Why?

It's easy, you just ***write down*** your requirements

Well, actually it takes a ***bit of work*** to make them **precise and verifiable**

We don't bother with requirements, we ***just capture*** Use Case **scenarios**

We haven't found use cases adequate. People ***tell us solutions***, so we work back towards people's **real goals**.

Perhaps this is something that ***looks simple, but isn't easy***

Model-based Requirements Discovery

- Scenarios
- Context
- Goals
- Stakeholders
- Trade-offs
- Rationale
- Metamodel
- Tool Support
- Reflections

3

A Small Example

- **A Vending Machine**
  - such as this one …
  - but making *no assumptions* about its design or user interface …

4

Why Not Just Scenarios*?

Let's try "**Buy Lemonade**":

- Traveller wants refreshment in railway station.
- Traveller sees Vending Machine.
- VM displays list of refreshments on sale.
- Traveller/Customer presses button (#5) for lemonade.
- VM displays price (Đ20) of lemonade.
- Customer inserts coins in slot to the displayed value.
- VM checks and counts the coins.
- VM dispenses the goods, a can of lemonade.

OK ? Requirements Clear Now ?

* UML Use Case is a way of organising a set of Scenarios along with its Goal, trigger, actors, and guarantees.

Is this a pure statement of need?

"Buy Lemonade":

- Traveller wants refreshment in railway station.
- Traveller sees Vending Machine.
- VM displays list of refreshments on sale.
- Traveller/Customer presses button (#5) for lemonade.
- VM displays price (Đ20) of lemonade.
- Customer inserts coins in slot to the displayed value.
- VM checks and counts the coins.
- VM dispenses the goods, a can of lemonade.

Why not show choices directly through glass ?

Why not insert coins first, display credit ?

Why not take Credit/Debit cards ?

Why not work on prepaid travel card ? *

Why not respond to spoken commands ?

Why not let customer wave RFID card and just open any door to take / pay for goods ?

….. etc …..

* Like Transport for London's "Oyster" card

What is the Scenario Doing?

"Buy Lemonade":
- Traveller wants refreshment in railway station.
- Traveller sees Vending Machine.
- VM displays list of refreshments on sale.
- Traveller/Customer presses button (#5) for lemonade.
- VM displays price (Đ20) of lemonade.
- Customer inserts coins in slot to the displayed value.
- VM checks and counts the coins.
- VM dispenses the goods, a can of lemonade.

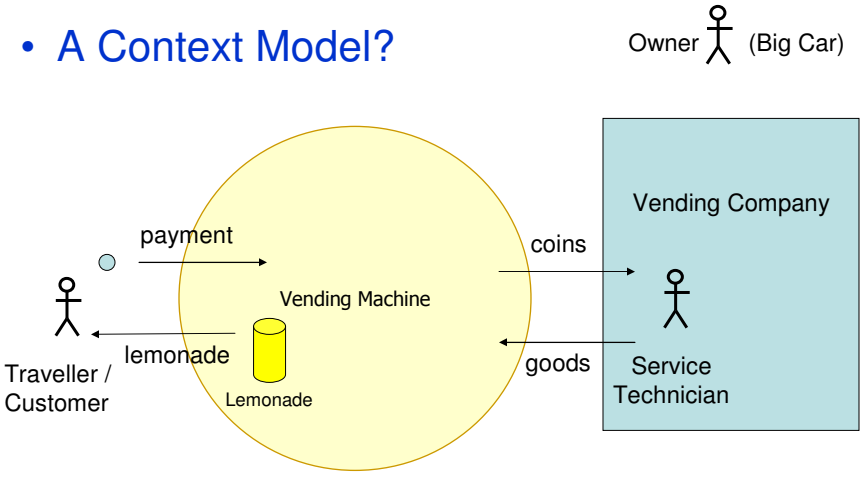
- Scenario *faithfully records* **OUTPUT** of a specification/design decision-making process
- In response to *[unstated?] needs* (traveller, vendor)

∴ Scenarios are a ***LATE-STAGE*** requirements technique.

7

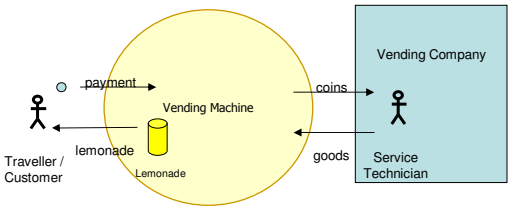
What comes before Scenarios?

- A Context Model?



8

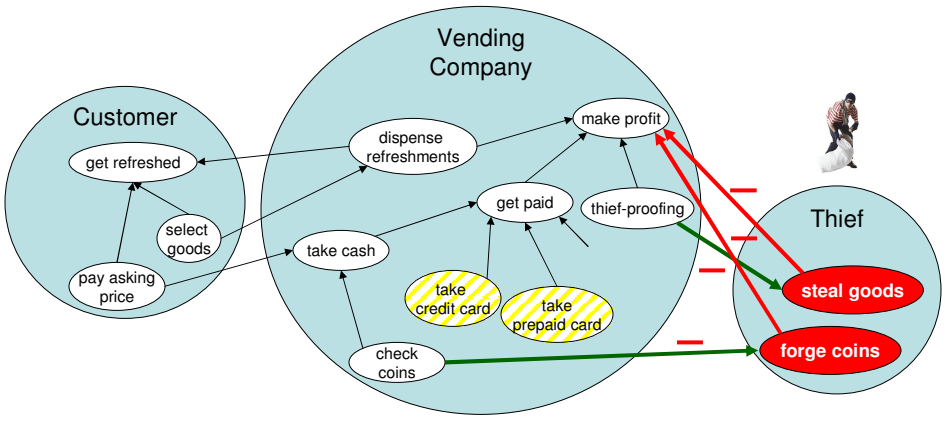
What is Context Model doing?



- Quite helpful in showing *future situation*
- Still ***many decisions*** in here, too
  - e.g. vending is automated, coins are used
- At least no "scenario" sequence of steps is imposed
  - could be view-then-pay or *vice versa*

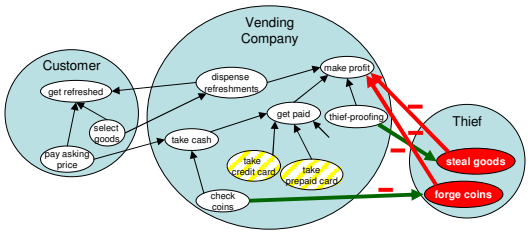
9

Goals behind the decisions?



10

What are the Goals saying?

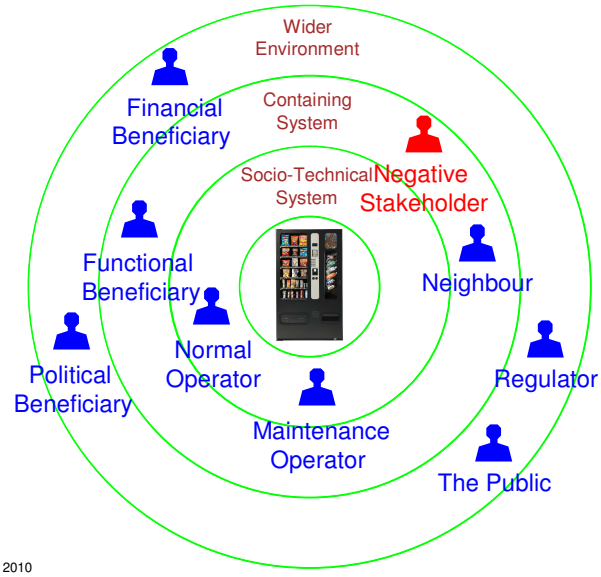


- Very helpful in showing who wants what
- Can explain *threats/obstacles* (e.g. theft)
- Can show *mitigations*
- Can show *alternatives* (candidate reqts, options)

- … but we need *Stakeholder Analysis* first!

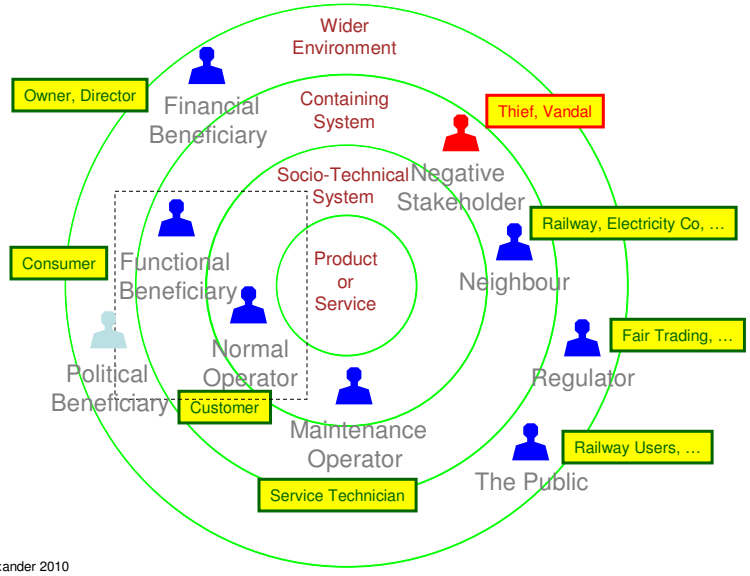
11

Stakeholder 'Onion' Model *Slots*

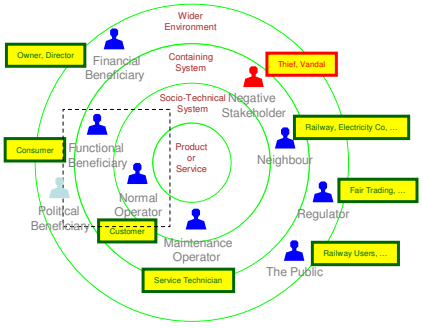


12

Stakeholder 'Onion' Model *Roles*



Stakeholders First, then?



- A big improvement
  - identified several roles *not previously considered*
  - will discover numerous missed requirements

- But even here, *assumptions* about design
  - it's a thing that needs servicing, could be vandalised
  - (are we sure it couldn't just be a human with a tray?)

Optioneering, Trade-offs

Option	+	-
Select then pay (1 item)	Simple to build, easy to operate	Only sells 1 item Code nos can cause errors
Pay then select	Encourages multiple sales	Slower to use? More complex, costly to build
Give Change	Fairer, more flexible for customer	More complex, more costly; bigger takings for thief

15

Trade-offs

Based on *Goals*

Options \ Criteria	C_1 C_2 C_3 C_4 C_5 ...
O_1 O_2 O_3 ...	Benefits, Costs

For Design

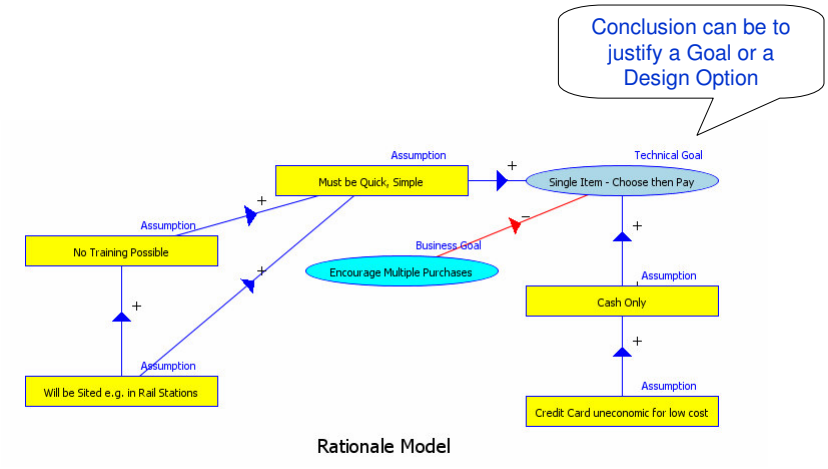
Aren't these also *desired qualities & results* (meta-requirements?)

Always ***multi-dimensional***

Can use multi-D tools like Principal Components Analysis (PCA)

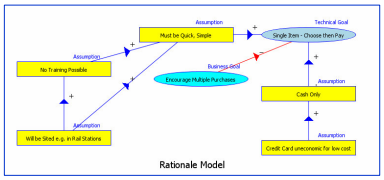
16

A Trade-off output: Rationale



17

Rationale for Rationale



- Demonstrates a proper process of evaluation
- Reduces risk
- Prevents rework to rediscover thinking behind requirements & design
- Enables reuse for variants, product lines
- Explains key decisions on project to new joiners

18

What is a Trade-off Doing?

- Inputs:
  - Goals
  - Stakeholder Priorities *(yet another model…)*
- Outputs:
  - The chosen design option(s), architecture
  - Measurements *(yes, more modelling)*
  - Rationale for decisions on options, requirements
  - Requirements
  - Project Priorities

We don't even know which goals turn into requirements until we've done the trade-offs!

Goals, Options, Requirements

- Goals:
  - things wanted by a stakeholder, candidate requirements
  - not necessarily realistic, affordable, verifiable
  - may be in conflict with other goals
  - may only be possible with certain of the Options
- Options:
  - alternative solution approaches at any level
    - *rival software features competing for development time/budget*
    - *rival hardware features competing for power, mass, …*
    - *alternative software algorithms for given purpose*
    - *complete rival system architectures*
- Trade-off criteria, based on goals:
  - dimensions on which to measure and compare options
- Requirements, after trade-offs:
  - agreed, measurable, verifiable, traded-off/prioritised goals

Goals, Options, Requirements

- Goals:
  - (traveller): *be refreshed, quickly, conveniently, cheaply*
  - (company): *make money (by selling)*
- Options (assuming vending machine*):
  - cash only, coins or notes, with or without giving change
  - credit card, debit card
  - prepaid card
- Trade-off criteria, based on goals:
  - benefit/cost of operation
  - benefit/cost of construction
  - attractiveness to customers
  - (reduction in) risk of theft, fraud
- Requirements, after trade-offs:
  - The VM shall accept payment in Đ coins.
  - The VM shall give change in Đ coins.

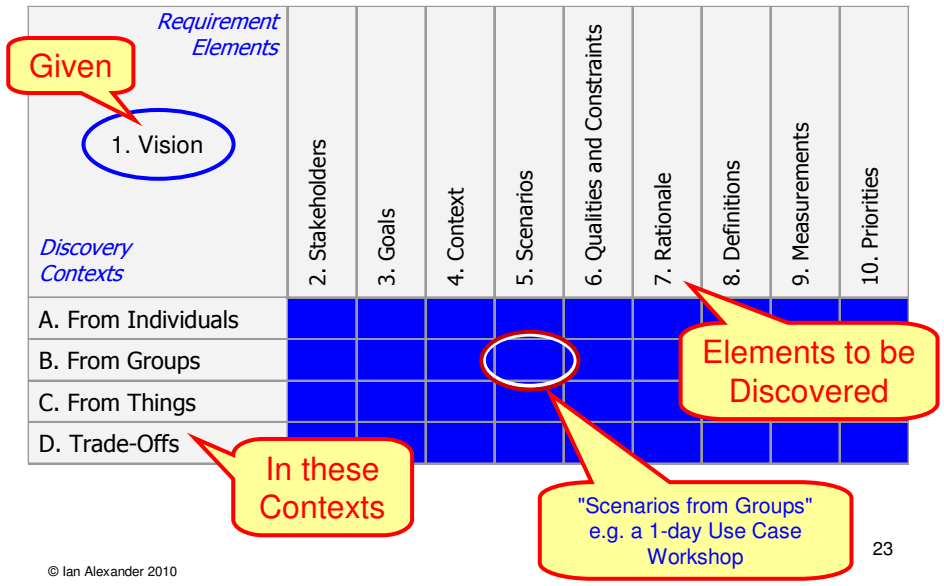
Could these also be goals? Yes, as goals are broken down, they naturally get into more solution detail

Goals can be real and central but useless as criteria. If all options refresh the traveller *and* permit payment*, these criteria do nothing to help find winner!*

* at the system level, if VM is not assumed, then options include VM on platform, shop on platform, street-seller (with tray/trolley) on platform, trolley on train, restaurant on train, …

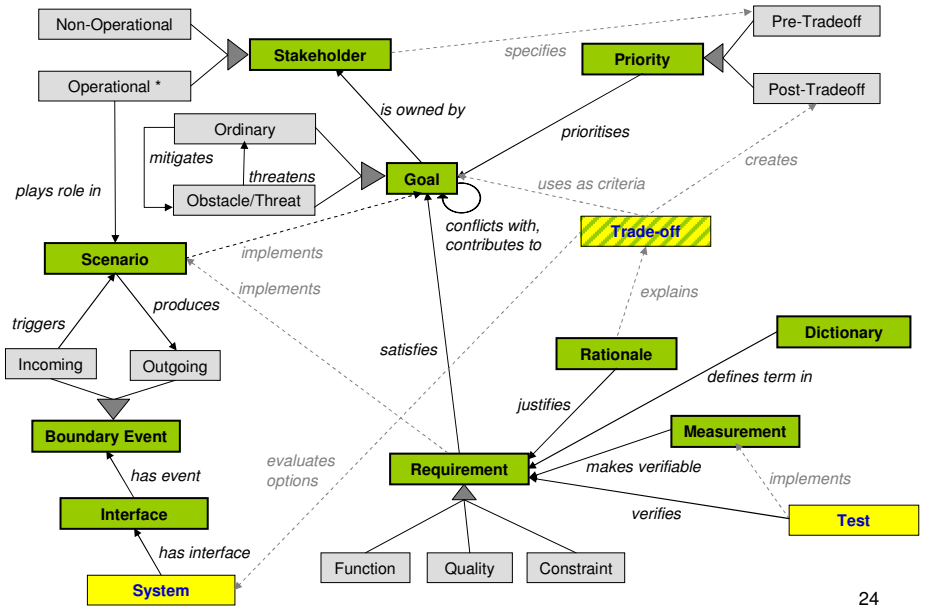
21

MBRD Matrix



23

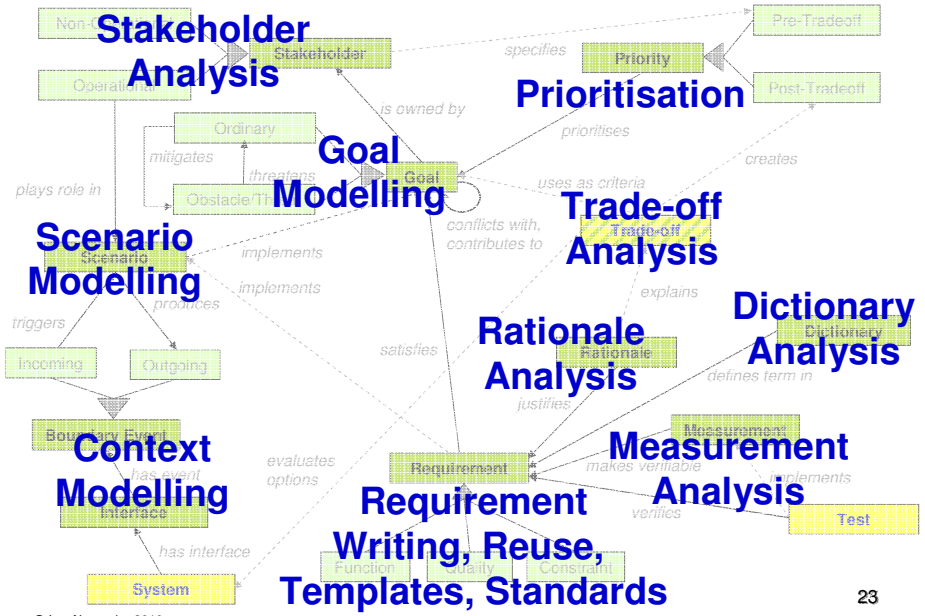
Metamodel for MBRD



* Operational = involved in ops, but not necessarily product-touching "actors"

24

Activities for MBRD



23

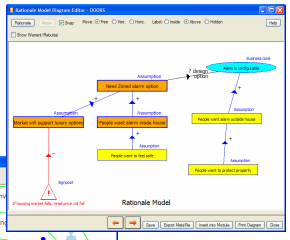
Model-Based Requirements Validation

- Check that each:
  - Goal *is owned by* a Stakeholder
  - Operational Stakeholder *plays a role in* a Scenario
  - Goal *is prioritised by* a Priority
  - High Priority Goal *is used as a criterion in* a Trade-off
  - Goal Conflict *is resolved by* a Trade-off
  - Obstacle, Threat *is mitigated by* a Goal
  - Goal *is satisfied by* a Requirement
  - Requirement *is made verifiable by* a Measurement
  - Trade-off *is explained by* a Rationale
  - <Term>* *in* Requirement *is defined by* Dictionary

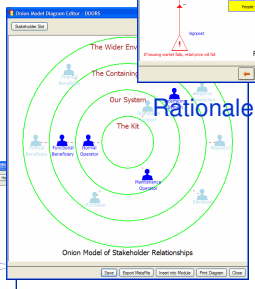
* <Term> can be any State, Goal, Operational Role, Measurement

Tool Support

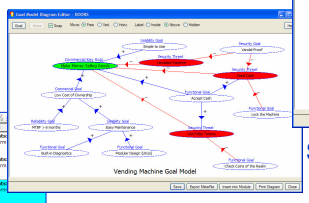
- Existing free add-on toolkit for DOORS
  - *Full traceability to individual diagram objects*
  - *Full industrial SE eg history, baselines*
- PEN AND FLIPCHART often ideal for requirements discovery
- Any graphics editor, if no traces needed



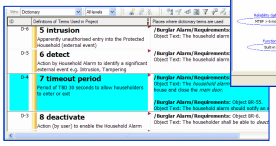
Rationale Model Editor



Stakeholder 'Onion' Model Editor



Goal Model Editor



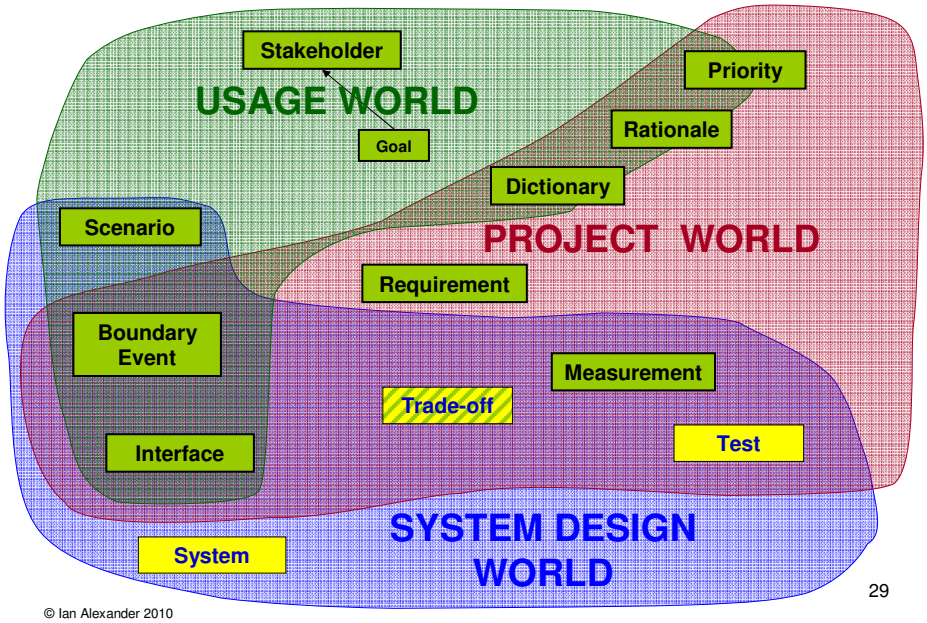
Dictionary Builder & Linker

Why MBRD?

- Ensures full value to
  - Stakeholders, Goals, Conflicts, Obstacles, Scenarios, Rationale, … … …
- Ensures all parts of the ***problem*** understood
- Works out how they fit together to make a larger whole

28

Requirements Model, System Model?



29

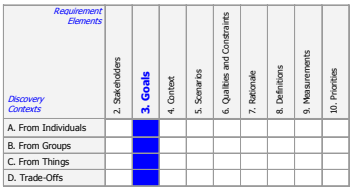
Isn't MBRD just a part of normal MBSE?

- **YES, BUT…**
  - Only if you're doing it right
    - *do you model stakeholders, goals, rationale already?*
  - Many practices long familiar
    - *e.g. context, operational scenarios*
  - Often poorly traced
    - *especially if the goals and decisions that should be traced to are not recorded*
  - Stakeholders & their Goals often undervalued
    - *missed requirements*
  - MBRD also needed in *Software* Engineering

30

What about GORE Research?

- **Goal-Oriented RE** is an approach that emphasises Goals as key to requirements

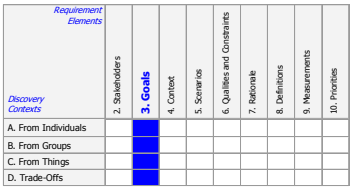


- 2 main schools of GORE research:
  - **i*** from Toronto (Eric Yu)
  - **KAOS** from Louvain/Leeuwen (Axel van Lamsweerde)
- Both ~20 years old

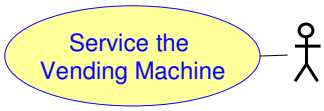
31

What about Goals in Industry?

- In industry, explicit modelling of Goals is rare
- More attention to goals would reduce gold-plating



- Not clear that i* or KAOS will be adopted widely
- UML's Use Case bubble is a simple starting-point
  - a Functional Goal *("to do something")*
  - implemented by a set of scenarios
  - so why not abuse the notation for non-functional goals as well?!



35

Competing Approaches?

Approach (Element)	How This Describes Need	Schools of Thought that Favour This	Disadvantages of Using This Alone
Stakeholder Analysis	Identifies political, economic, social, cultural forces on design	*Soft Systems Methodology*	Not precise, verifiable; unsuitable for contracts
Goal Modelling	Says what stakeholders want	*KAOS* method; i** goal modeling language*	Poor for timing (scenarios); danger of diving into design
Event-Driven Analysis	Identifies events at interfaces, and says how to handle them	*Event-Driven methods*	Assumes context is well-defined: ignores soft systems issues, conflicting goals
Scenario Analysis	Says how design will deliver results to human operators	*Cockburn-style Use Cases*; *Agile (User Stories)*	Over-emphasis on product behaviour
Reuse – Standards & Templates	Defines generally needed qualities, constraints, and procedures for achieving them	Standardisation, Regulation, Quality Assurance	Functions and innovative aspects not covered
Rationale Modelling	Explains why design is needed, or what is needed to make it safe	*Compendium*; *GSN, CAE*	lacks timing (scenarios); danger of arguing for a chosen solution
Data Modelling	Defines rules to ensure business works properly	*UML* class modelling	Lack of end-to-end behaviour (scenarios), context, purpose
Measurement	Shows exactly what results design must provide	Traditional; list of *"System shall…"* reqts	Requirement wording has to carry whole burden
Priorities	Shows which design features are needed most	*Business Case; Benefit/Cost Analysis; Value Engineering*	Suggests requirements are independent, compared by cost

36

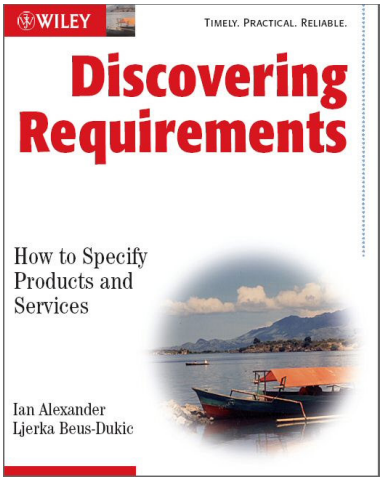
Complementary Approaches

Requirement Elements / *Discovery Contexts*	Stakeholders	Goals	Context, Interfaces, Scope	Scenarios	Qualities and Constraints	Rationale	Definitions	Measurements	Priorities
From Individuals									
From Groups									
From Things									
Trade-Offs									

Your Project May Need Any or All of These

37

Thank you for Listening



38