Overview – then and now

SoSyM

- Recap: Requirements and Analysis

- Bad analysis

- Problem Frames

- *Elicitation – very brief overview*

- Recap and Implications

Recap – Analysis

- Requirements (domain), spec (interface between domain and system).

- Traditional analysis:

- SA: Tendency to model existing system, but Still provides useful notational tool-kit

- OO: Ok for design, but not analysis objects, use cases not really suited to requirements.

- So what else is there?

Homework / Exercise

- EX 1 Find an example of analysis (structured or OO, or even use cases) that you don't like and:
  - Say what's wrong with it. Remember:
    - Does it really model the problem or domain?
    - Are the objects real?
    - Does it reflect the problem?
  - IFF you can't find one, try and invent one.
  - Send to me (as 1 or 2 Slides: e.g., picture and critique).
- EX 2: Types of Requirements (courtesy if Ian Bray).
  - Identify type from given list.

Some Homework Answers

- More to come (no doubt) but so far:
- Use Case Diagrams (2)
- Class diagrams
- DFDs

PDOA Premise

- *"…knowing the **type of problem** should help **guide** the analysis and specification:*
  - *what questions to ask*
  - *what aspects of the PD to describe*
  - *the nature of the requirements*
  - *which modelling techniques to use etc..*
  - *(and, ultimately, give guidance on the idesign solution)".*

  Bray – RE Slides

SoSyM

Problem Frames & PDOA

- Bray Outlines (his) PDOA method as:
  - *"Collect basic information and develop **problem frame**(s) in order to establish the **type** of the PD*
  - ***Guided by the PF type(s),** collect further detail and develop a **description** of the relevant characteristics of the **PD.***

    (This description may well incorporate conventional models such as Context diagrams, ERDs etc.)
  - *In conjunction with the foregoing, collect and document the **requirements** for the new system"*

Types of Problem Frame

Jackson suggests a classification:

- **Control:**
  - **Required behaviour**
  - **Commanded behaviour**
- **Workpiece**
- **Information**
  - **Continuous display**
  - **Requested reports**
- **Transformation**
- **Connection**

(and recognises that there may be more)

Problem Frame Domains

- **Problem frames diagrams** are similar to context diagrams (where domains are like terminators, that is, interact with the system),
- In addition to showing the connection to the (proposed) system we show the relationship *among* domains.
- Note that this is forbidden in standard context diagrams, but here were are not just showing the system context (delineating the system boundary), but want to show the entire problem context.
- That is, we model the **relationship**s among **domains** (or the sub-domains of the problem).
- Domains are relevant, identifiable, elements of the problem.
  - E.g., For a transformation problem, the input and output files would each be a domain

SoSyM

Domain Notation
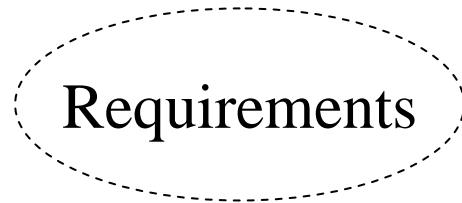
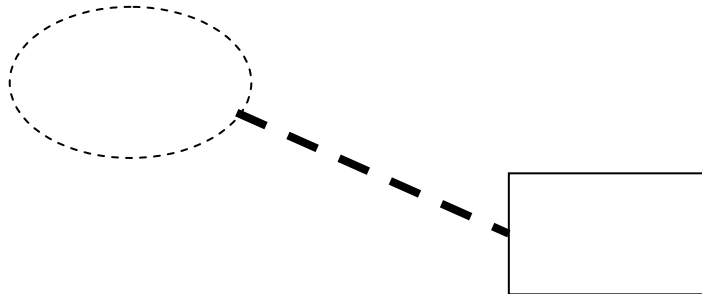*SoSyM*

Domain name

- Domains

Solution
system name

- The SS (machine) is a "special" domain and has a special symbol:

Domain name

- Domains (apart from the SS) that can be designed by the developer (realised domains).
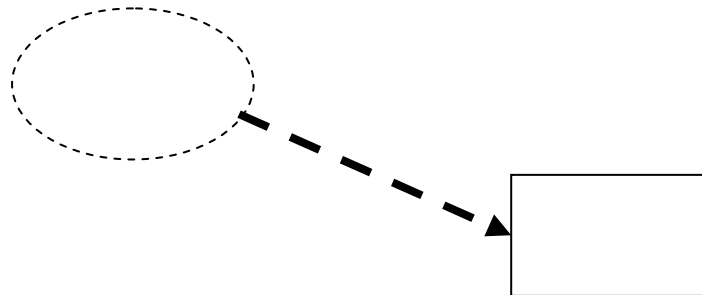
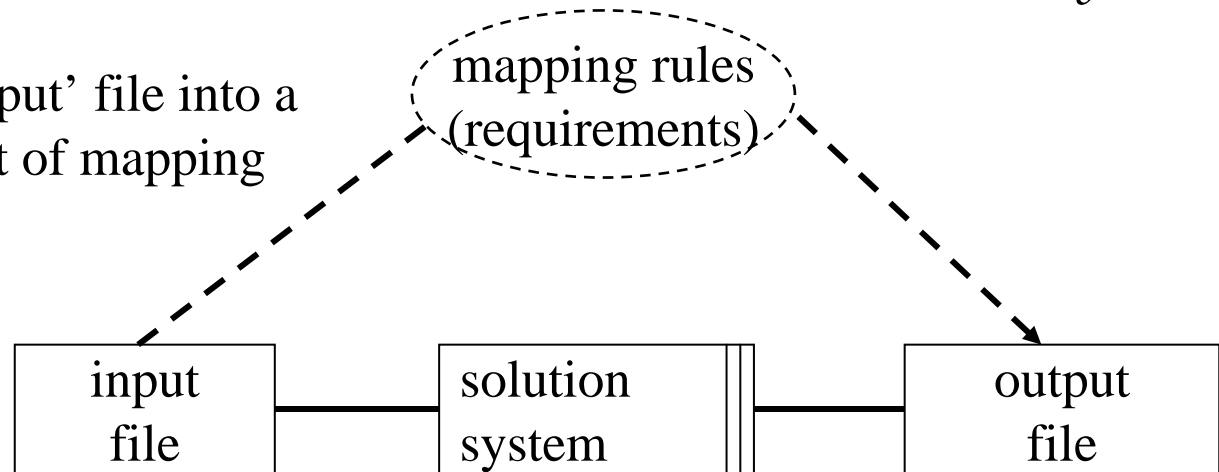- Relationships (interactions) among domains.

Requirements

- The requirements

- Where the requirements reference (refer to) a domain.
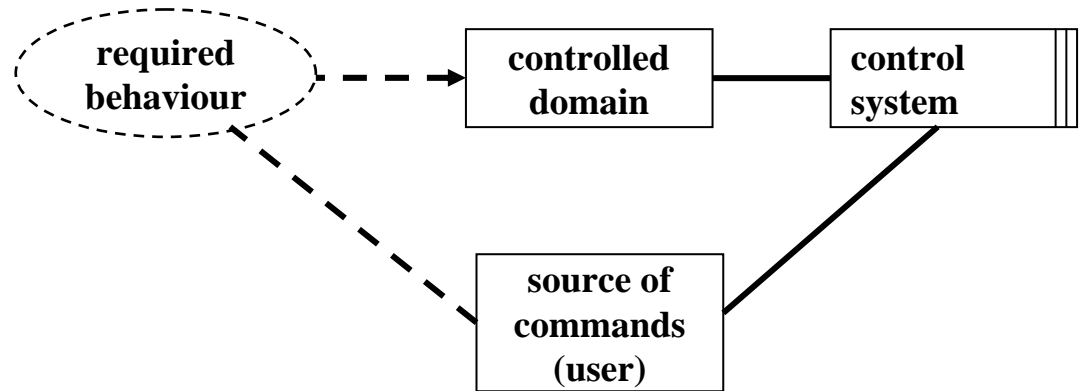
- Where the requirements constrain (impose upon) a domain.

Transformation Frame

SoSyM

- Transforms some 'input' file into a 'output file', via a set of mapping rules.

```
    mapping rules
    (requirements)
```

| input file	solution system ‖	output file

- Hence, the transformation must follow the mapping rules (which thus constitute the requirements).

- Therefore, the domains are the input file, the output file and the system, which has an interaction with both of them.

- The requirements refer to the transaction file and constrain the statement file.

Control Frame

- Required behaviour: controls some part of reality according to defined rules.

- Commanded behaviour: controls some part of reality according to operators commands

```
  ( required )          controlled          control
  ( behaviour )  ---->   domain     ------   system
                                    \
                          source of  \
                          commands
                          (user)
```

- Control system 'controls' (interacts with) the controlled domain.
- The user (source of commands) interacts with the system.
- The requirements refer to the user (source of commands).
- The requirements (required behaviour) constrain the domain.

Information Frame

- Continuous (automatic) reporting: automatically provides information about some part of reality

- Requested (upon demand) reporting: provides information about some part of reality upon request
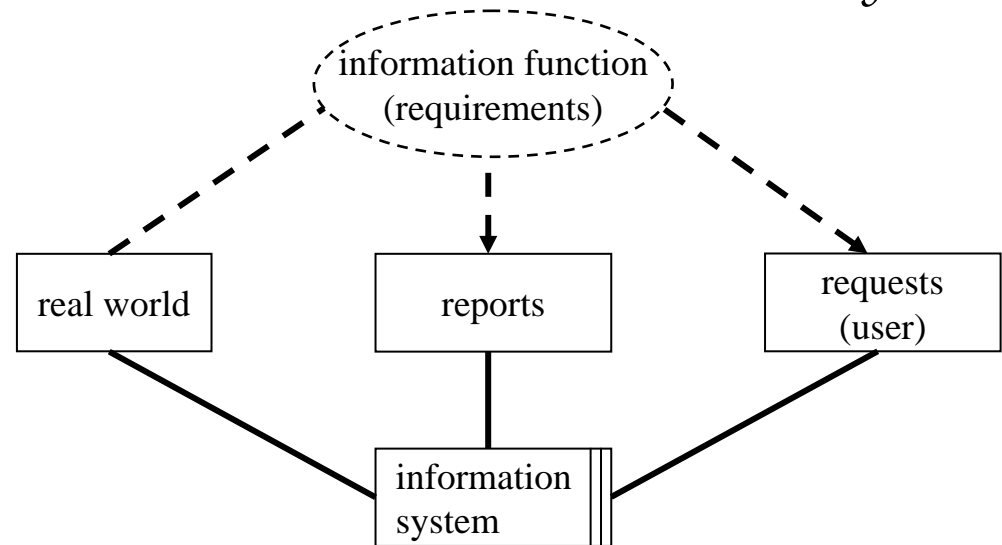
- Information function requirements refer to the *real world*.
- Information function constrains *user* (and requests), and *reports*.
- The information system interacts with the real world, user and reports (which it produces).
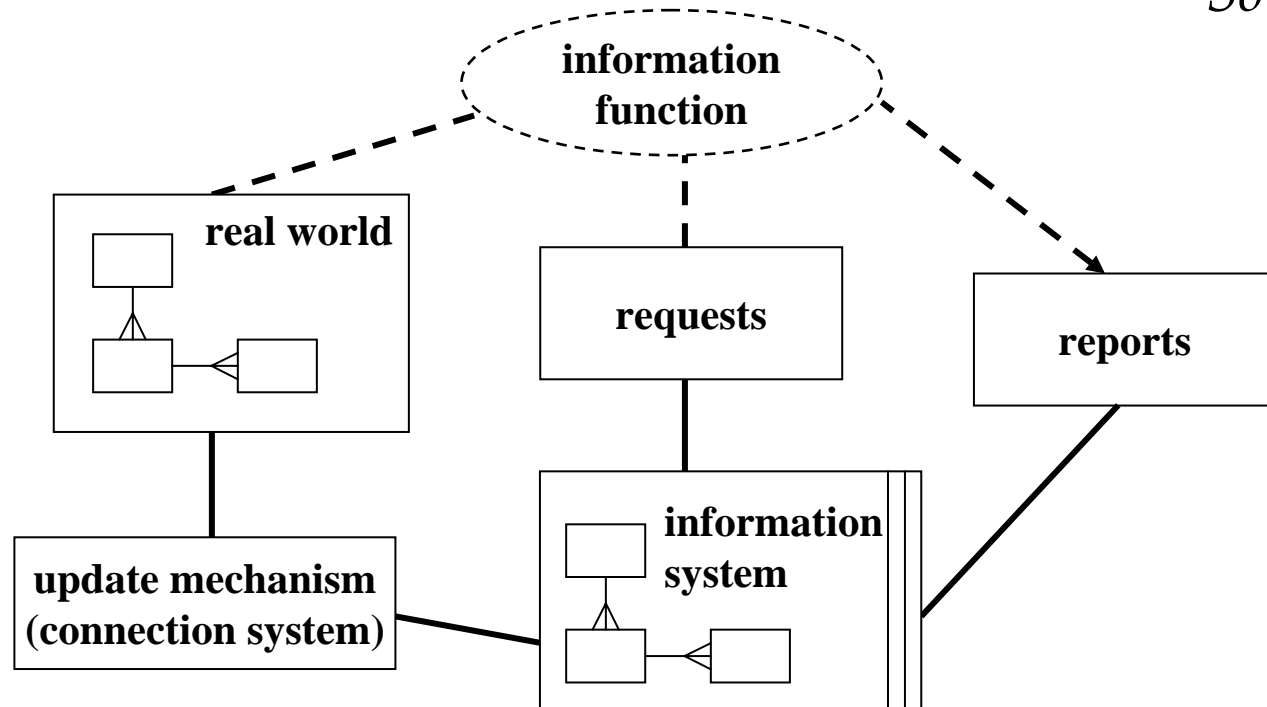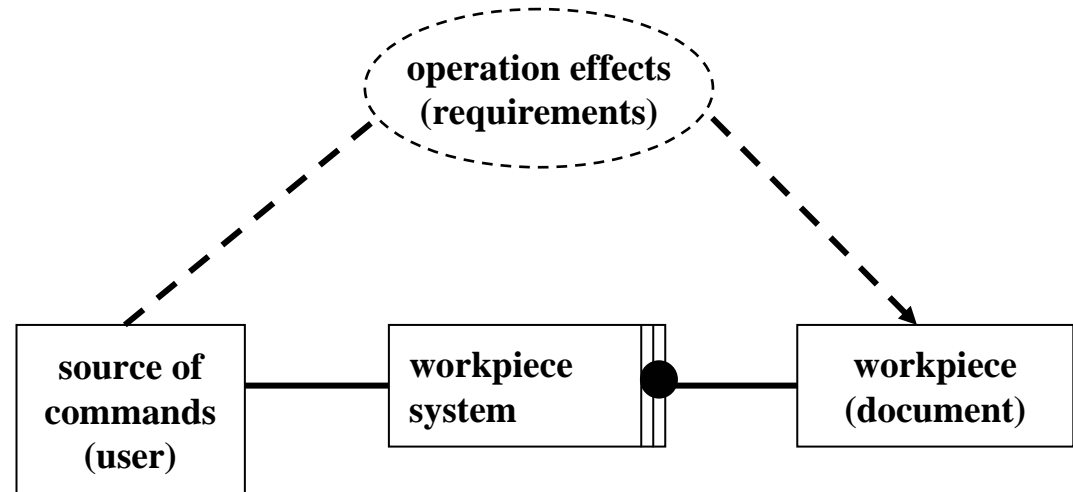
- The IS will often *contain a model* of the real world data.
- Note data model of domain (analysis), may be different to data model within the system.

Workpiece Frame

- User creates and edits a workpiece (e.g., a document or other artefact which exists within the system).

- The dot shows that one domain (the document) is contained within another (the workpiece system).


- Workpiece system contains the workpiece

- The user (source of commands) interacts with the workpiece system.

- The requirements refer to the user (source of commands).

- The requirements (required behaviour) constrain the workpiece domain.

Connection Frame

- Common version resembles transformation frame but: the transformation system **generates new data** from old, the connection system just moves (logical) data from A to B.
- Requirements concern the acceptable delays and distortions that the system can introduce.
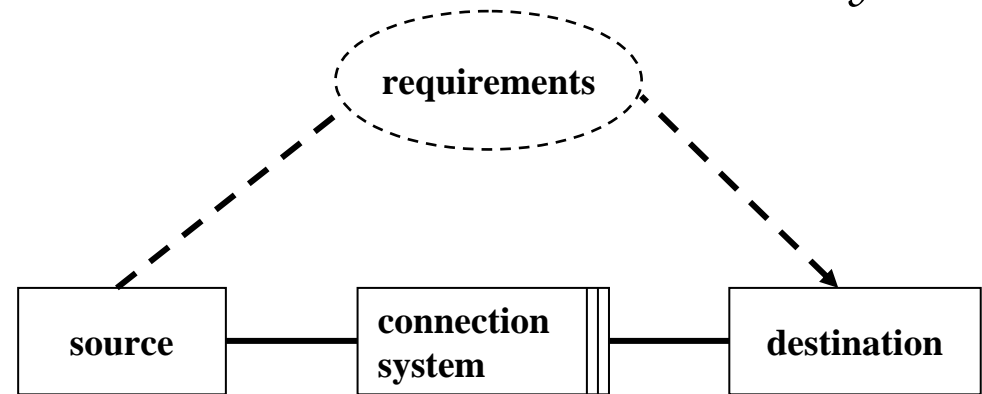
- Connection system interacts with source file and destination file.
- The requirements refer to the source file and constrain the destination file.

PDOA Guides

- *"type of problem should help guide…"*
- Kovitz suggests tables for each frame.
- "This is just as useful for knowing what not to do, as what one does need to do". E.g., transformation frame

requirements document content	(some) relevant techniques
Input and output data sets	data dictionary (BNF) structure charts
Source and destination for data	Text
Requirements - mapping between input and output	Text, table, (Jackson) structure charts

SoSyM

Control System Frame

requirements document content	(some) relevant techniques
Data model for sub-domains in the controlled domain (Note, *optional – there may be nothing significant*)	Entity Relationship Diagram (ERD) and Data Dictionary (BNF)
Causal laws in the controlled domain, (i.e., the **innate** behaviour of **each thing** in the controlled domain) including actions/events that the sub-domains can perform/undergo	Event list, Finite State Machine (STD), Decision table
Shared phenomena through which the computer can **monitor** the controlled domain	Text (event list)
Actions in the controlled domain that the machine will be capable of initiating	Text (action list)
Requirements - **behaviour rules** (i.e., how the controlled domain as a whole should be made to behave)	Text, FSM, DT

Information System Frame

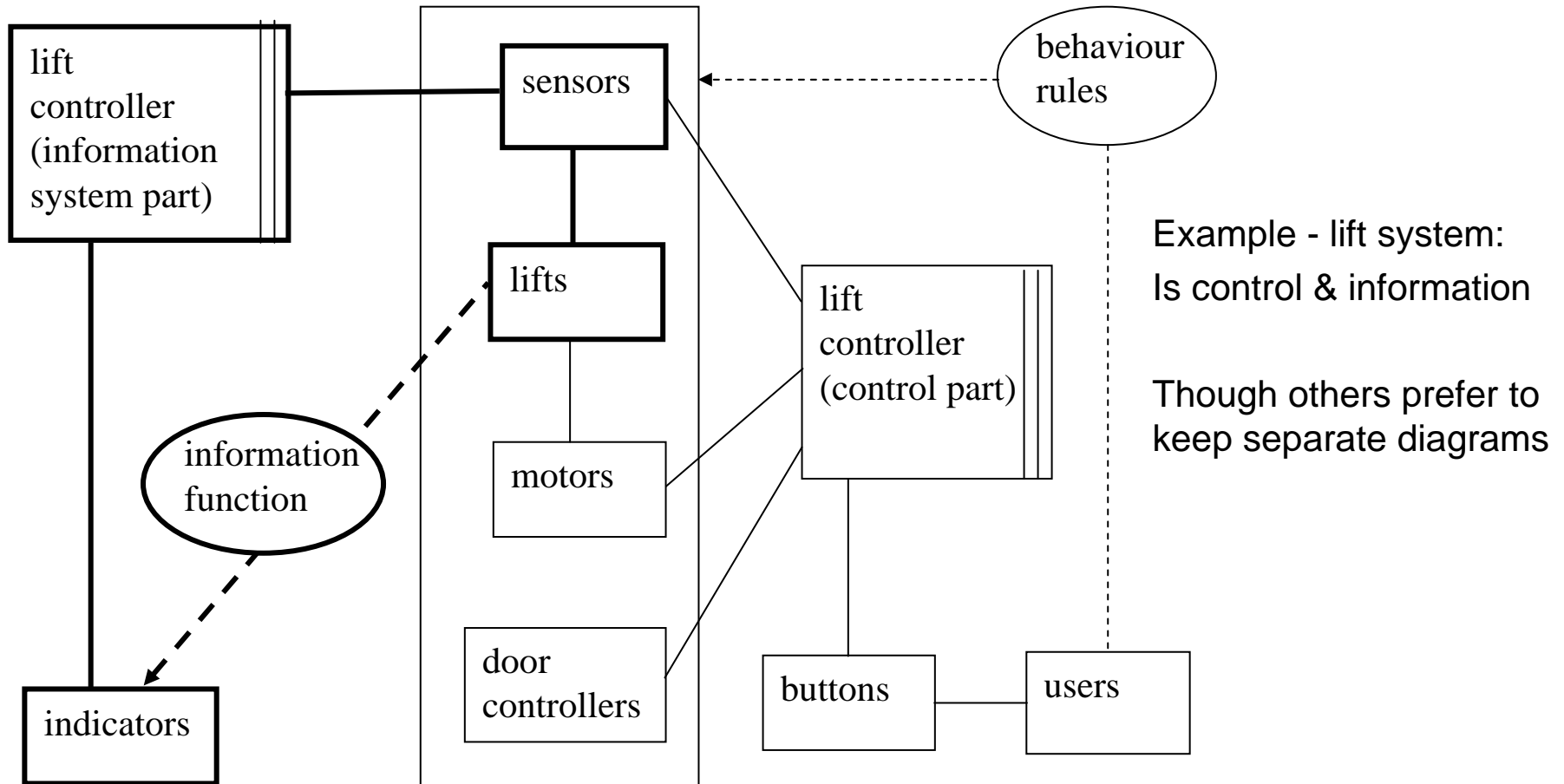| requirements document content	(some) relevant techniques
Things in the real world and their attributes and relationships (**data model**)	Entity Relationship Diagram (ERD) and Data Dictionary (BNF)
All real world **events** that change the results of queries and the sequences in which those events can occur	Event list and Entity Life History
How the system can **access** the real world state and events?	Text
Requirements – the **queries** to be supported and the corresponding **reports**	Text
File formats for any existing files that the system needs to access	File maps, structure charts, BNF
Distortions and delays introduced by any connection domain	Text

Workpiece and Notations

- Note, suggests ERD etc, never, say, class diagram.
- Some (e.g., Choppy) tried to substitute other notations (and some BU projects).
- And others to domains (e.g., GIS frames).
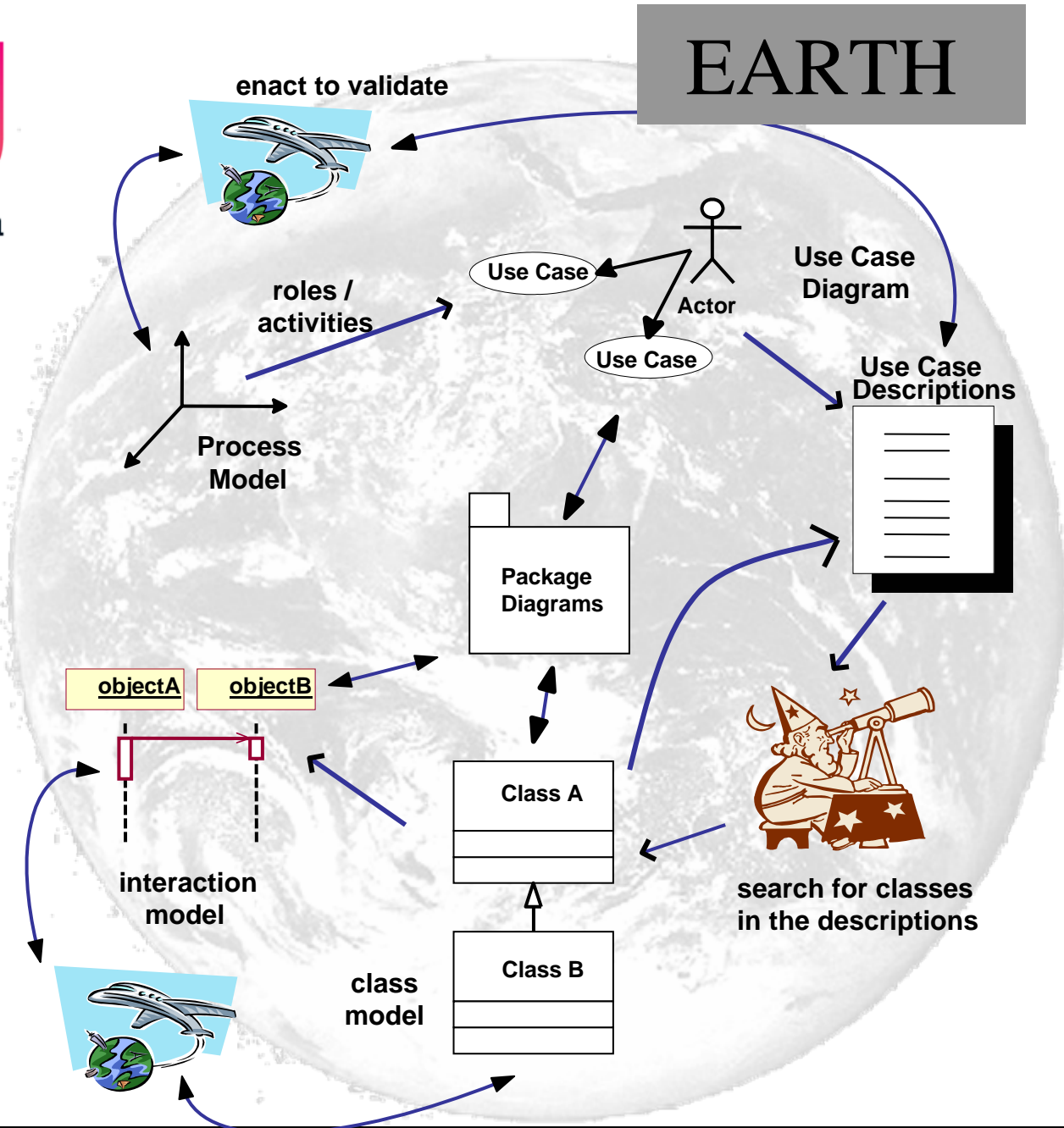- So some guidance (albeit may have to substitute) where problem is one frame
- BUT…

requirements document content	(some) relevant techniques
Workpiece – legitimate, logical structure	BNF, File map, SC(?)
Requirements – the required commands/operations and the effects that the operations should have upon the workpiece	FSM/STD, text, DT (?)

SoSyM

Introduction to Requirements Engineering

Multiple Frames

lift controller (information system part)

sensors

behaviour rules

lifts

information function

motors

lift controller (control part)

door controllers

buttons

users

indicators

Example - lift system:
Is control & information

Though others prefer to keep separate diagrams
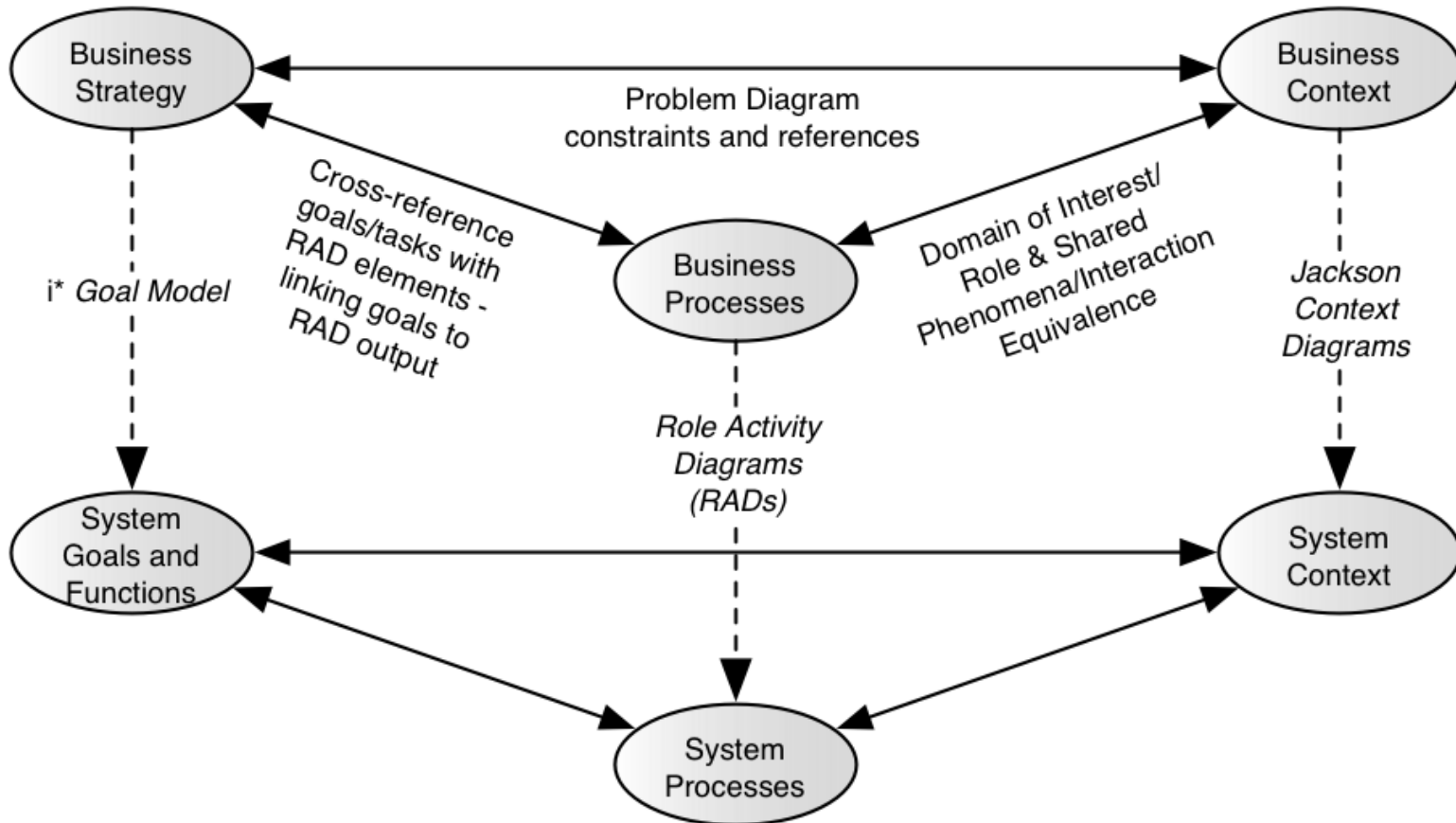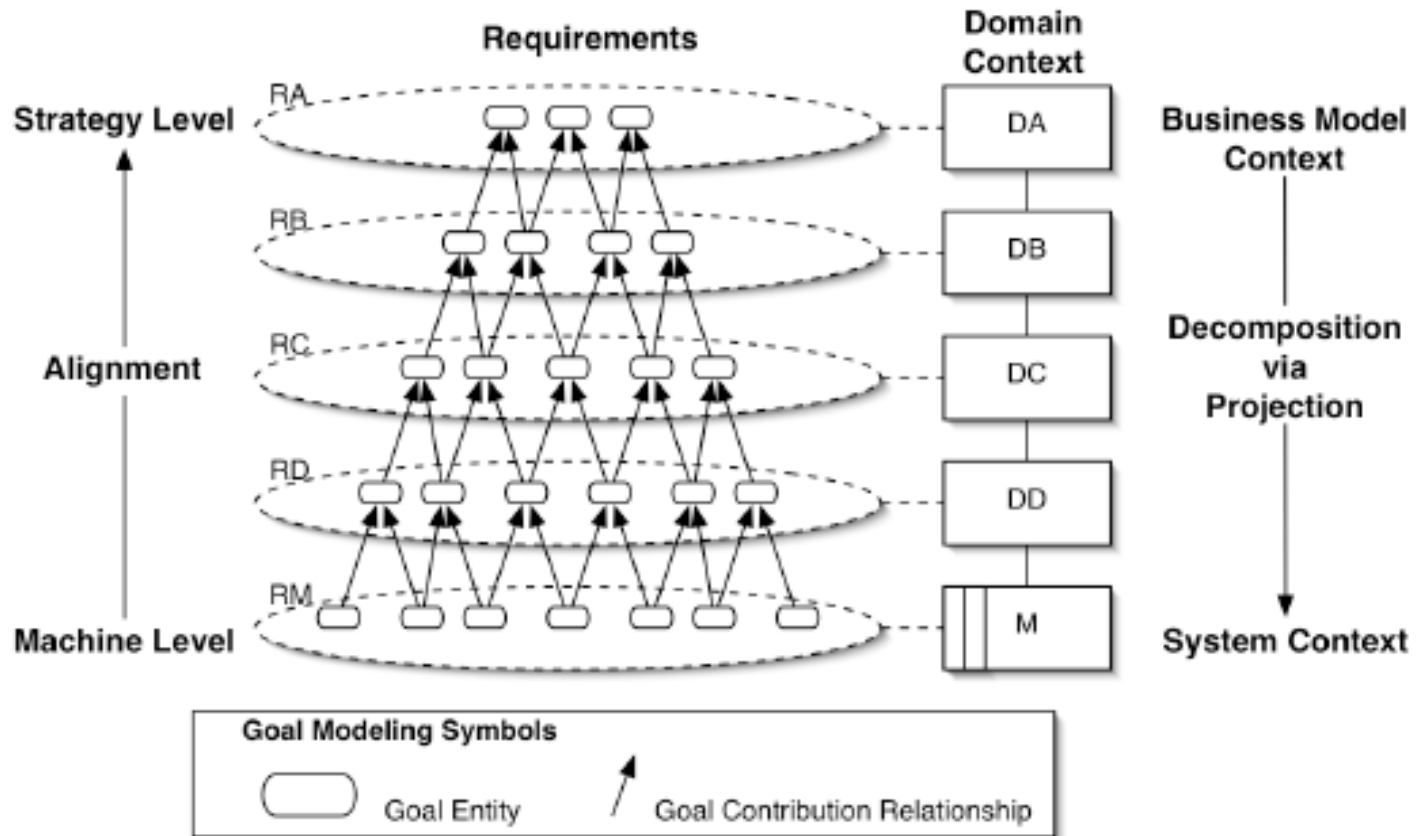
Process Oriented Approaches

- Ideas originate in 1990s (but are taken up…)
- Follow process modelling ideas (BPM)
- Early methods tend to adopt proprietary approaches (e.g., Warboys et al.)
- Later (pragmatic) attempts to fit to popular approaches, notably the UML (EARTH)
- Recent innovations move upstream to business goals and business strategy (e.g., B-SCP)
- Recent moves back to tool support
  - Simple mappings BPM to UC or
  - Complex tool-sets (VIDE)

EARTH

enact to validate

roles / activities

Process Model

Use Case

Use Case

Actor

Use Case Diagram

Use Case Descriptions

Package Diagrams

objectA   objectB

interaction model

Class A

Class B

class model

search for classes in the descriptions

Layers in B-SCP

REBNITA

- Process modelling, role based models & enactable models
  - Involving *stakeholders*.
  - User-facing models. (***Audience***)
  - Industrial users: Like them but *too much effort*
- Use Cases (stuck with them)
  - Support for use case case guidelines
- Mapping
  - Problems moving from business models to specification – loss of 'richness'

Business model (strategic)

Process model (operational, e.g., RAD)

Use case (specification)

SoSyM

Recap and Implications

- SA/SD flawed, but contains useful techniques / notations.

- OO Ok for design, NOT analysis

- Problem frames: useful to understand, help consider appropriate methods, but difficult for multi-frame.

- Process oriented. Some thorough (BSCP), but significant effort (goals, PFs, PMs etc…)
  - Goal modelling particularly time-consuming

- Key elements…

Implications for RE

- Process models useful (and used) to understand domain.
- Useful as part of elicitation AND documentation.
- Must still have description for requirements.
- Other elements dependent upon type of problem.
  - E.g., DFDs to show document flow or processing
  - ERDs are good to show data models
  - State-based models good to show control / dependencies
- Use cases (scenarios) liked by users. Even if predominantly spec.
- *Implications for RE document?*

- Methods of investigating and eliciting requirements:
  - Interview
  - Document/record searching
  - Procedures (existing) and process models (existing)
  - Strategy or goal documents.
  - Modelling the 'essence of' process (Business models)
    - Note SA modelled existing processes or systems
    - Possibly: modelling the frames or domains of interest.
  - Questionnaires
  - Observation (includes task analysis etc.)

Exercise

- *The identification of problem frames*

- *You are asked to ID the frame for a given title / scenario.*

- *Please do this exercise on your own.*

References

- Michael Jackson, *Software Requirements and Specifications*, Addison-Wesley, 1995.
    - A seminal work. An entertaining, insightful book. Problem frames were presented to world for the first time. A classic. Most of this presentation is inspired by or taken from this work.
- Michael Jackson, *Problem Frames*, Addison-Wesley, 2001.
    - Yet another seminal work. This book focusses entirely on understanding and describing problems in much more detail than his 1995 work. This book tends to frighten (some) academics. Practitioners love it. A classic.
- Ben Kovitz, *Practical Software Requirements*, Manning Publications, 1999.
    - Described as idiosyncratic by an academic, described as delightful by a practitioner. Who are you going to believe? An excellent introduction to problem frames but can be hard going elsewhere.
- Ian Bray, *An Introduction to Requirements Engineering*, Addison-Wesley, 2002.
    - Was going to be called a "duffer's guide" but Ian soon realised that duffers are no good at requirements! An excellent introduction to a whole range of techniques and tools, with a focus on problem frames.

More references

- Don Gause and Gerry Weinberg, *Exploring Requirements*, Dorset House, 1989.
  - One of the very first books on requirements elicitation and still just about the best. An entertaining, seminal classic.
- Don Gause and Gerry Weinberg, *Are Your Lights On?* Dorset House, 1990.
  - Yet another classic, more generally focussed than the 1989 book. More cartoons in this book!
- Alan Davis, *Just Enough Requirements Management*, Dorset House, 2005
  - Al Davis talks about what is really needed in practice and discusses the critical idea of requirements triage further.
- Bashar Nuseibeh and Steve Easterbrook, "RE: The Roadmap", Int. Conf. on Software Engineering, 2000
  - A paper! This set out the future research directions of RE and provides an excellent introduction to the subject. A classic of its own.

Use Case Example

Clifford Williams
Sarah Alexander
Lee Gladding
David Zimmermann
David Harding
Julia Stanton

SoSyM
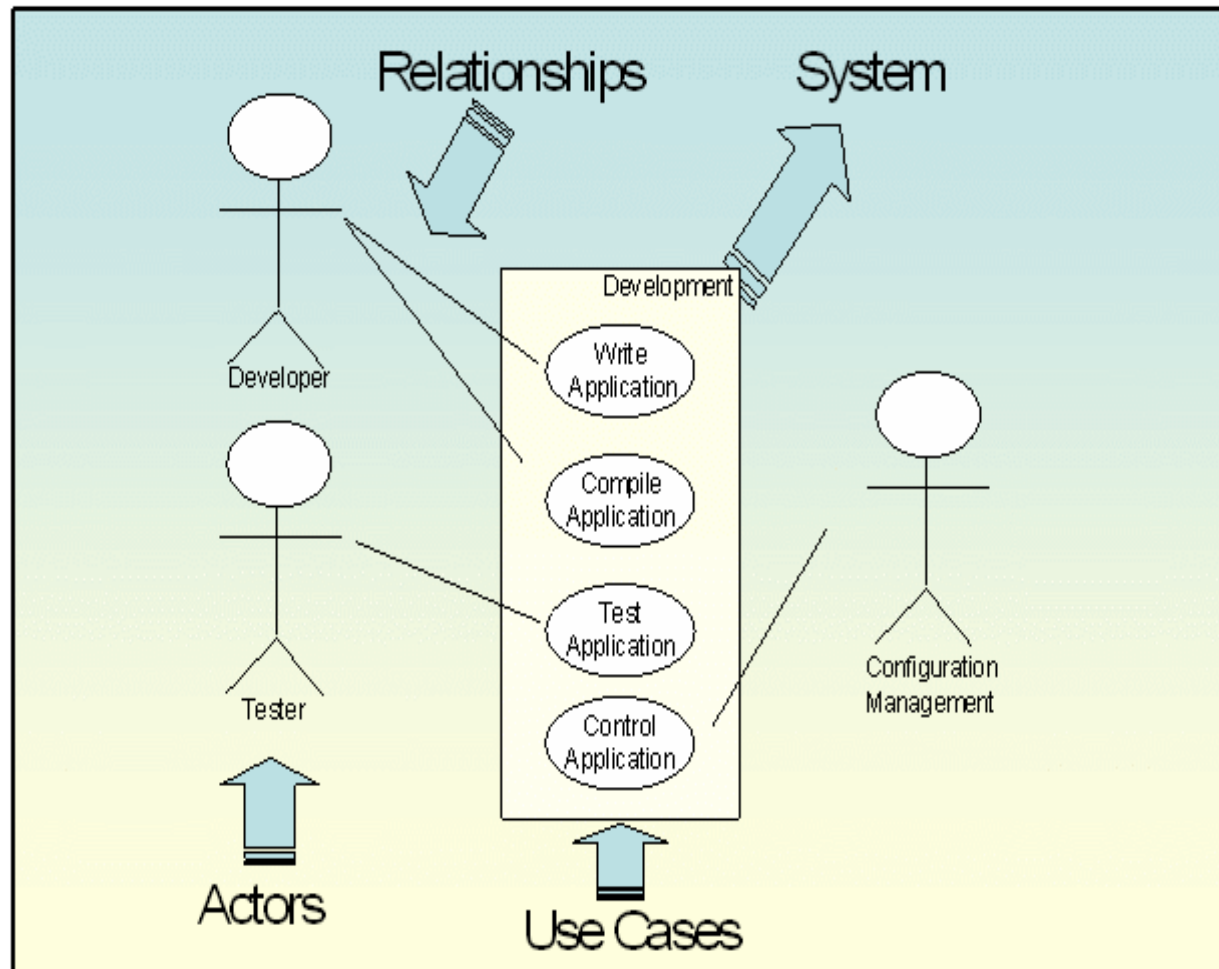
Problems

- Doesn't state the problem domain

- No input of a design

- Processes don't link to each other

- The model does not show any role for designing the problem

- Very ambiguous cases

- Use case bubbles do not imply or show a flow, even though the problem looks like it needs a flow of the system

- No sign of any deliverables after any function

- Very low level of detail – not enough to capture the functional requirements of the system

Pragash. V

Mark Smith

Ricky Dunn

Leigh Darlow

Oliver Trendle
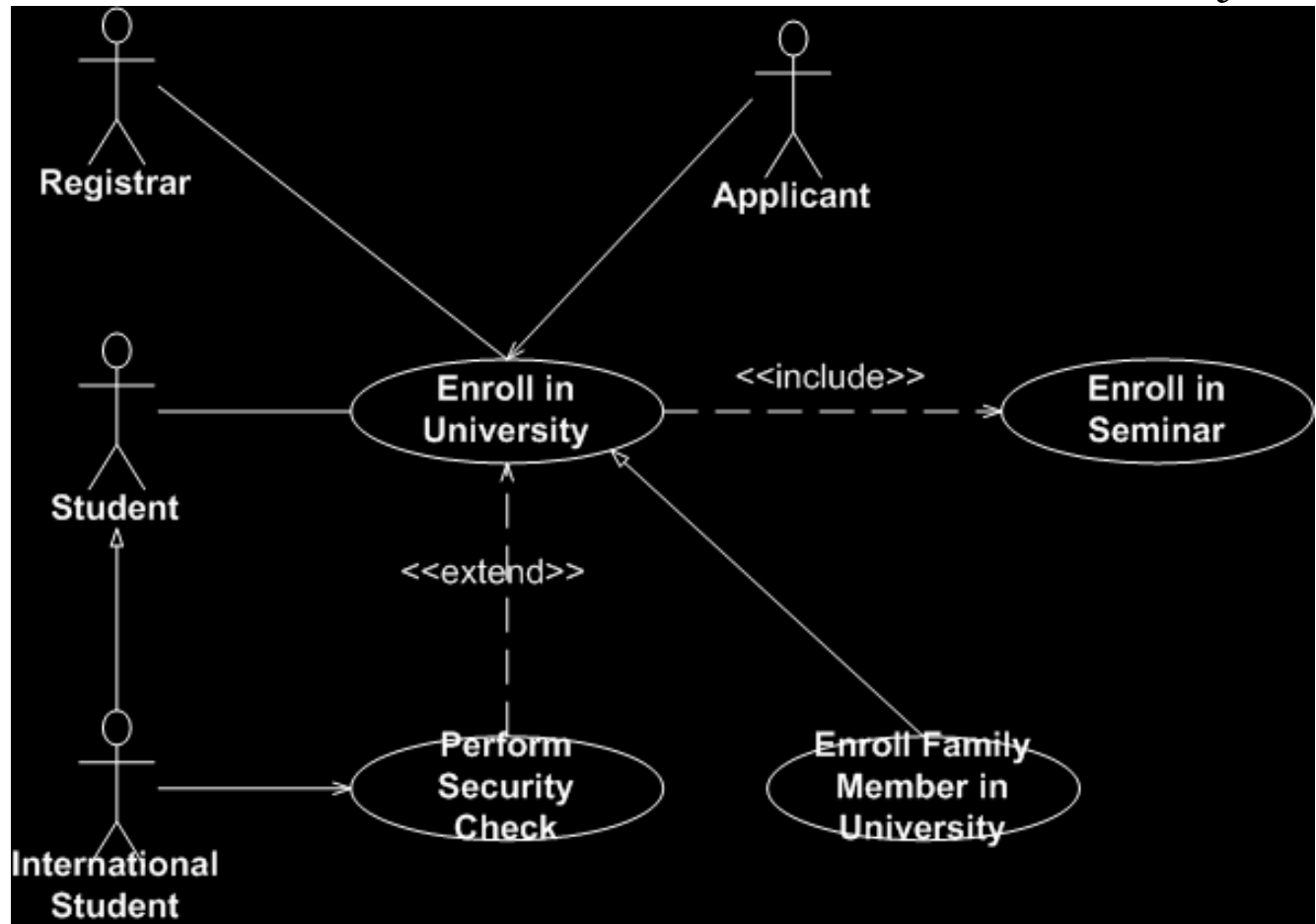
Problems

Doesn't state the problem domain

Use case is too vague, needs more detail

Undefined flow of data

No outputs

Ambiguity between actors (Student and applicant)

Ambiguity between actors roles eg what does Registrar do?

Why only international students get security check??

SoSyM