

Domain Mapping & Modelling

Bill Olivier

Development Director, JISC

- You may have heard of
 - “Problem Domains” &
 - “Solution Domains”
- But even the Problem Domain is often technical
- What is meant here is much broader:
- a recognisable area of work or activity

- We like to think of Universities as autonomous
- For our purposes, we can think of the University as the top level Domain
- But there are Funding & Research Councils...
- So for some purposes, universities might be seen as part of a higher domain, the University System
- Or seen as linked to the Gov. Funding Domain

The e-Framework Consortium sees a University as composed of five sub-domains:

1. Learning and Teaching
2. Research
3. Libraries
4. Administration
5. Information Services

Each of these in turn break down into further sub-domains, e.g.

Learning & Teaching:

- Course management

- Content preparation and management

- Student enrolment

- Course delivery (lectures, seminars, projects, etc.)

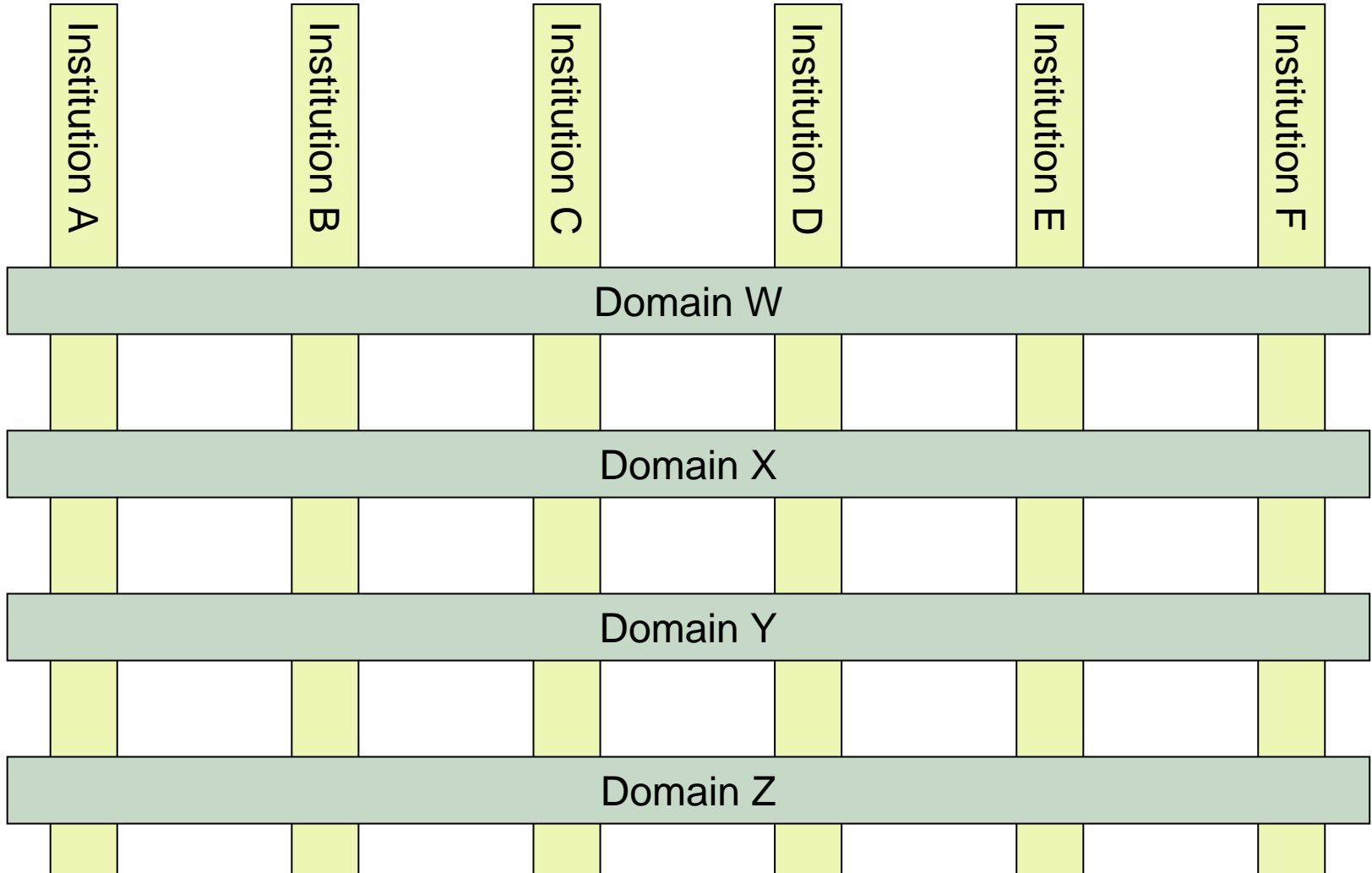
- Assessment

- etc.

Typically a domain has:

- Practitioners
- Specific functions and expertise
- Specialised vocabulary
- With associated inter-related concepts
- Tend to form (professional) communities
- Exchange ideas, share problems and solutions

Domains Cut Across Institutions



Service Usage
Models

- Service Usage Models, derived from Domain, Information, and Process Models

Services

- Services: definitions and descriptions

Guides
Methodologies
and Analysis

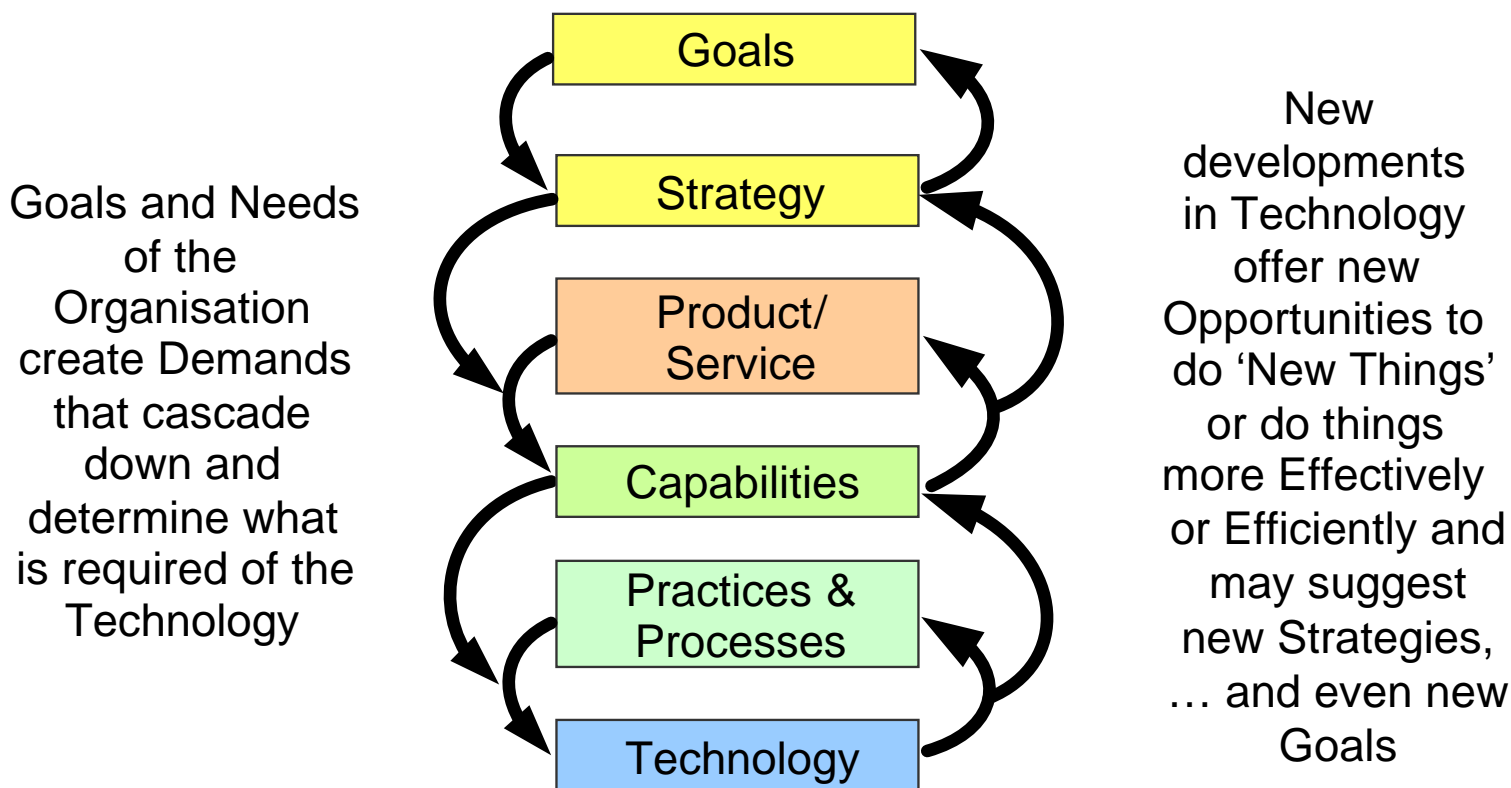
- Guides, Methodologies, Analyses

Why is it so difficult to specify requirements?

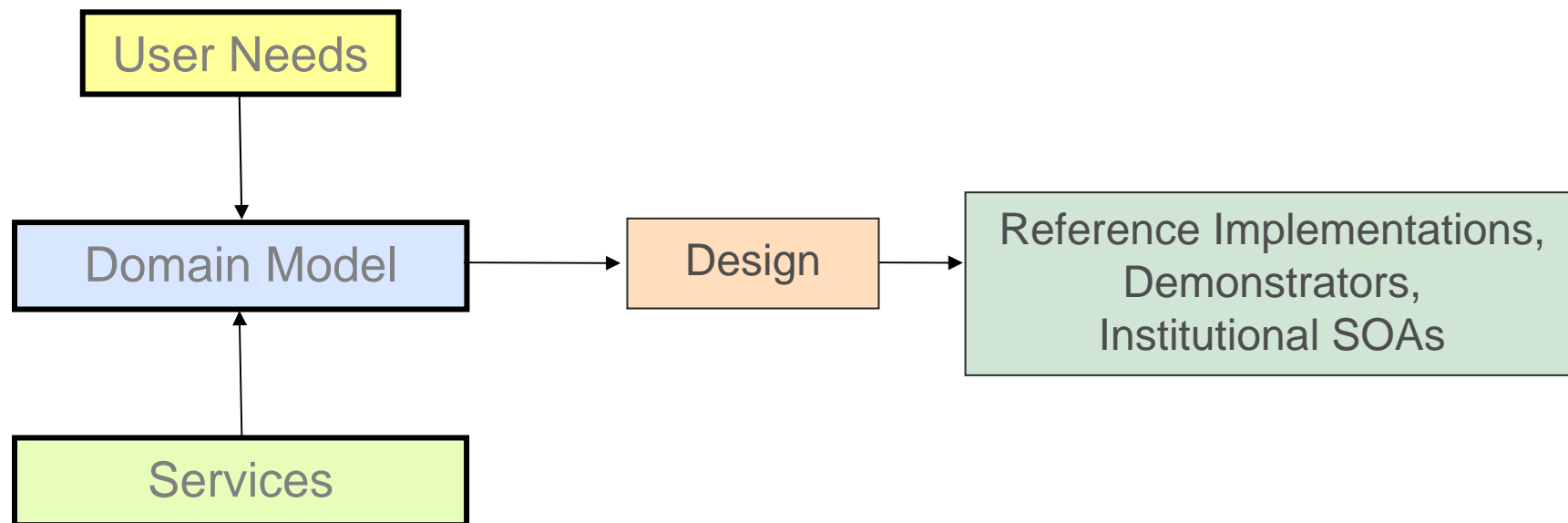
- Introducing new software changes practices & processes
- It's hard to co-design software and new ways of working
- Users don't know what is feasible
- Technologists don't understand the practices
- It's even hard to identify where to apply development
- A dialogue is needed between users & technologists...

How should we balance/integrate:

1. Technology Push & Demand Pull?
2. Blue-skies/Disruptive & Incremental/Sustaining Innovation?



The Role of Domain Models



A Domain Model shows how the needs of its practitioners can be met by a set of Services.

A Domain Model may be used to help build Reference Implementations, Demonstrators and/or Institutional Service Architectures.

Domain Models form a Bridge between Users' Needs & Services

- A “Domain Model” **bridges** :
 - the World of Users and their Work
 - the underlying Technical Services and their associated Interface Specifications
- We therefore need to make clear:
 - The Human Context:
 - The Machine Context

- The recently completed ‘Reference Model’ projects identified a number of parts:
 - A human domain map / model / ontology
 - The stakeholders and roles and their goals
 - The functions and tasks
 - The entities used and their relationships
 - The workflows of practices and processes
 - As is (then identify problems and opportunities)
 - To be (innovations &/or improvements)

They also have a technical part

- The user interface, tool or application layer
- The services that lie behind
- Between the UI & Service layers: A Co-ordination Layer
 - An “orchestration” of services
 - Several services composed behind one front-facing coordination service
 - A service “choreography”
 - Where multiple services work together without a central organiser, as when working across departmental or organisational boundaries and systems

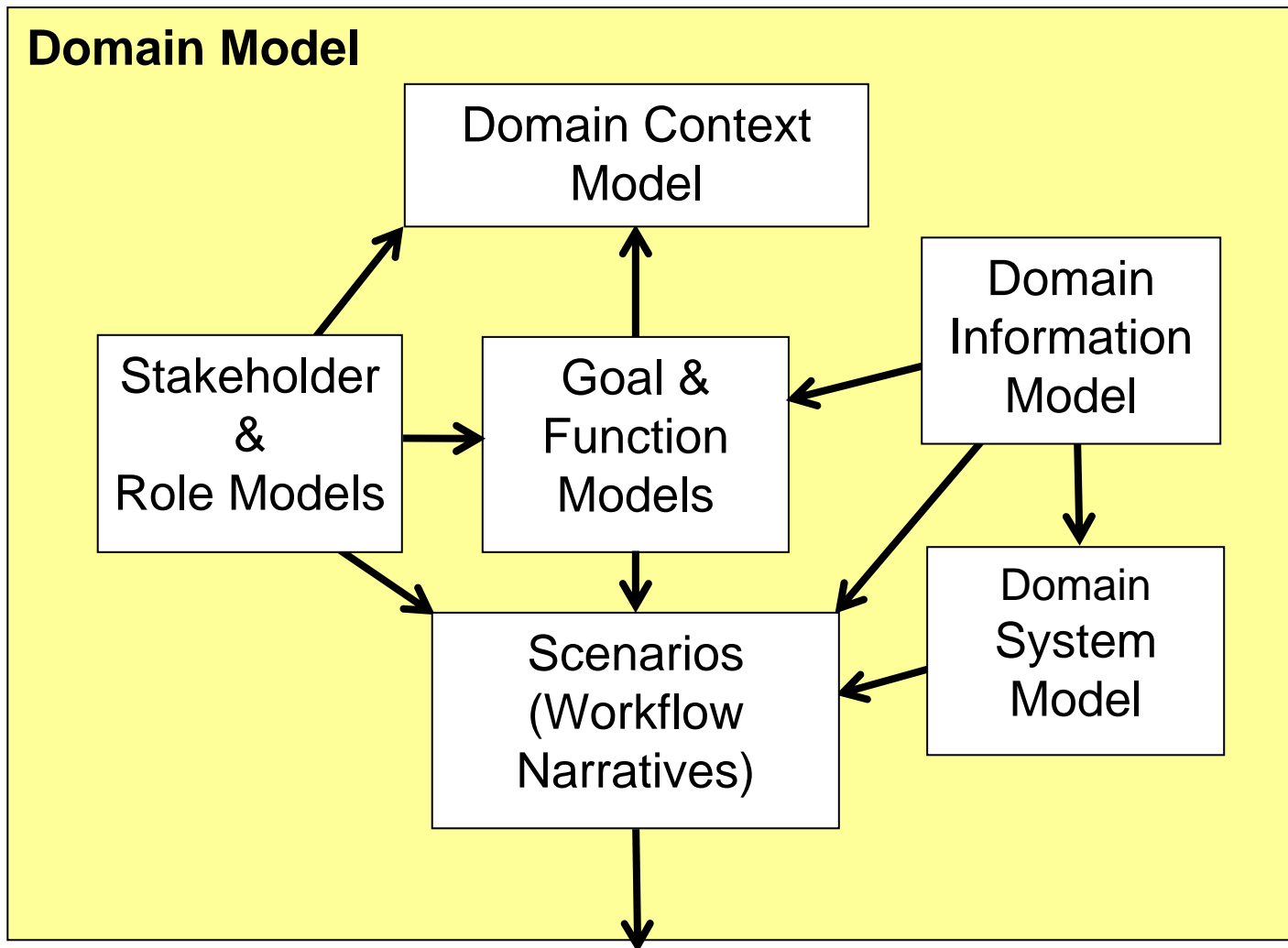
- But how are Needs to be identified?
 - not just for an Institution,
 - but across a Sector?

- Need to work with
 - Users and Practitioners
 - Domain Experts and broader Stakeholders

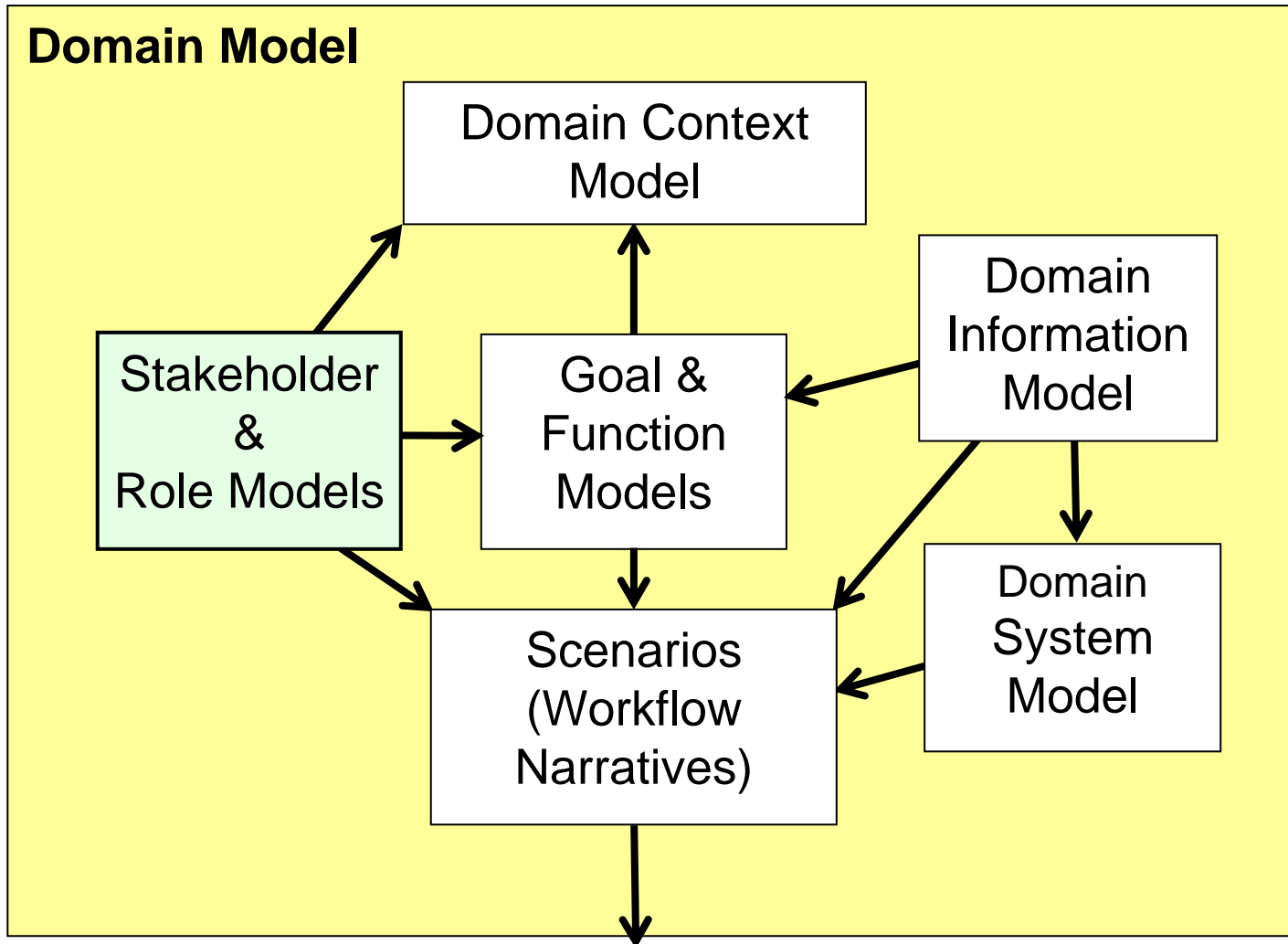
- To Map and Model their Domain, so as to:
 - Reflect on current Practices and Processes
 - Identify Problem Areas & New Opportunities
 - Set out what is common across multiple applications, as a basis for identifying services

- Stakeholders & Roles Who?
- Aims & Goals Why?
- Information Models / Ontologies What?
- Functions / High Level Tasks What / How?
- Scenarios How?
Where?
When?
- Practices & Process Models How?
- System Model (Software mapped to Hardware) Where?

Elements of a Domain Map or Model



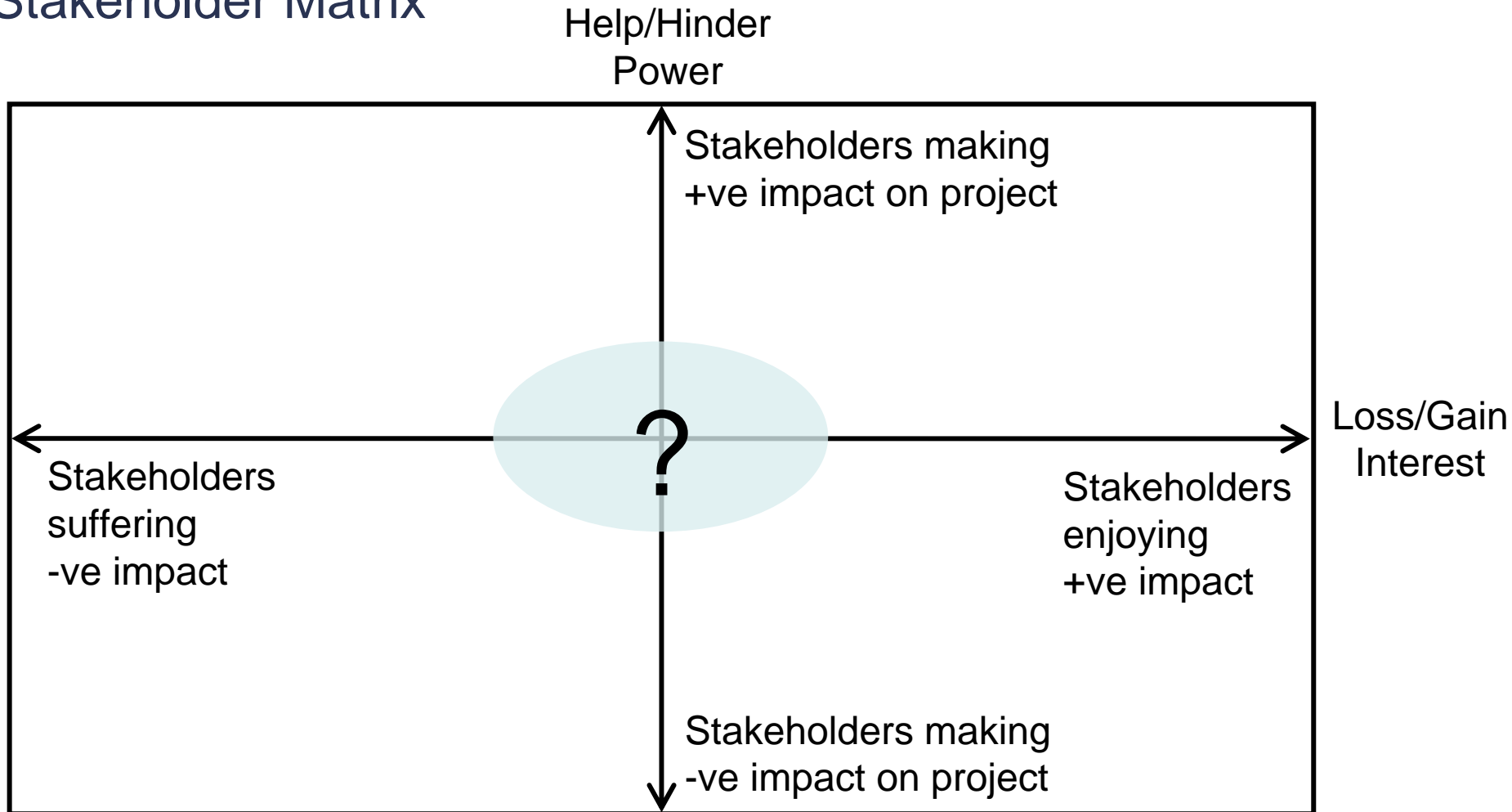
Elements of a Domain Map or Model

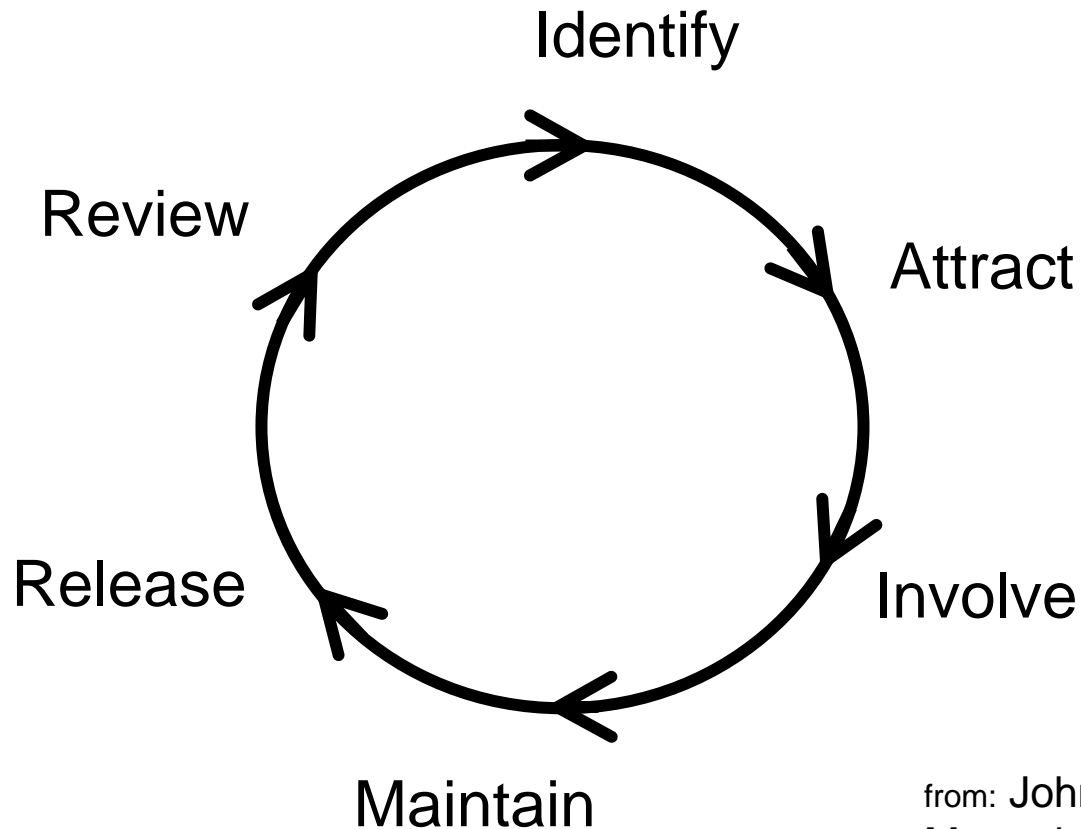


■ Stakeholders can:

- Help identify clarify the issues the project will address
- Help frame your project
- Help your project get off the ground
- Assist with development and provide review & feedback
- Get your project through difficult patches
- Spread the word to others
- Support adoption and realise potential benefit
- **but they can also Damage or even Kill your project**

Stakeholder Matrix





from: John McManus,
Managing Stakeholders in Software
Development Projects,
Elsevier Butterworth-Heineman, 2005

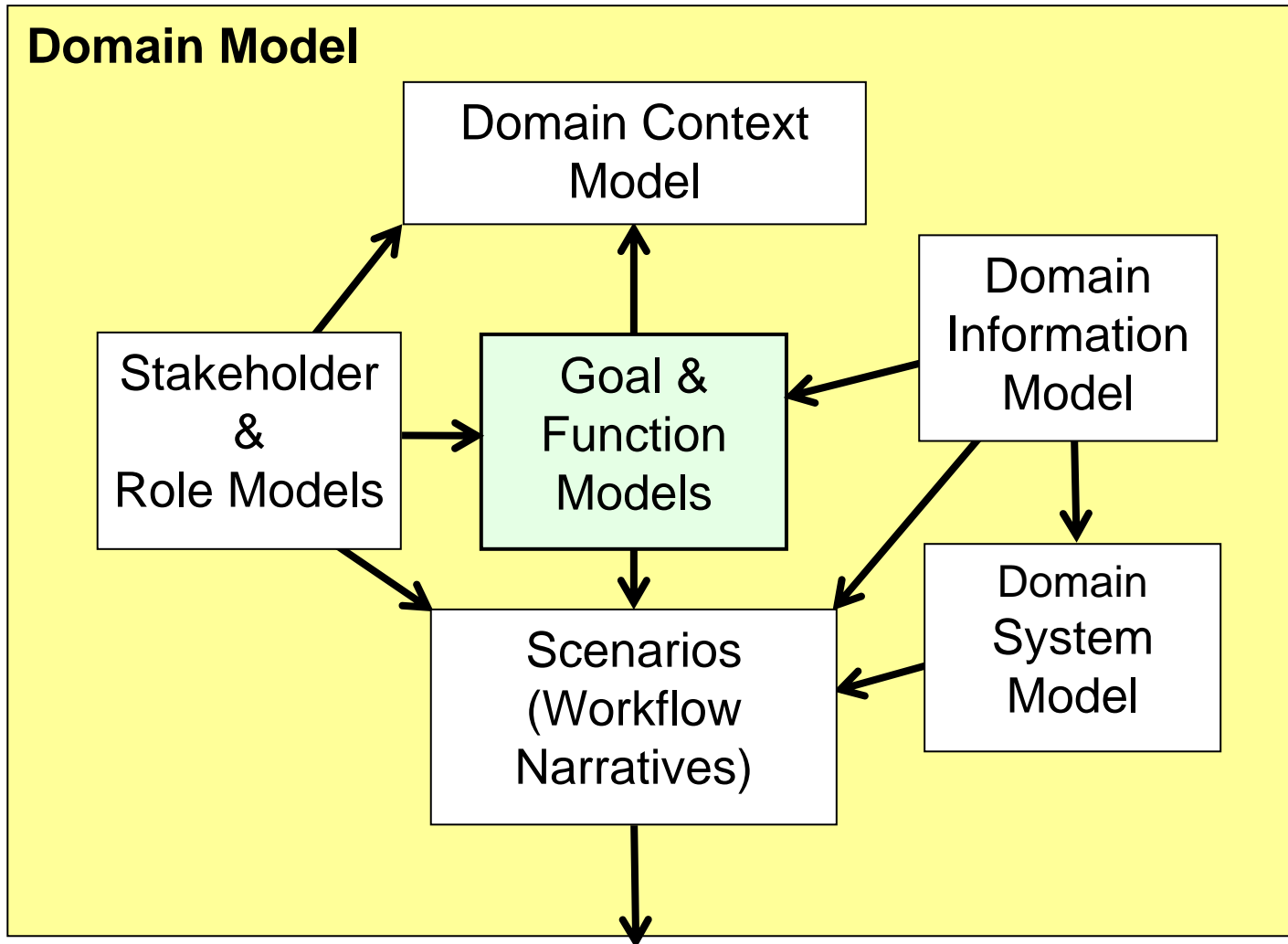
Performance Prism & the Stakeholder's Viewpoint

- Performance Prism (Cranfield University) Strategic Management Framework based on Stakeholder value
- Takes 360 view of stakeholders
- What are Stakeholders' wants and needs?
- What is needed or wanted from Stakeholders?
- Stakeholder modelling is used to assess value created
- Assessing value of Development Projects is difficult
- This, with adaptation, may be a useful approach

Andy Neely et al, The Performance Prism, Pearson Education, 2002

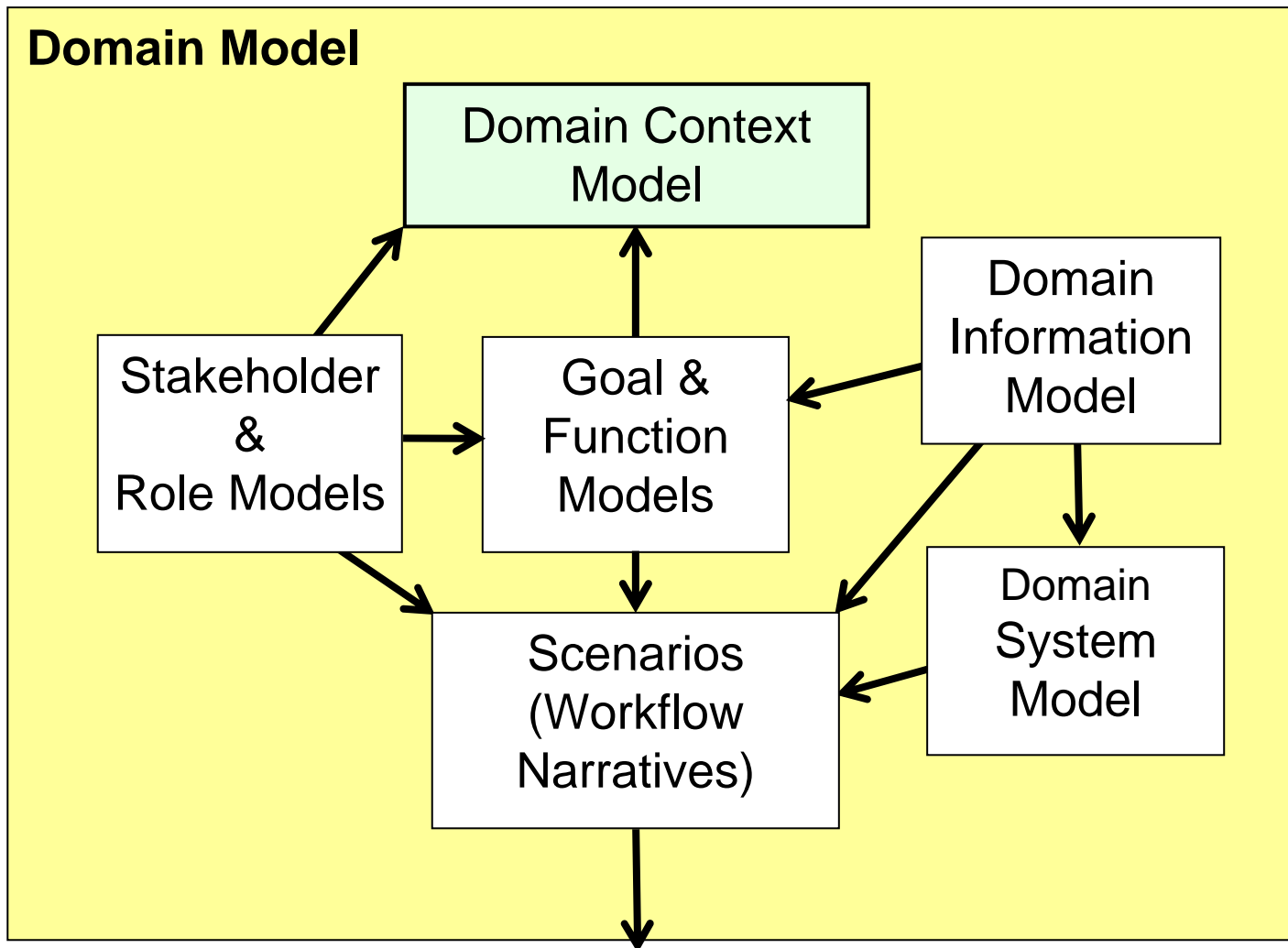
<http://www.som.cranfield.ac.uk/som/research/centres/cbp/products/prism.asp>

1. Stakeholder Satisfaction – who are the key stakeholders and what do they want and need?
2. Strategies – what strategies do we have to put in place to satisfy the wants and needs of these key stakeholders?
3. Processes – what critical processes do we require if we are to execute these strategies?
4. Capabilities – what capabilities do we need to operate and enhance these processes?
5. Stakeholder Contribution – what contributions do we require from our stakeholders if we are to maintain and develop these capabilities?

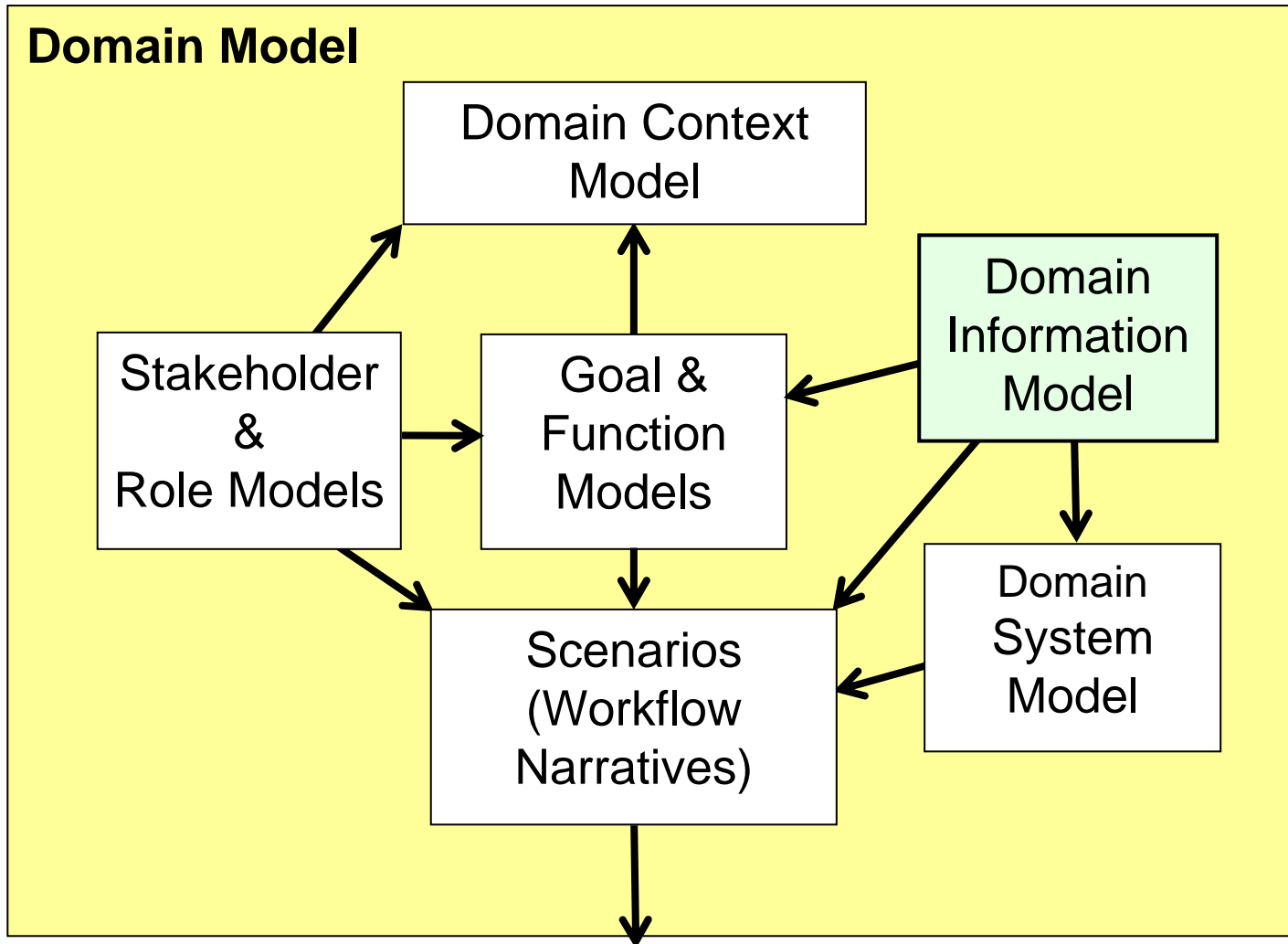


- Always ask “Why?”
- And then ask again three times! – Goals are nested
- What people do is, directly or indirectly, towards some goal
- To understand what they are doing, you need to understand their goal
- and usually, what further goal that achieving this goal will help to bring about
- If you haven't got any goals, you won't have any problems!
- To understand a problem, first identify the goal/s

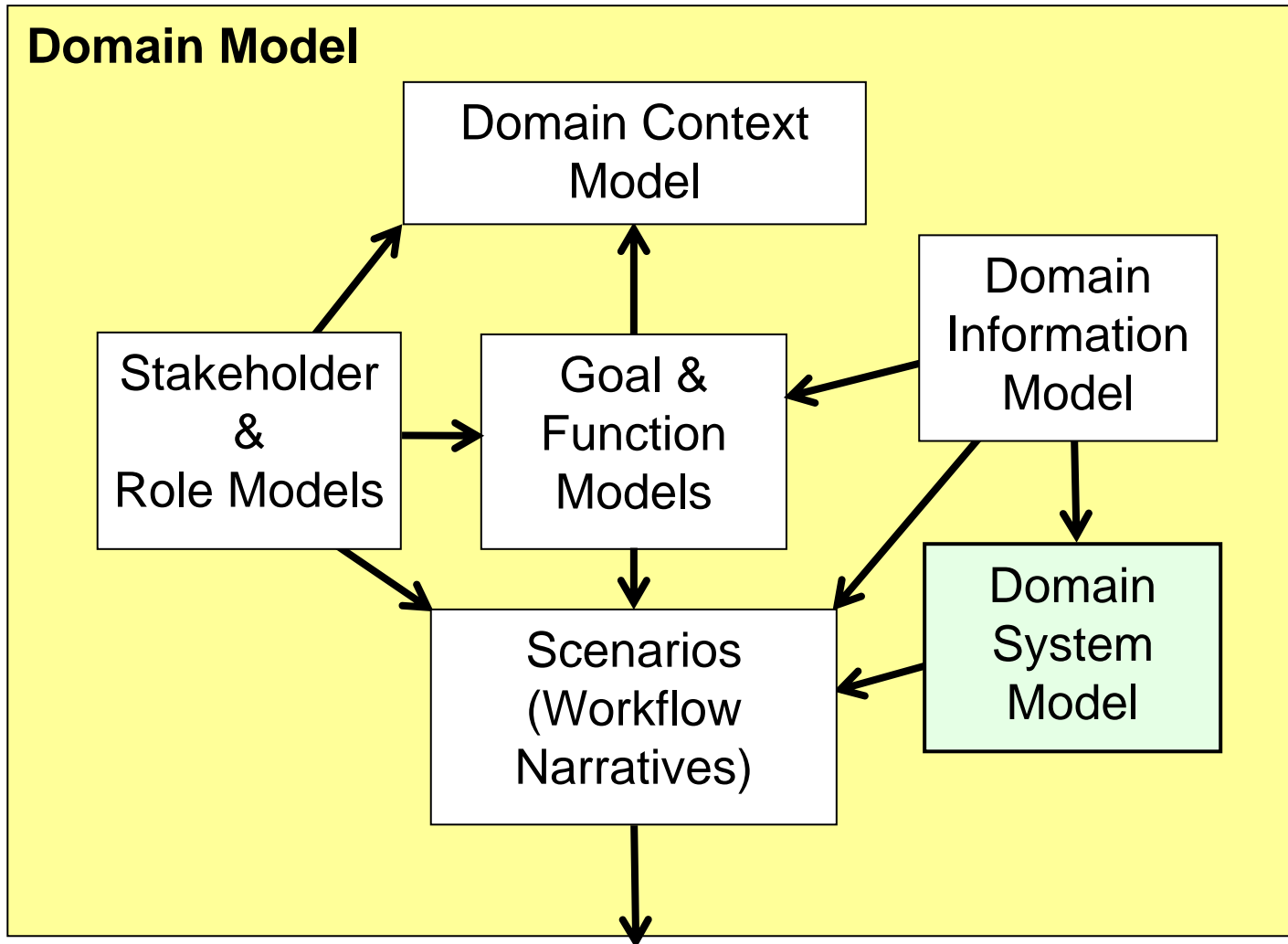
- Other terms could be used for Function: Task, Activity
- Here, a 'Domain Function' is a description of
 - WHAT is done to achieve the goals
- It is NOT a description of HOW that Function is performed
 - That comes later
- Some modelling approaches describe a Function in terms of its Inputs and Outputs and the Transformation it brings about
- Functions are linked to Goals, and may be nested also.



- A Domain does not carry out its activities in isolation
- It provides products or services to others
- It receives services and products from others
- A Domain Context Model
 - maps out other Domains
 - maps interactions with them
 - and is often the home of several external Stakeholders
- Sometimes internal problems can be ‘dissolved’ by making external changes (always worth checking first!)

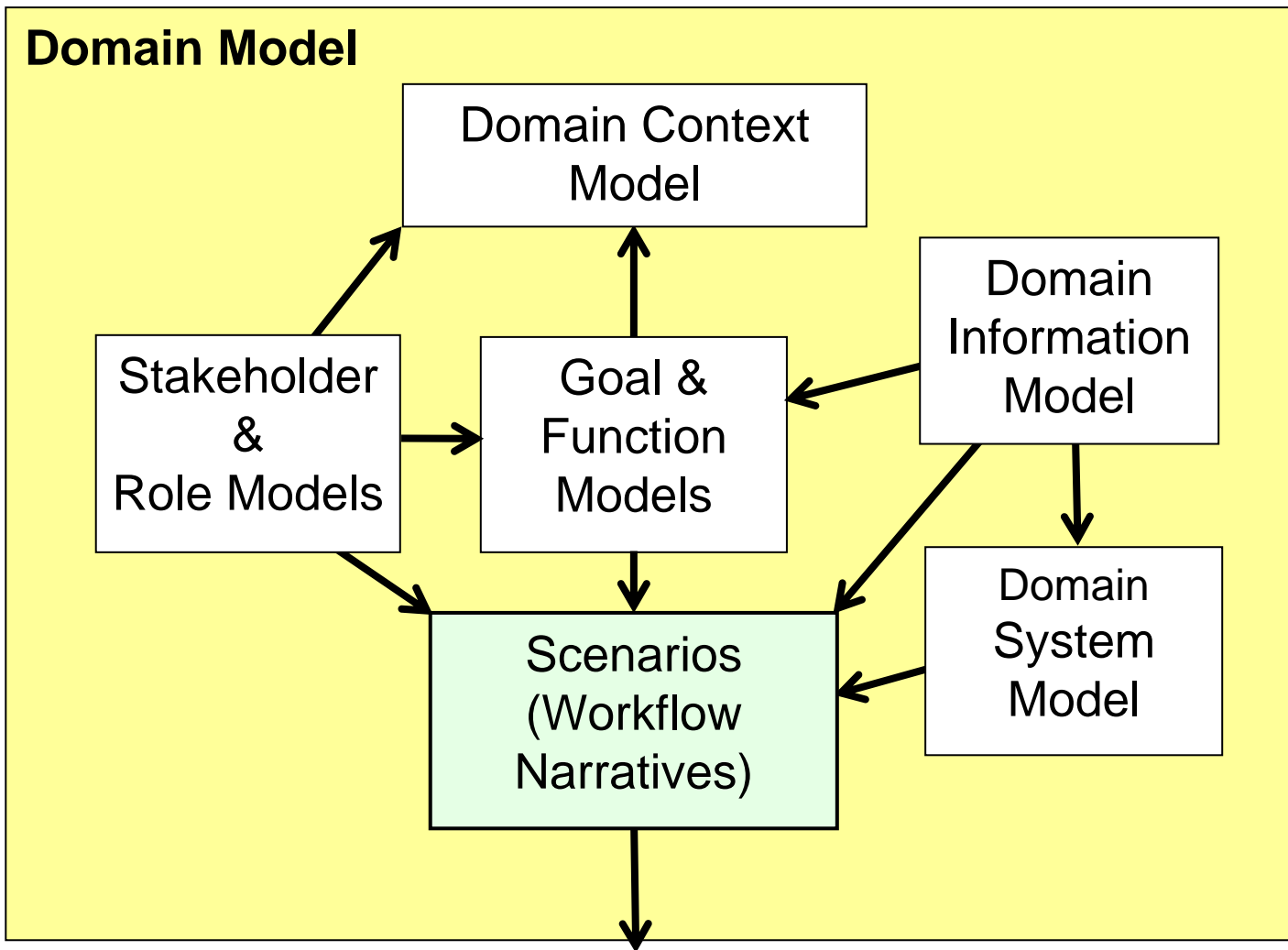


- This captures:
 - The core Information Entities that are commonly used within the Domain
 - The relationships between these Information Entities
 - It also captures what information is used for which Functions, within a Domain
 - It may reveal or capture cohesive sets of Information Entities that can be managed by Domain Services and reused in several Domain Applications
- It is generally more stable than Domain Processes



- This is intended to map the main systems typically used within a domain and how software functions are distributed (or not) across hardware
- It also maps the information exchanges between systems
- These are typically the points where interoperability specifications are needed
- Where there is no overall consistency within a domain, there may yet be clusters conforming to a small number of System models
- Failing this there will be an interesting set of institutionally unique models and worth knowing by those developing for the domain

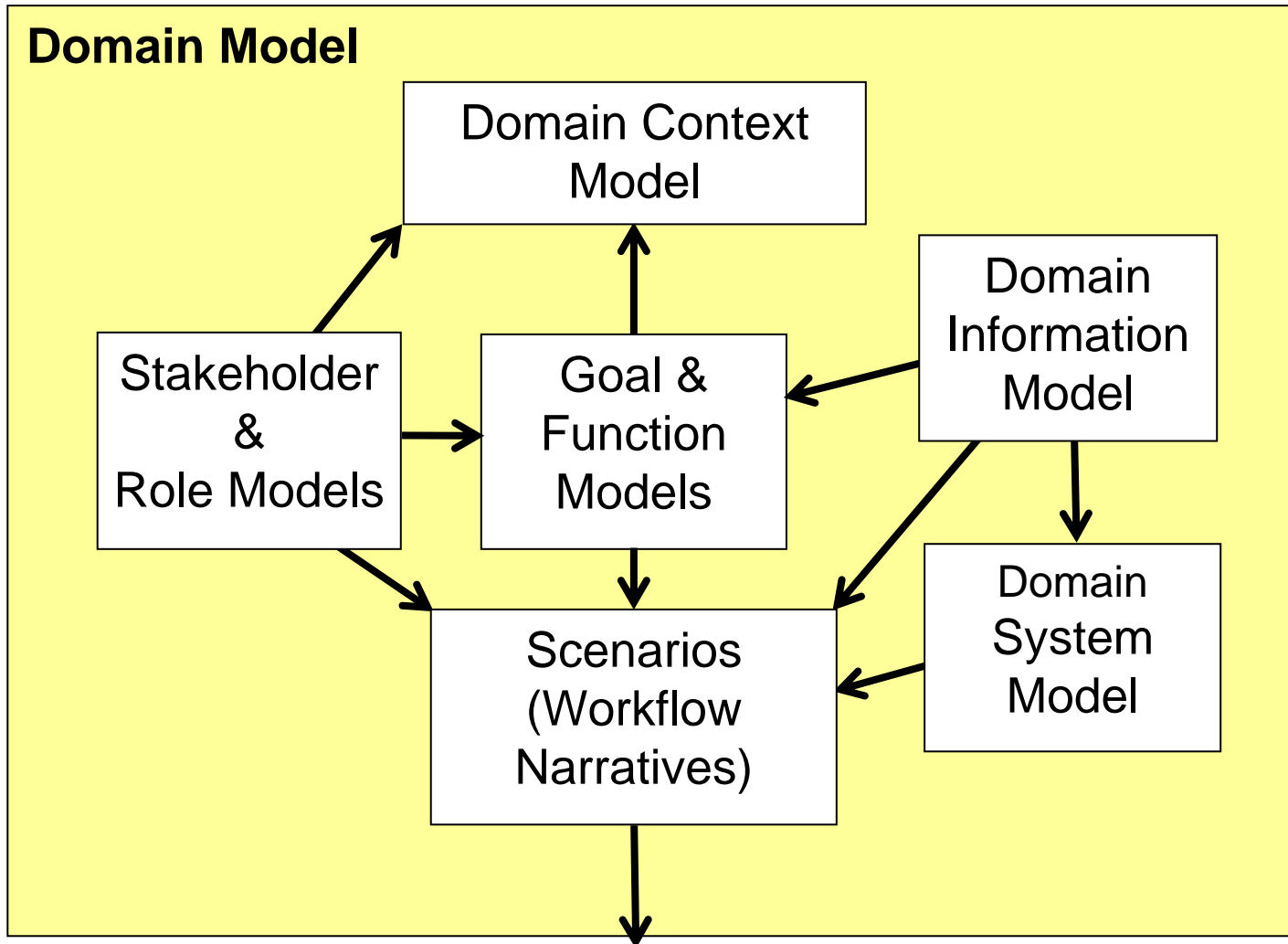
Scenarios (Workflow Narratives)



- A given Function can be carried out in many ways
 - from completely manual
 - to completely automated
 - but often it is a combination of the two
- Scenarios can be used map out these different alternatives
- They provide, in narrative form, a description of the HOW of a Function
- For some purposes, this narrative level is sufficient
- In other cases, more detailed study is required
- Here Scenarios provide input to formal Process Modelling

- Scenarios have been found to be a good way to engage people in the process of developing ICT based solutions
- They can be used to analyse the current situation (As-Is)
- They can also be used to explore future possibilities (To-Be)
- Being concrete in form, they enable all stakeholders to engage in the key issues of co-evolving practices, processes and ICT systems
- But more generally, Scenarios can be used for all type of modelling, not only for workflow narratives.

Elements of a Domain Map or Model



- Two approaches are often adopted to Modelling:
 - Top Down
 - Bottom Up
- There's also Middle Out
- But many have converged on the “Table and Legs” approach
 - Get a shallow but broad map of the whole space (table)
 - Based on priorities, drill down & model in detail the area of focus, typically initiating a development project (leg)
 - Repeating this process provides more legs for the table (as well as incrementally building the domain model)

- A Domain Map and Model **MUST** be owned by the community that owns the Domain
 - Only they will maintain it
 - Only they will put it to use with developers
- Start rough and ready and produce a sketch map
- Check it early with other members of the Domain
- It has to represent a consensus, where possible
- Where not, more than one map or model may be needed
- Focus on the agreed core and don't spend too long on areas of disagreement, only if they are critical to current work

- Alistair Sutcliffe (2002), *The Domain Theory: Patterns for Knowledge and Software Reuse*, Lawrence Erlbaum Associates
- Irene Polikoff et al. (2006), *Capability Cases: A Solution Envisioning Approach*, Addison Wesley
(Covers that creative gap between analysis and design that most methodologies seem to skip over. Good on including users and stakeholders. They provide detailed description of a rapid (typically three day) process, that will take you at least a week to feel comfortable with)
- Czarnecki and Eisenbaum (2000), *Generative Software*, Addison Wesley
(First part focuses on Domain Engineering as the basis for requirements-driven, component-based development)
- John McManus (2005), *Managing Stakeholders in Software Development Projects*, Elsevier Butterworth-Heineman
- Andy Neely et al (2002), *The Performance Prism*, Pearson Education