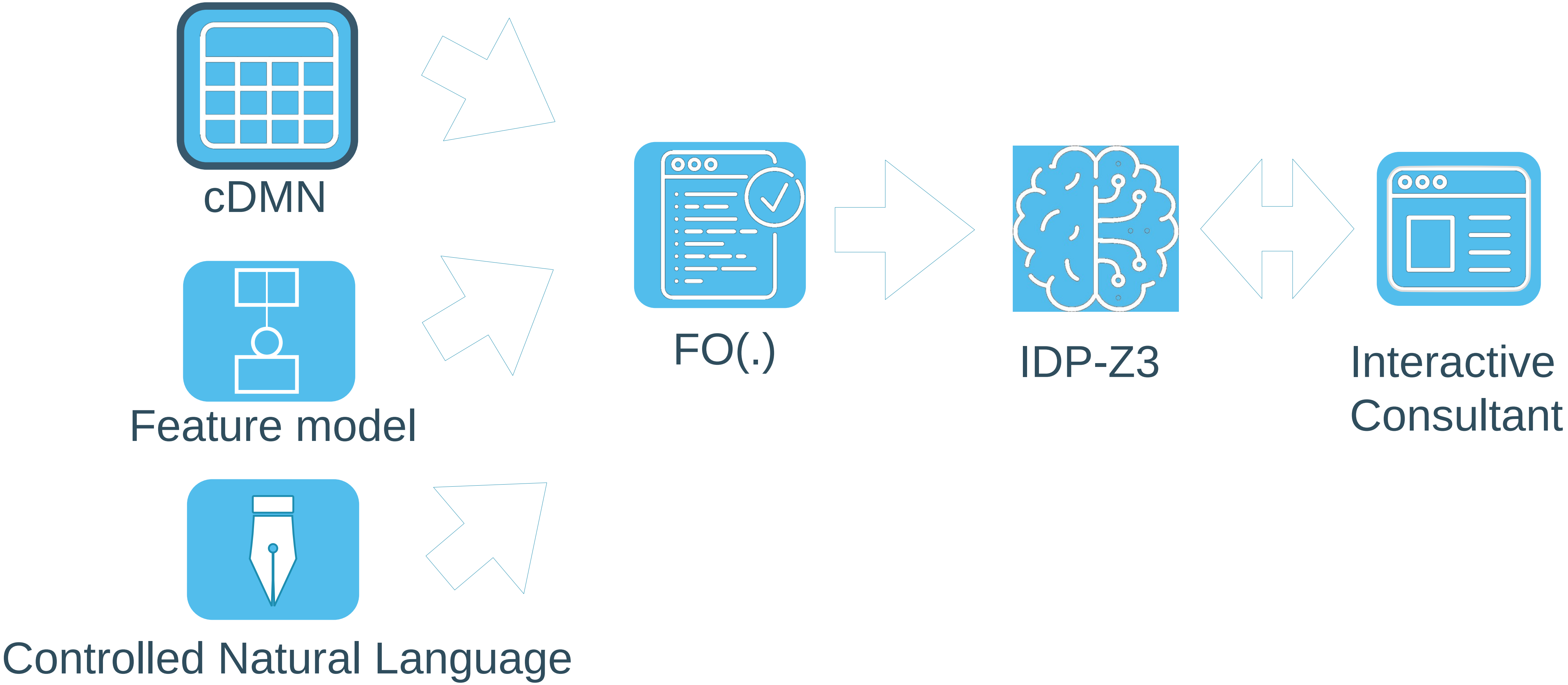
Design to order

jemima @ https://www.cleanpng.com

- Domain expert friendly
- Joint Interactive modeling

- Rich
- Modular

- Knowledge Base Paradigm
- User-friendly interaction



cDMN

Feature model

Controlled Natural Language

FO(.)

IDP-Z3

Interactive Consultant

A deeper look:
At the inference tasks

```
wall := Brick.
method := Nail.
weight := 1.
```

```
wall := Brick.
method := Glue.
weight := 1.
```

```
wall := Brick.
method := Nail.
weight := 2.
```

```
wall := Brick.
method := Nail.
weight := 24.
```

...

Knowledge

About the difference between good and bad situations

- Nails support weights up to 20kg, screws support up to 40kg and glue supports only 15kg.
- Nails and screws require holes in the wall, which is possible for brick or wooden walls, but not for tiles.

?

Models

```
wall := Brick.
method := Nail.
weight := 1.
```

```
wall := Brick.
method := Glue.
weight := 1.
```

```
wall := Brick.
method := Nail.
weight := 2.
```

Non models

```
wall := Brick.
method := Nail.
weight := 24.
```

...

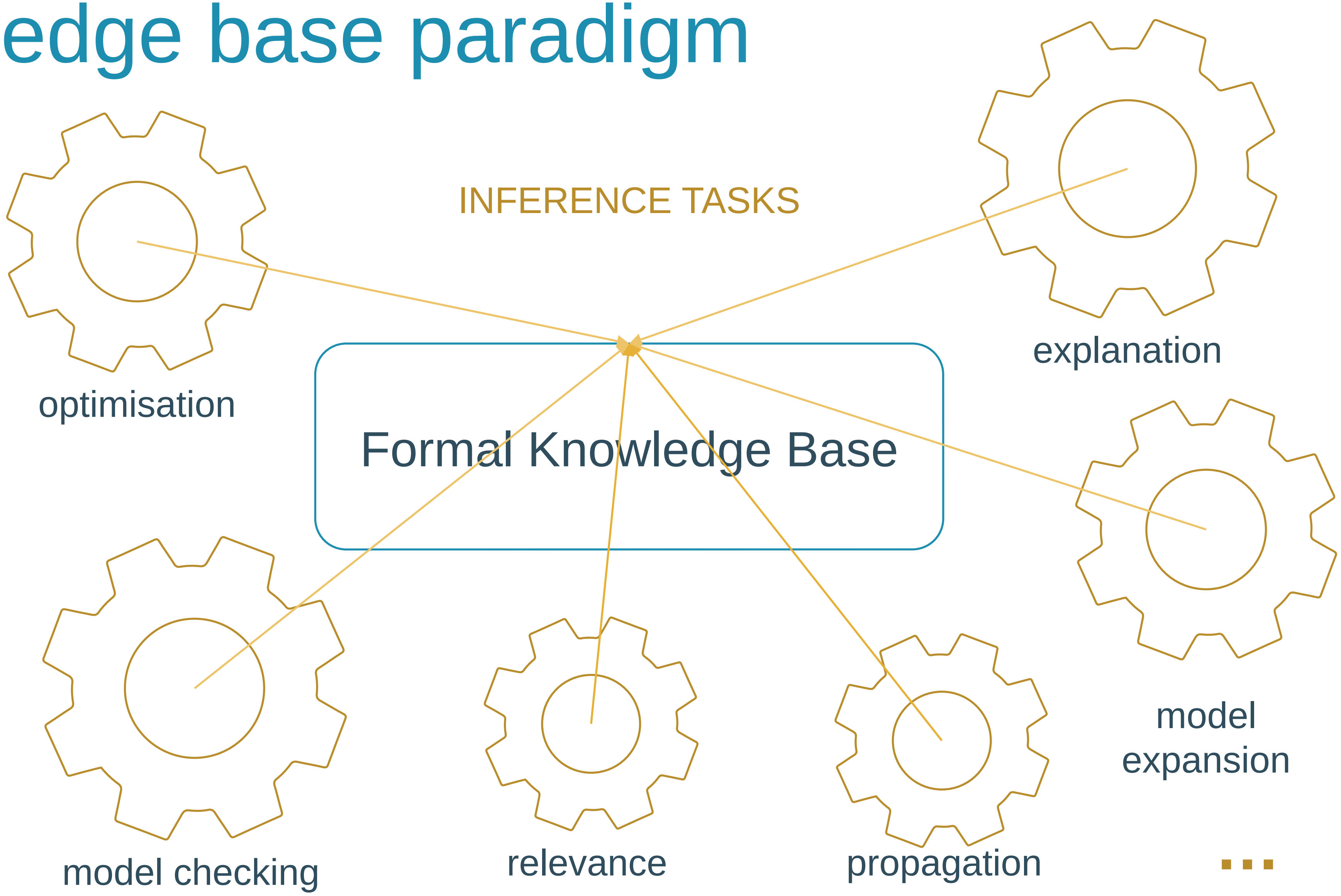
Knowledge

About the difference between good and bad situations

- Nails support weights up to 20kg, screws support up to 40kg and glue supports only 15kg.
- Nails and screws require holes in the wall, which is possible for brick or wooden walls, but not for tiles.

?

Knowledge base paradigm



INFERENCE TASKS

Formal Knowledge Base

optimisation

explanation

model expansion

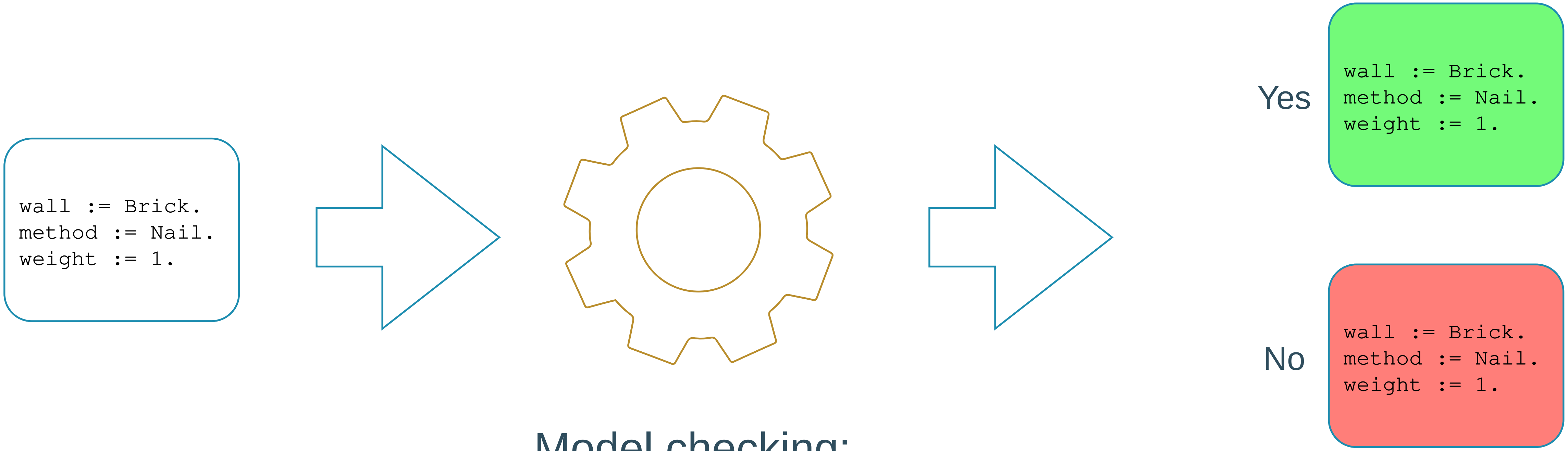
model checking

relevance

propagation

...

all defined in a model-theoretic way



```
wall  := Brick.
method := Nail.
weight  := 1.
```

Yes

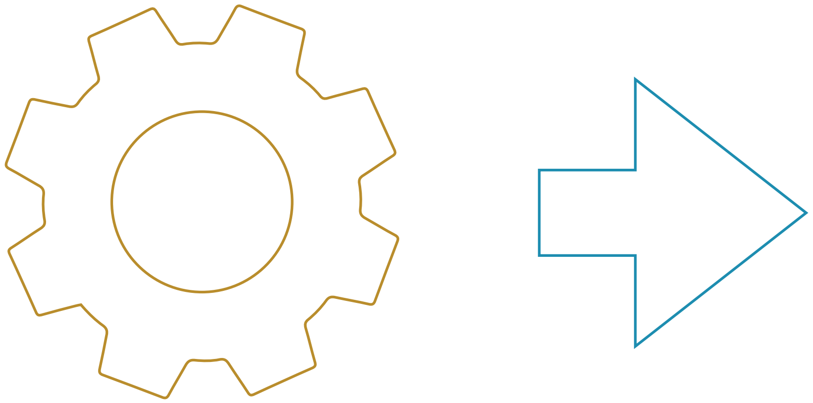
```
wall  := Brick.
method := Nail.
weight  := 1.
```

No

```
wall  := Brick.
method := Nail.
weight  := 1.
```

Model checking:

Is a situation good?



```
wall  := Brick.
method := Nail.
weight := 1.
```

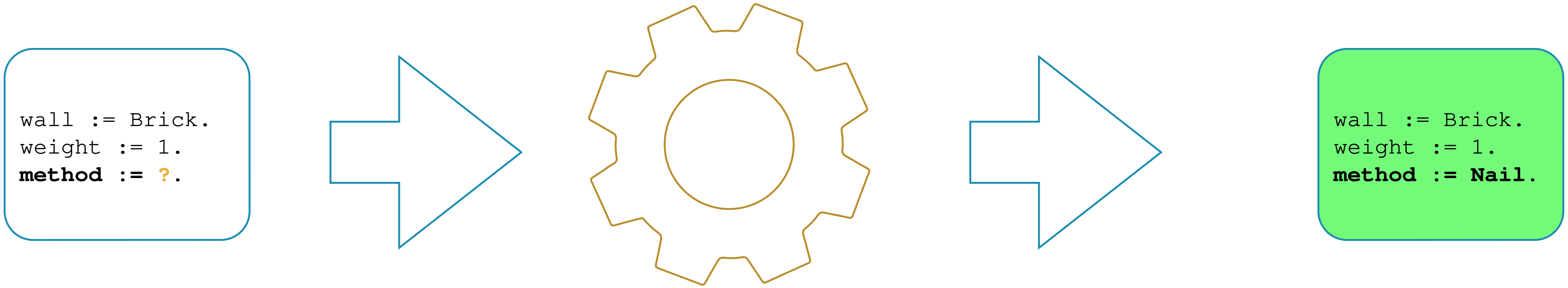
```
wall  := Brick.
method := Glue.
weight := 1.
```

```
wall  := Brick.
method := Nail.
weight := 2.
```

Model generation:

Find (one or more) good situation(s)

Generalisation



```
wall := Brick.
weight := 1.
method := ?.
```

```
wall := Brick.
weight := 1.
method := Nail.
```

Model expansion:
Given part of situation,
extend it to (a number of) model(s)

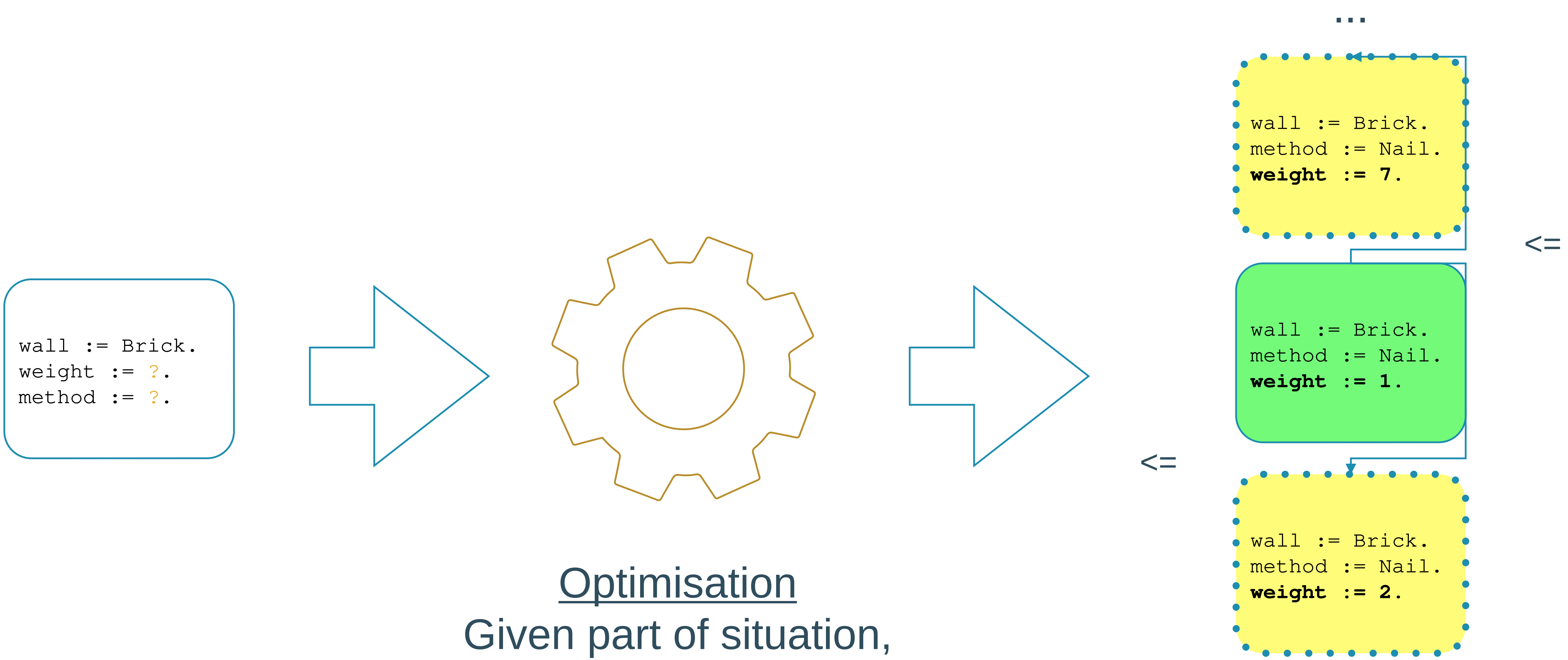
Goal
Get *N* models

Which of these can you answer?

- Which methods are suitable for hanging a 25kg weight on a brick wall?
- There is a nail in my wall. How much weight will it support?
- Can we hang a 25kg object on a tile wall using glue?
- I want to hang a heavy object on my wooden wall, without knowing precisely how heavy it is. Which method will support the most weight?
- When would we use screws instead of nails?

Which of these can you answer?

- *Which methods are suitable for hanging a 25kg weight on a brick wall?*
- There is a nail in my wall. How much weight will it support?
- *Can we hang a 25kg object on a tile wall using glue?*
- I want to hang a heavy object on my wooden wall, without knowing precisely how heavy it is. Which method will support the most weight?
- When would we use screws instead of nails?



```
wall  := Brick.
weight  := ?.
method  := ?.
```

...

```
wall  := Brick.
method  := Nail.
weight  := 7.
```

<=

```
wall  := Brick.
method  := Nail.
weight  := 1.
```

<=

```
wall  := Brick.
method  := Nail.
weight  := 2.
```

Optimisation

Given part of situation,
find a model(s) in which constant
has maximal/minimal value

Goal
Minimize *weight*

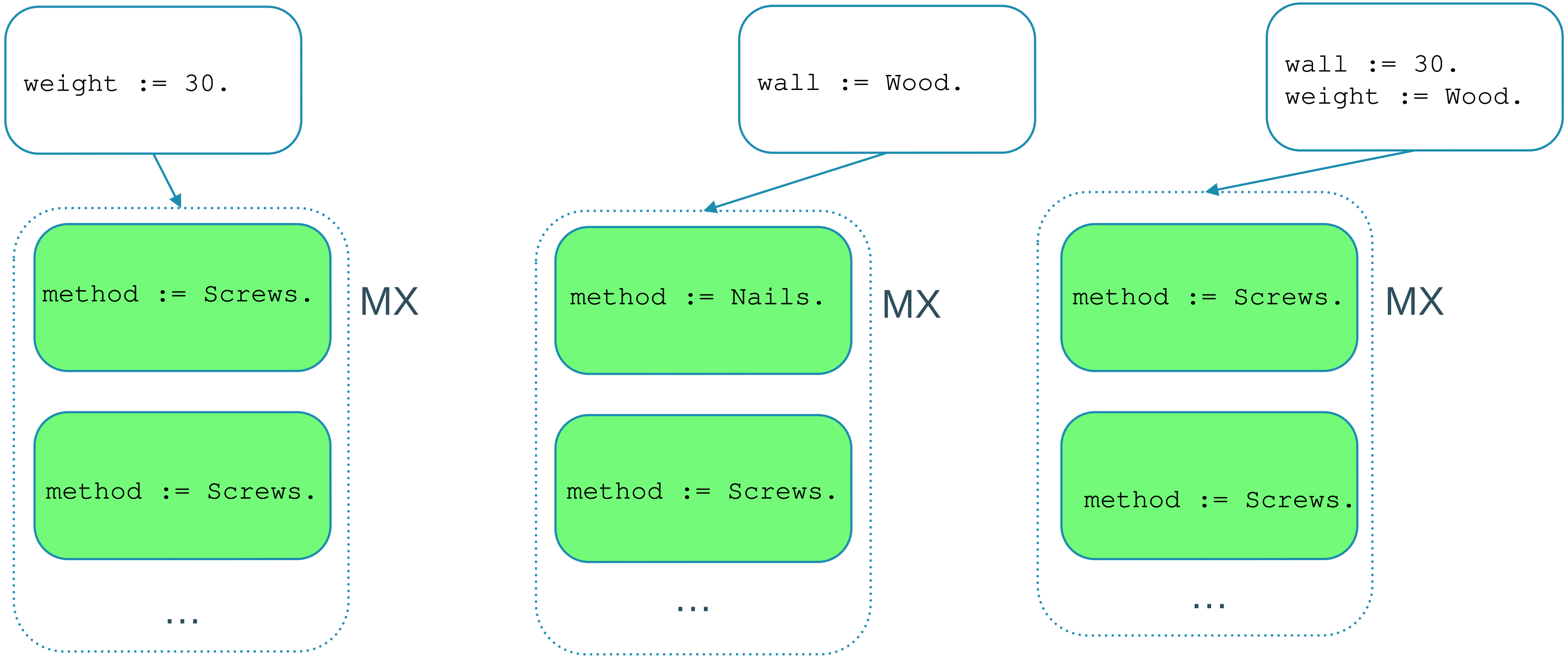
Which of these can you answer?

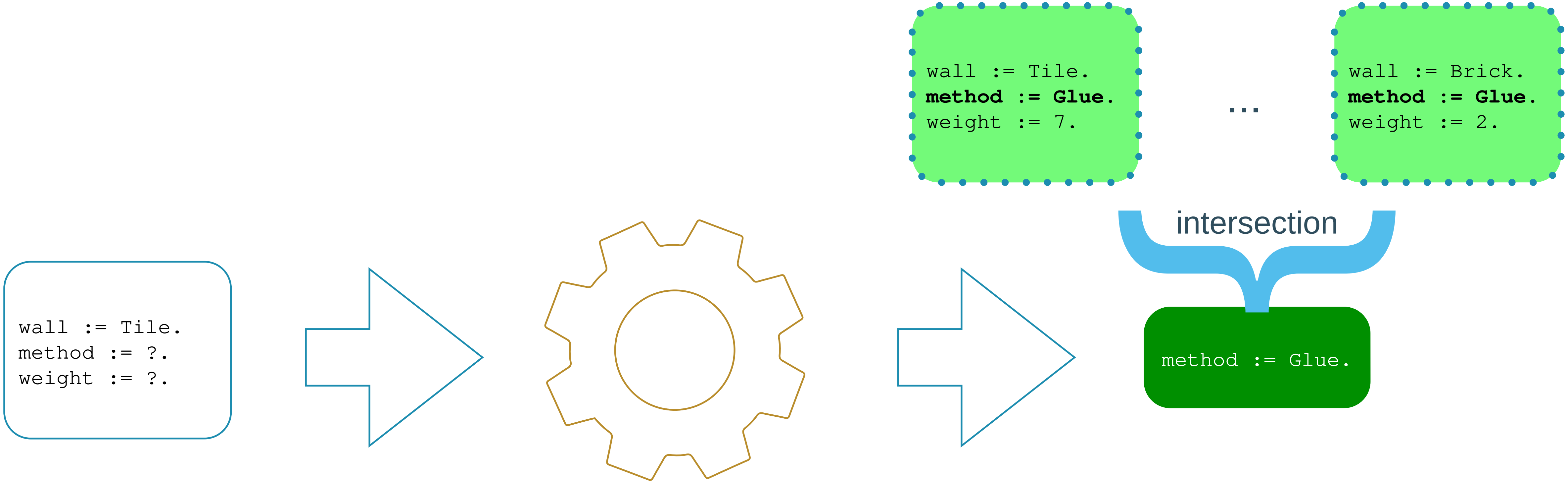
- *Which methods are suitable for hanging a 25kg weight on a brick wall?*
- There is a nail in my wall. How much weight will it support?
- *Can we hang a 25kg object on a tile wall using glue?*
- I want to hang a heavy object on my wooden wall, without knowing precisely how heavy it is. Which method will support the most weight?
- When would we use screws instead of nails?

Which of these can you answer?

- *Which methods are suitable for hanging a 25kg weight on a brick wall?*
- *There is a nail in my wall. How much weight will it support?*
- *Can we hang a 25kg object on a tile wall using glue?*
- *I want to hang a heavy object on my wooden wall, without knowing precisely how heavy it is. Which method will support the most weight?*
- When would we use screws instead of nails?

- When would we use screws instead of nails?
  - A set of circumstances
  - That make it impossible to use nails
  - While using screws is possible





```
wall := Tile.
method := ?.
weight := ?.
```

```
wall := Tile.
method := Glue.
weight := 7.
```

...

```
wall := Brick.
method := Glue.
weight := 2.
```

```
method := Glue.
```

Propagation

What are the consequences of set of circumstances?

Goal
Propagate

Try it: Same "Handyman in cDMN"

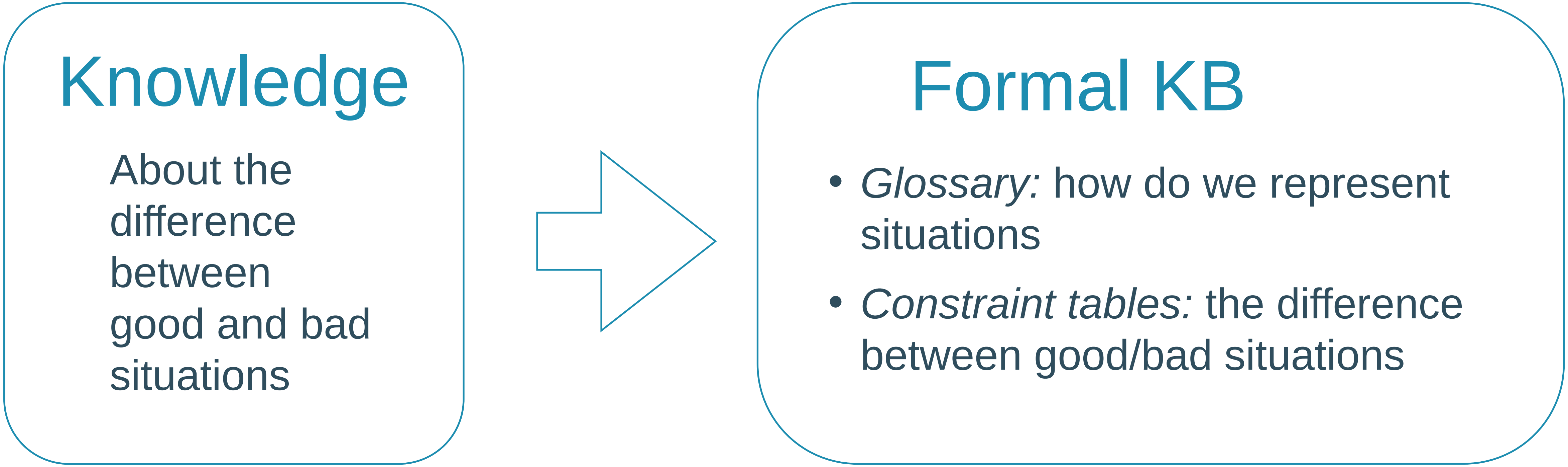
- What functionalities are there?
- Which inference tasks do you recognise?

Knowledge Base Paradigm

- GUI and all functionalities are derived from single KB
- If you own a diamond drill bit, then it *is* possible to make a hole in tiles

Try it: Updating with diamond drill bit

A deeper look:
At the cDMN language

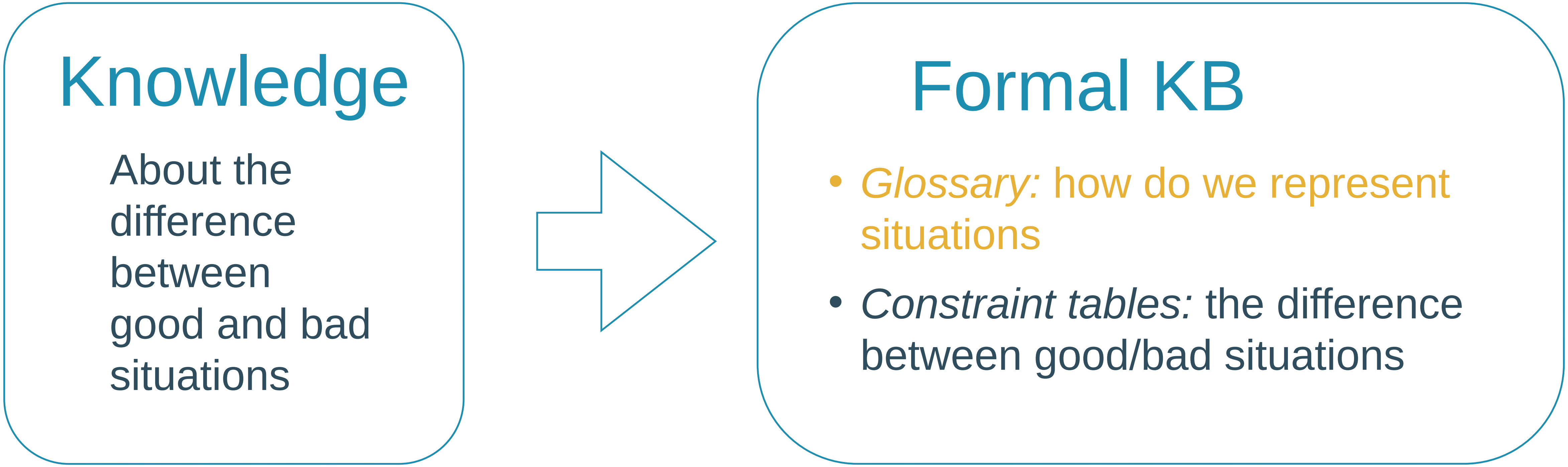


Knowledge

About the difference between good and bad situations

Formal KB

- *Glossary:* how do we represent situations
- *Constraint tables:* the difference between good/bad situations



Knowledge

About the difference between good and bad situations

Formal KB

- *Glossary:* how do we represent situations
- *Constraint tables:* the difference between good/bad situations

Glossary (~ RDBMS schema / class diagram)

- How do we represent situations:
  - Types
  - Constants
  - Functions
  - Relations
  - Booleans

Glossary (~ RDBMS schema / class diagram)

- Values belong to **types**
  - Built-in types:
    Int, Real, String
  - User-defined types derived from built-in types

- **Constant** is a specific value from a certain type

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain

Constant	
Name	**Data Type**
my bedroom wall	Wall
the object to hang	Decoration
the weight of the object	Weight

Symbol	Interpretation

Situation

Symbol	Interpretation
my bedroom wall	wood
the object to hang	mirror
the weight of the object	13.5

Situation

Symbol	Interpretation
my bedroom wall	brick
the object to hang	curtain
the weight of the object	1.2

…

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain

Constant	
Name	**Data Type**
my bedroom wall	Wall
the object to hang	Decoration
the weight of the object	Weight

Glossary (~ RDBMS schema / class diagram)

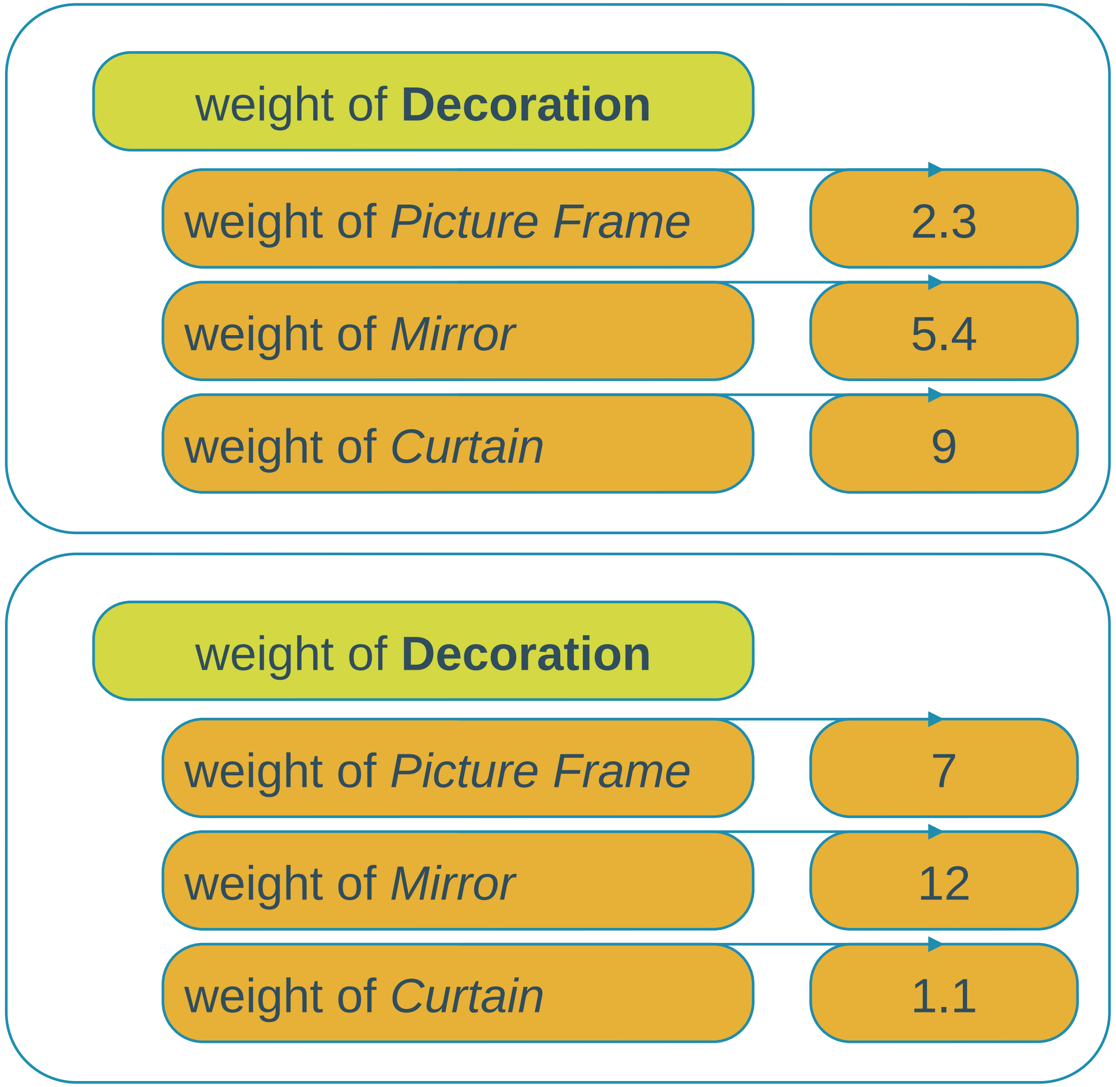
- Values belong to types
  - Built-in types: Int, Real, String, Datestring
  - User-defined types derived from built-in types

- **Function** maps tuples of values of a specific types to values of another type

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain

Function	
Name	**Data Type**
weight of **Decoration**	Weight

Whatever you want with **Type1** and **Type2**



weight of **Decoration**

weight of *Picture Frame* → 2.3

weight of *Mirror* → 5.4

weight of *Curtain* → 9

weight of **Decoration**

weight of *Picture Frame* → 7

weight of *Mirror* → 12

weight of *Curtain* → 1.1

…

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain

Function	
Name	**Data Type**
weight of **Decoration**	Weight

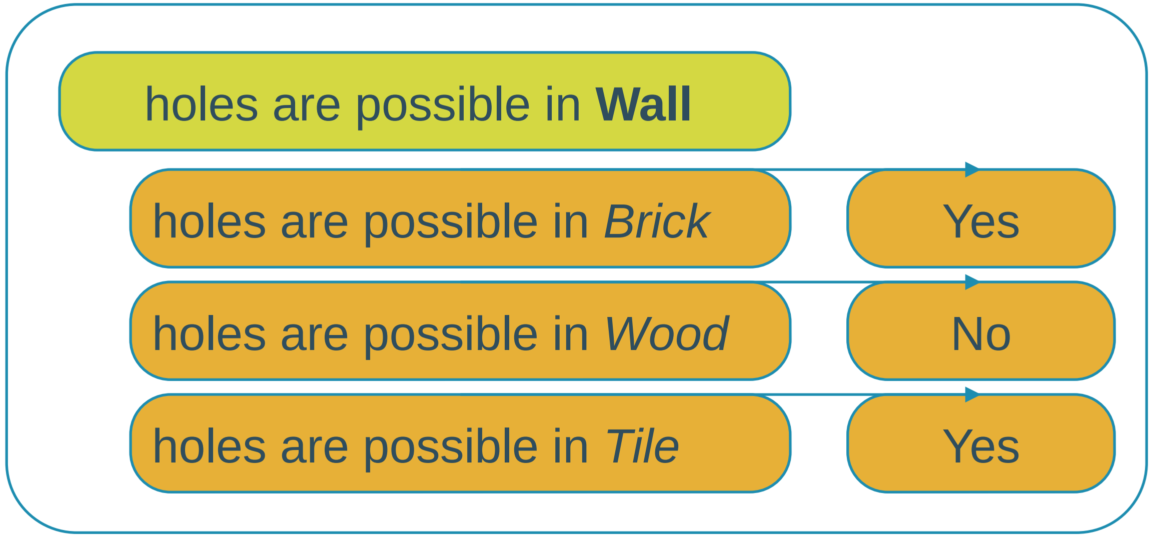
Glossary (~ RDBMS schema / class diagram)

- Values belong to types
  - Built-in types: Int, Real, String, Datestring
  - User-defined types derived from built-in types
- **Bool** is a constant with value Yes/No
- **Relation** is function that maps to Yes/No

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain

Relation
Name
holes are possible in **Wall**

Whatever you want with **Type1** and **Type2**

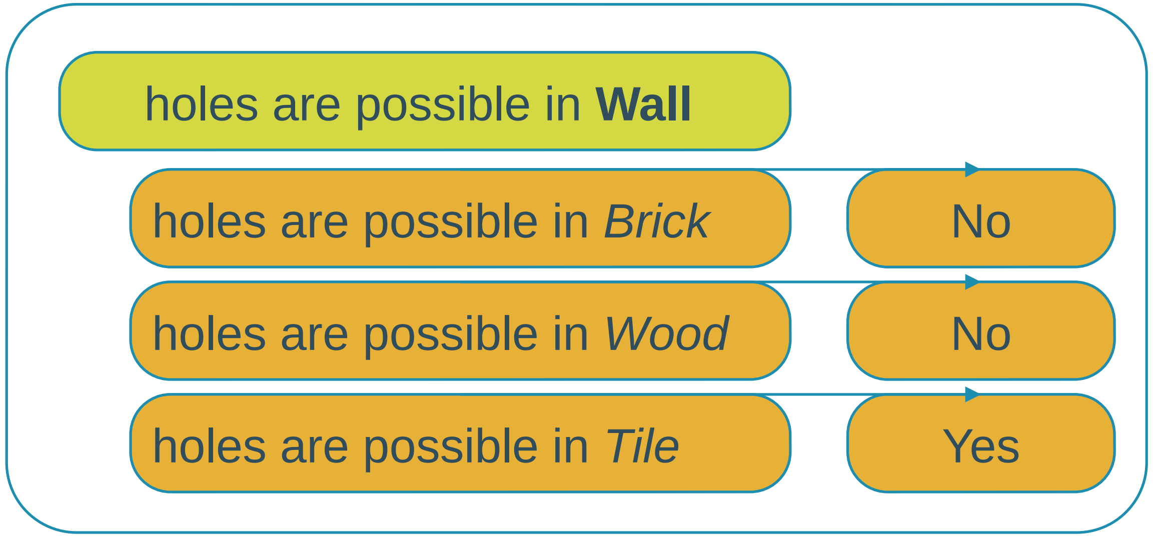


holes are possible in **Wall**

holes are possible in *Brick* → Yes

holes are possible in *Wood* → No

holes are possible in *Tile* → Yes



holes are possible in **Wall**

holes are possible in *Brick* → No

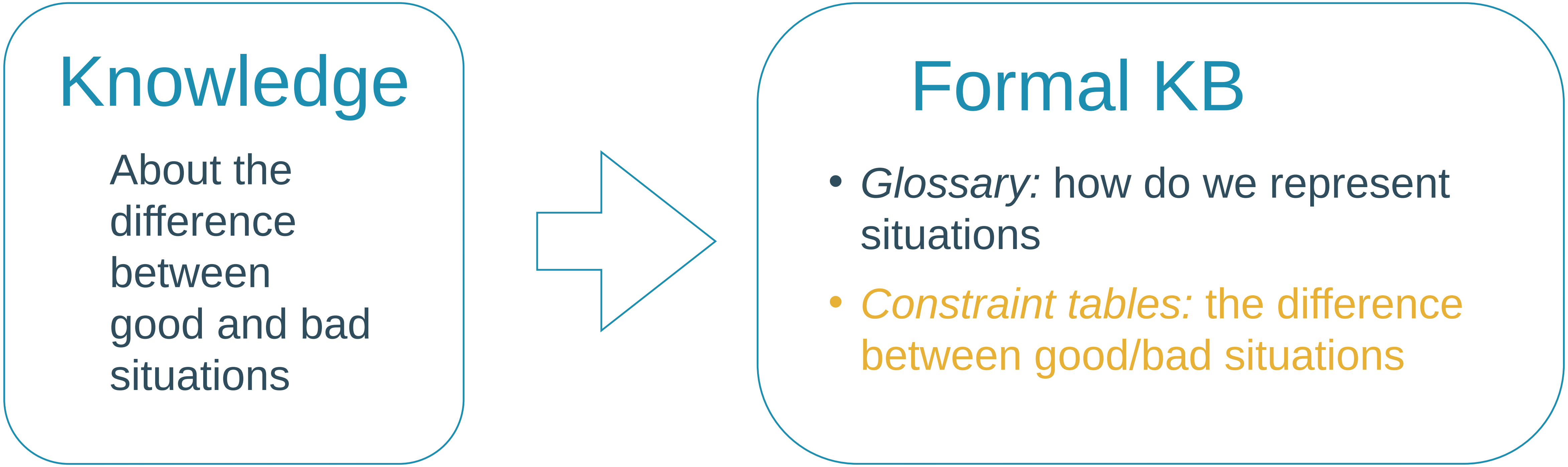
holes are possible in *Wood* → No

holes are possible in *Tile* → Yes

…

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain

Relation
Name
holes are possible in **Wall**



Knowledge

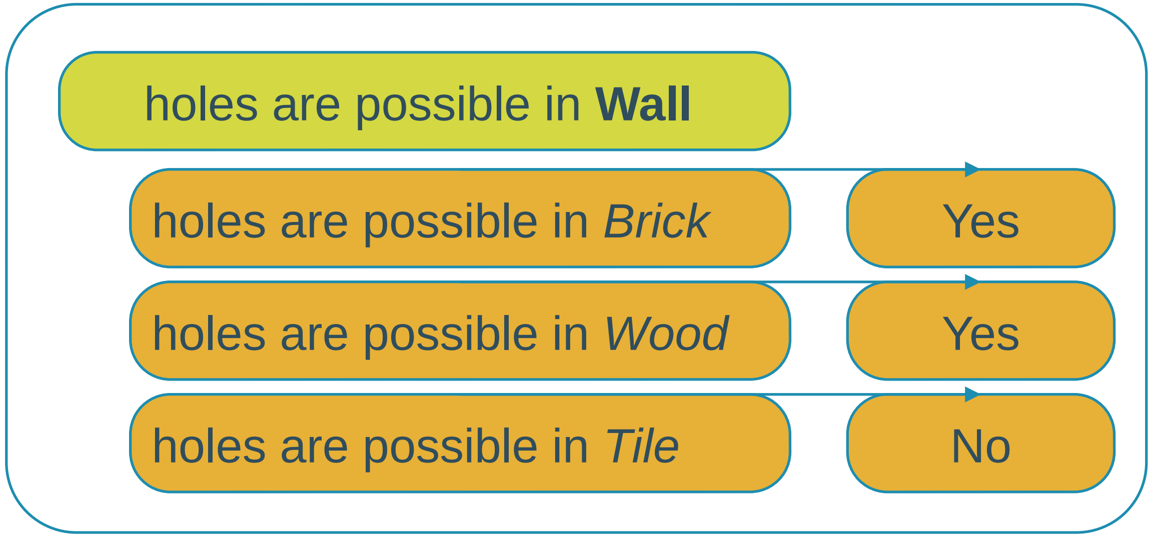
About the difference between good and bad situations

Formal KB

- *Glossary:* how do we represent situations
- *Constraint tables:* the difference between good/bad situations

Data tables

One correct interpretation for this relation



holes are possible in **Wall**

holes are possible in *Brick* Yes

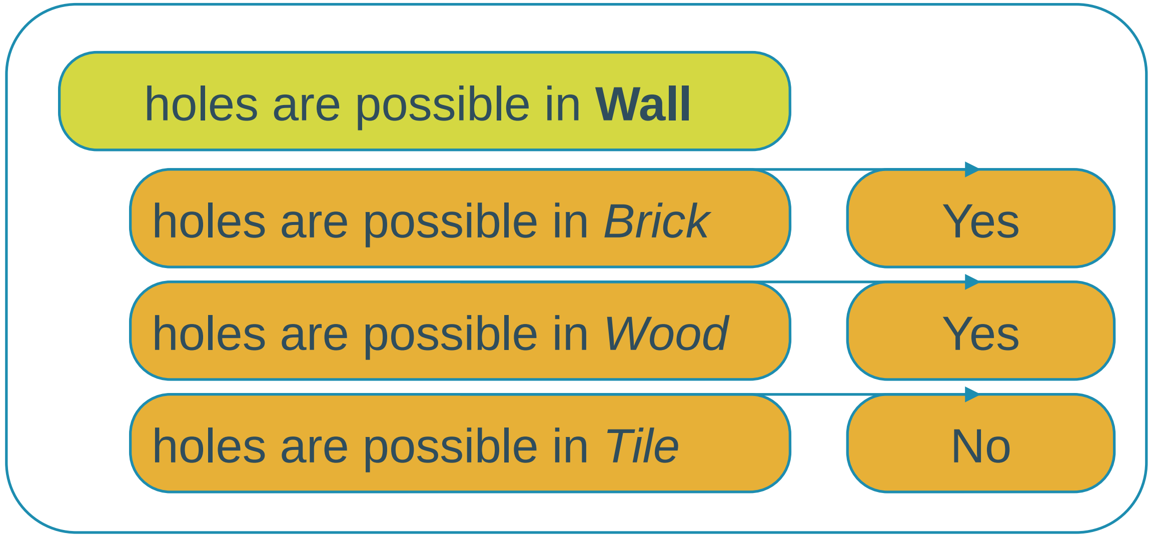
holes are possible in *Wood* Yes

holes are possible in *Tile* No

Relation
Name
holes are possible in **Wall**

Data tables

One correct interpretation for this relation



holes are possible in **Wall**

holes are possible in *Brick* → Yes

holes are possible in *Wood* → Yes

holes are possible in *Tile* → No

Relation
Name
holes are possible in **Wall**

D

Wall	holes are possible in **Wall**
brick	Yes
wood	Yes
tile	No

optional

Constraint tables

E*

chosen method	holes are possible in *my bedroom wall*
nail	Yes
screw	Yes

E*

chosen method	weight of *the object to hang*
nail	<= 30
screw	<= 40
glue	<= 15

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain
Method	String	nail, screw, glue

Constant	
Name	**Data Type**
my bedroom wall	Wall
the object to hang	Decoration
chosen method	Method

Relation
Name
holes are possible in **Wall**

Function	
Name	**Data Type**
weight of **Decoration**	Weight

Constraint tables

E*

chosen method	weight of *the object to hang*
nail	<= 30
screw	<= 40
glue	<= 15

=

E*

weight of *the object to hang*	chosen method
(15,30]	nail, screw
(30,40]	screw
<= 15	-

optional

E*

weight of *the object to hang*
<= 40

Type		
Name	**Data Type**	**Possible Value**
Wall	String	brick, wood, tile
Weight	Real	[0,50]
Decoration	String	picture frame, mirror, curtain
Method	String	nail, screw, glue

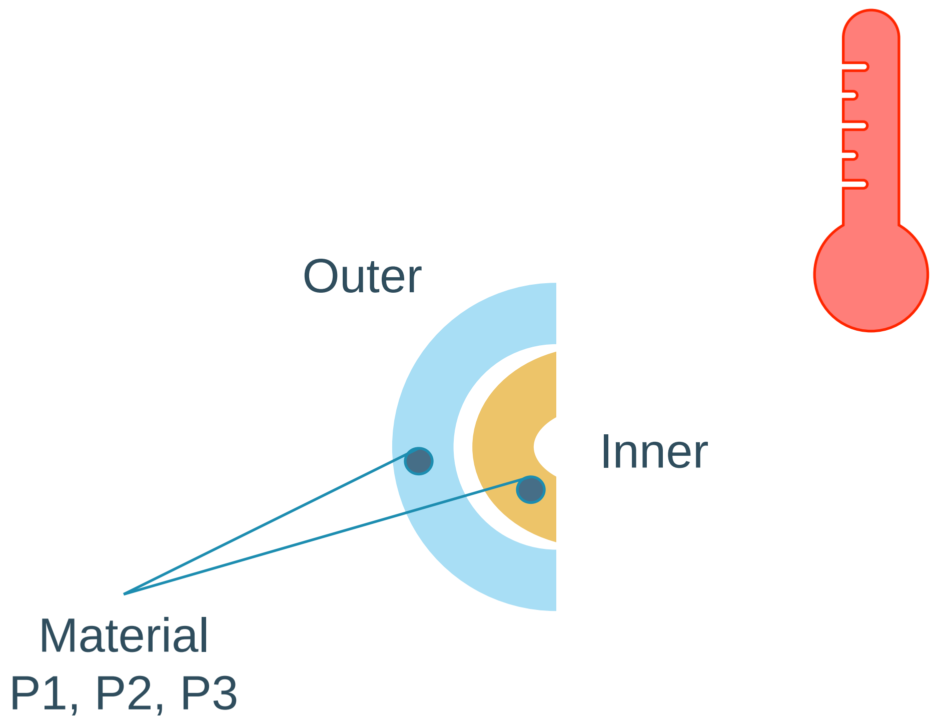
Constant	
Name	**Data Type**
my bedroom wall	Wall
the object to hang	Decoration
chosen method	Method

Relation
Name
holes are possible in **Wall**

Function	
Name	**Data Type**
weight of **Decoration**	Weight

Component design example

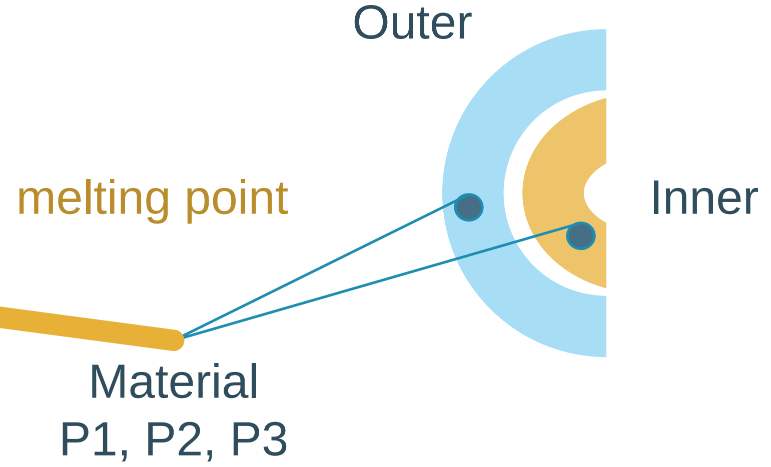
- For the Outer component
  - If its material is P1, then the maximum temperature <= 100
  - If its material is P2, then the maximum temperature <= 150
  - If its material is P3, then the maximum temperature <= 200
- (For the Inner component, same thing)
- Inner and Outer component materials should be compatible
  - P1 → P1,P2; P2 → P1,P2; P3 → P3

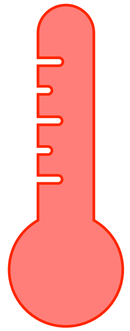


Try it: Component design -- basic model

Component design example

- For the Outer component
  - If its material is P1, then the maximum temperature <= 100
  - If its material is P2, then the maximum temperature <= 150
  - If its material is P3, then the maximum temperature <= 200
- (For the Inner component, same thing)
- Inner and Outer component materials should be compatible
  - P1 → P1,P2;   P2 → P1,P2;   P3 → P3



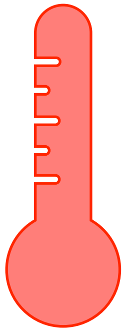


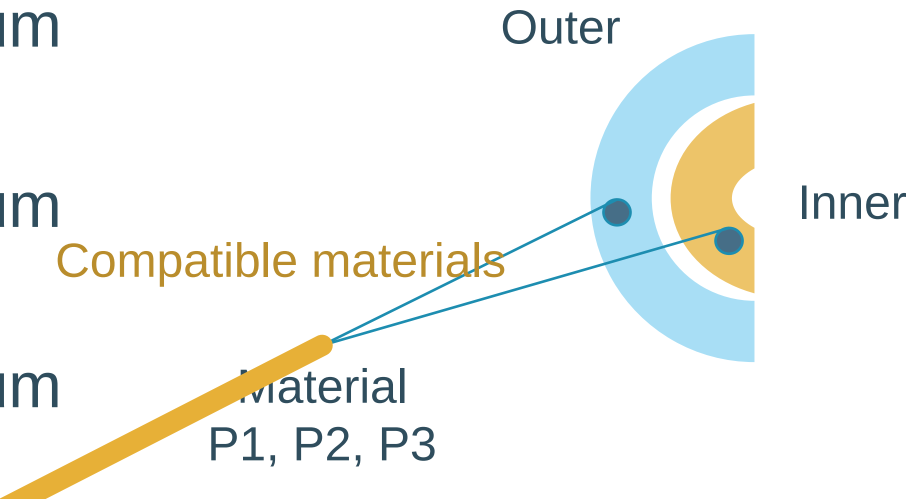
- Add to glossary
- Change constraint

Try it: Component design
-- with melting point model

Component design example

- For the Outer component
  - If its material is P1, then the maximum temperature <= 100
  - If its material is P2, then the maximum temperature <= 150
  - If its material is P3, then the maximum temperature <= 200
- (For the Inner component, same thing)
- Inner and Outer component materials should be compatible
  - P1 → P1,P2;   P2 → P1,P2;   P3 → P3





- *Add to glossary*
- *Define in data table*
- Change constraint

Demo

Try it

A realistic example

Team selection for R&D Project “Smart Compressors”

- Information about employees
  - Focus: Compressor, IoT and Prototype
  - Skills: Project Management, Functional analysis, Design
  - Experience: between 0 and 30 years
- Five employees:
  - *Edsger* knows **Compressor** and **IoT**. He has the "Design" skill. He has 12 years of experience.
  - Ada knows **Prototype** and **Compressor**. He has the "Design" and "Functional analysis" skills. She has 5 years of experience.
  - *Donald* knows **IoT**. He has the "Functional analysis" skill. He has 2 years of experience.
  - *Grace* knows **Compressor**, **IoT** and **Prototype**. She has the "Project Management" skill. She has 10 years of experience
  - *Brian* knows **IoT** and **Compressor**. He has the "Functional analysis" skill. He has1 year of experience.

Project team

- We need to select employees for a new project
- All selected employees should know Compressor
- All selected employees should have at least 2 years of experience
- One of the selected employees will be the Project Lead. This employee should have the "Project Management" skill and at least 5 years of experience.
- One employee should be the designer

Different inference tasks

- Who has to be part of the team?
- Who can't be on the team?
- Select a team
- What is the smallest / largest possible possible team?

Approach

- Only count the team size
- Add focus
- Add years of experience
- Add Project Lead
- Add skills
- Add Design constraint

Try it: Team selection

Conclusions and use cases

Seal design

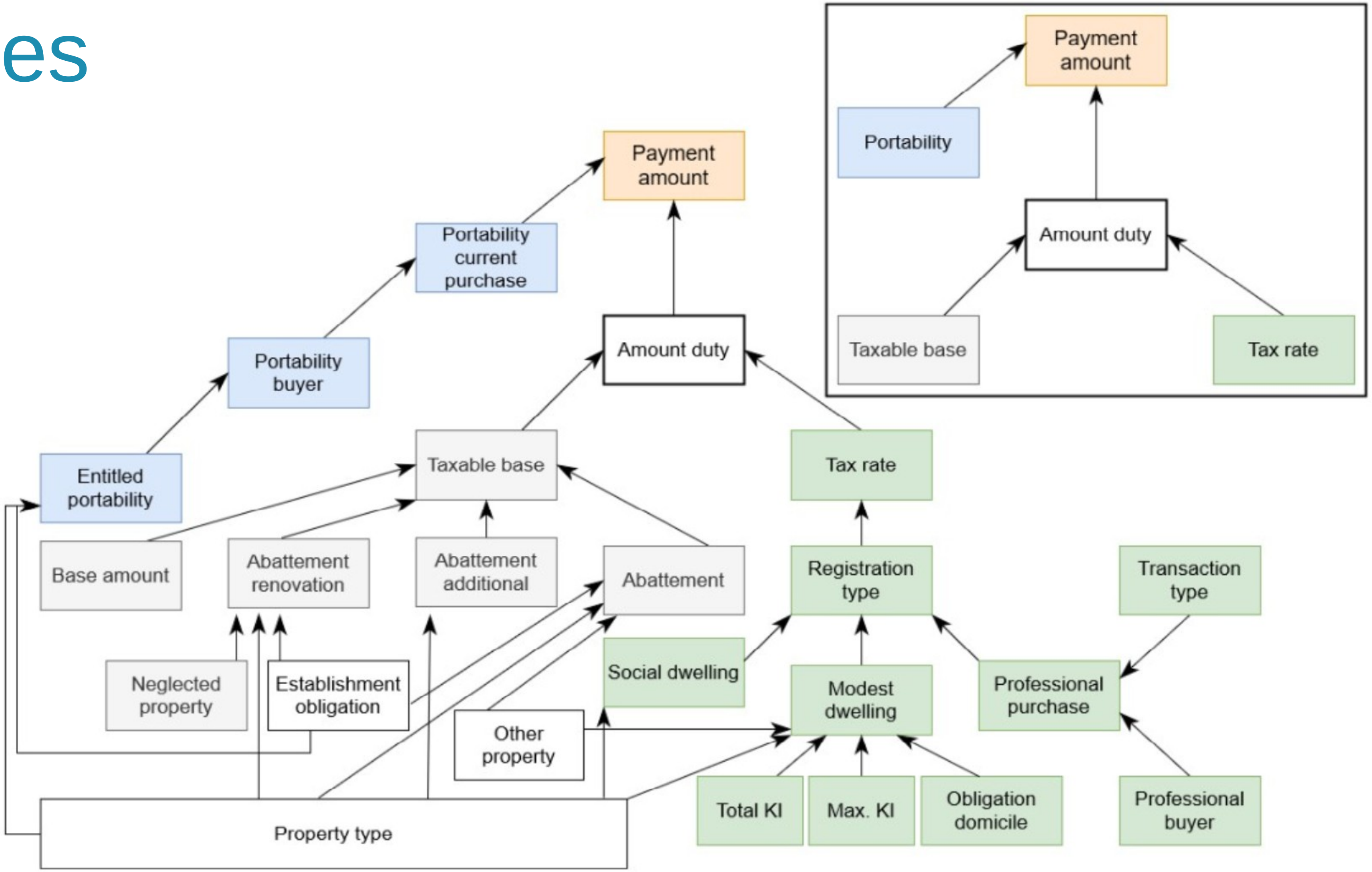
- Standard product range
  - System for sales staff using DMN
  - Currently in production
  - DMN approach is being applied elsewhere
- Design-to-order
  - System for design engineers using FO(.) and cDMN
  - Integration with Machine Learning from database
  - Positive reactions to prototype, but concerns about required effort
  - Continued investigations, somewhere on their roadmap

Adhesive Selector

- Currently in use (54 cDMN tables ~ 500 constraints)
- Benchmark: 3 hours → 3 minutes
- "Teaching" tool for juniors
- Seniors can discover "new" adhesives

Software company for notaries: Registration duties

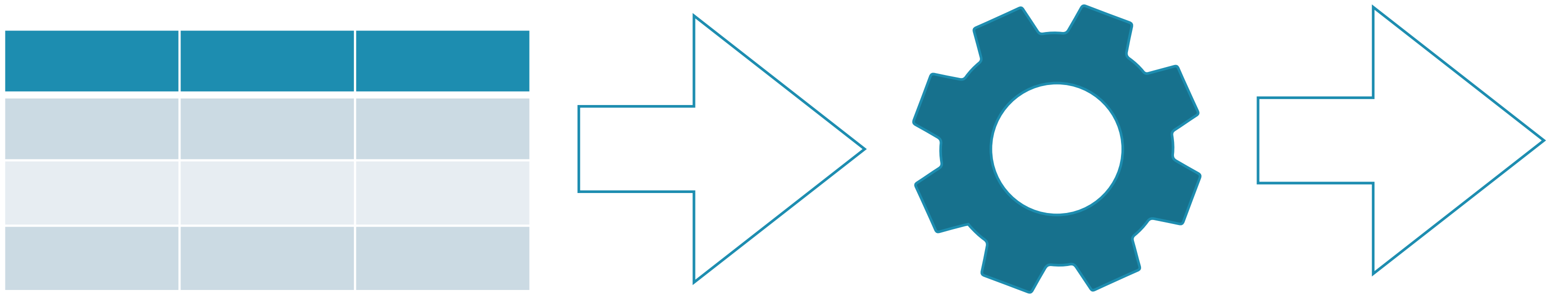
- *Flexible support* during interview: non-intrusive
- Represented the relevant laws in DMND and cDMN
- Big change in 2018
    - 42 articles:
      - 4, + 5, ~ 9
    - 3 hours of effort needed



Generic tools / methods

Joint interactive modeling

Knowledge base paradigm

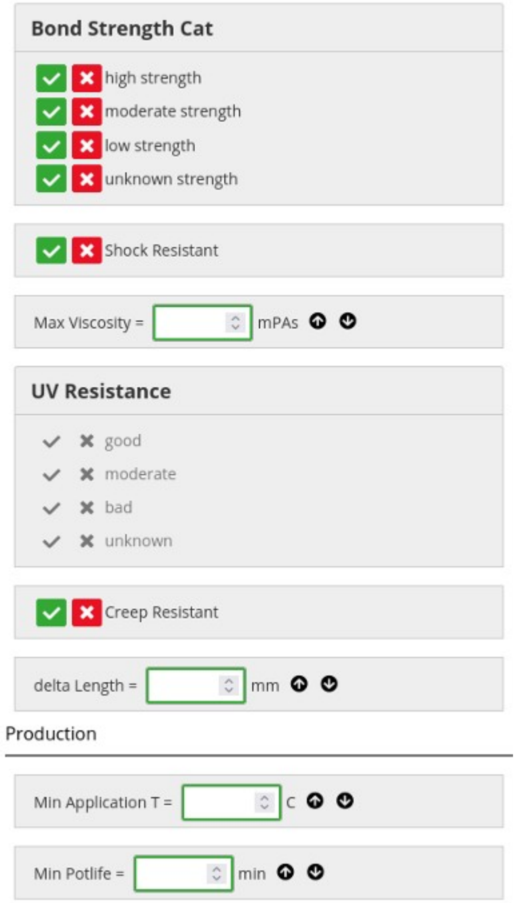


cDMN

www.cdmn.be

IDP-Z3

www.idp-z3.be



Bond Strength Cat

✓ ✗ high strength
✓ ✗ moderate strength
✓ ✗ low strength
✓ ✗ unknown strength

✓ ✗ Shock Resistant

Max Viscosity = ☐ mPAs

UV Resistance

✓ ✗ good
✓ ✗ moderate
✓ ✗ bad
✓ ✗ unknown

✓ ✗ Creep Resistant

delta Length = ☐ mm

Production

Min Application T = ☐ C

Min Potlife = ☐ min

Interactive consultant