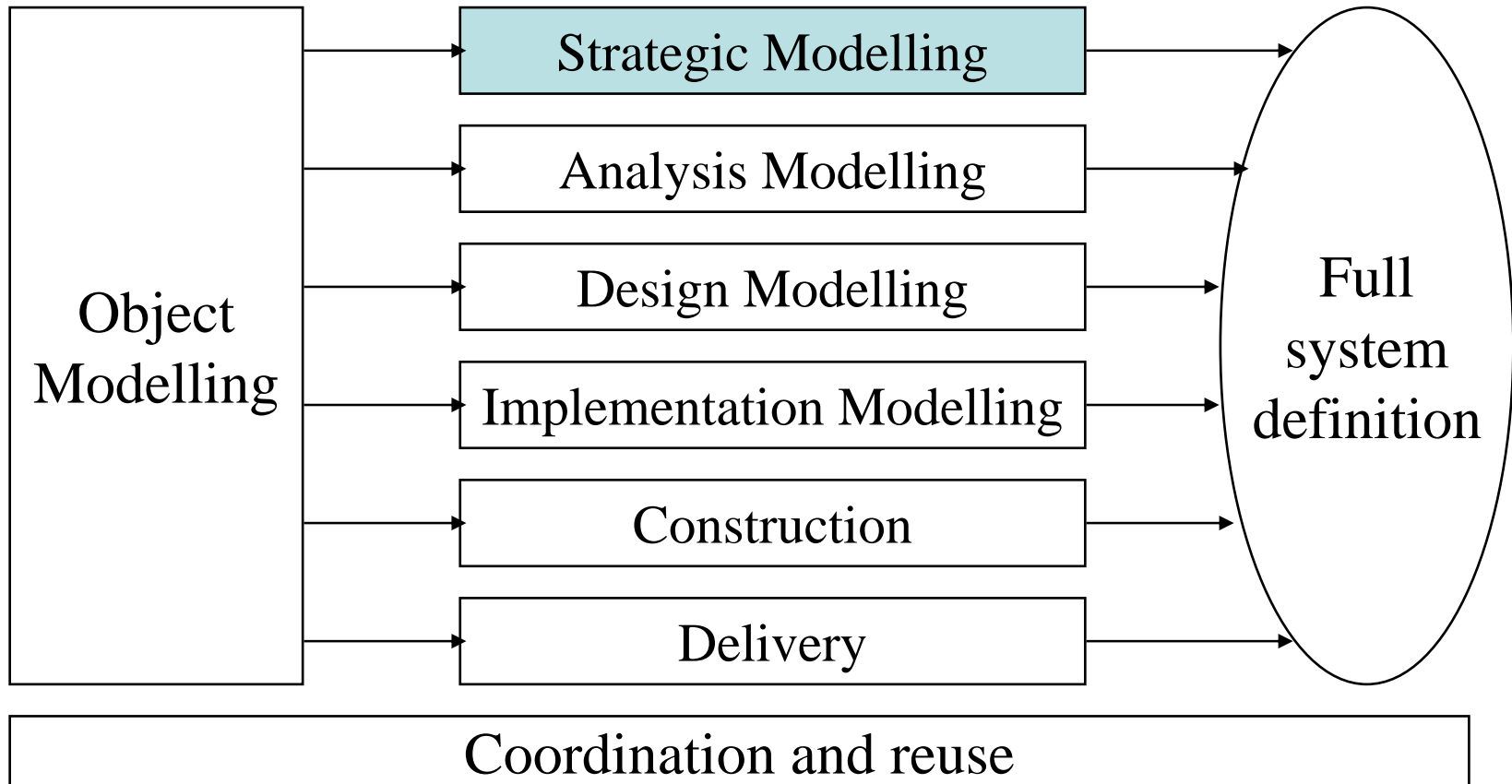


- Process modelling
- Strategic and business models
- Desirable model features (or requirements for our model).
- Role Models
- Role Activity Diagrams
- Examples and Exercises

# The OMG OO Lifecycle



## Lifecycle

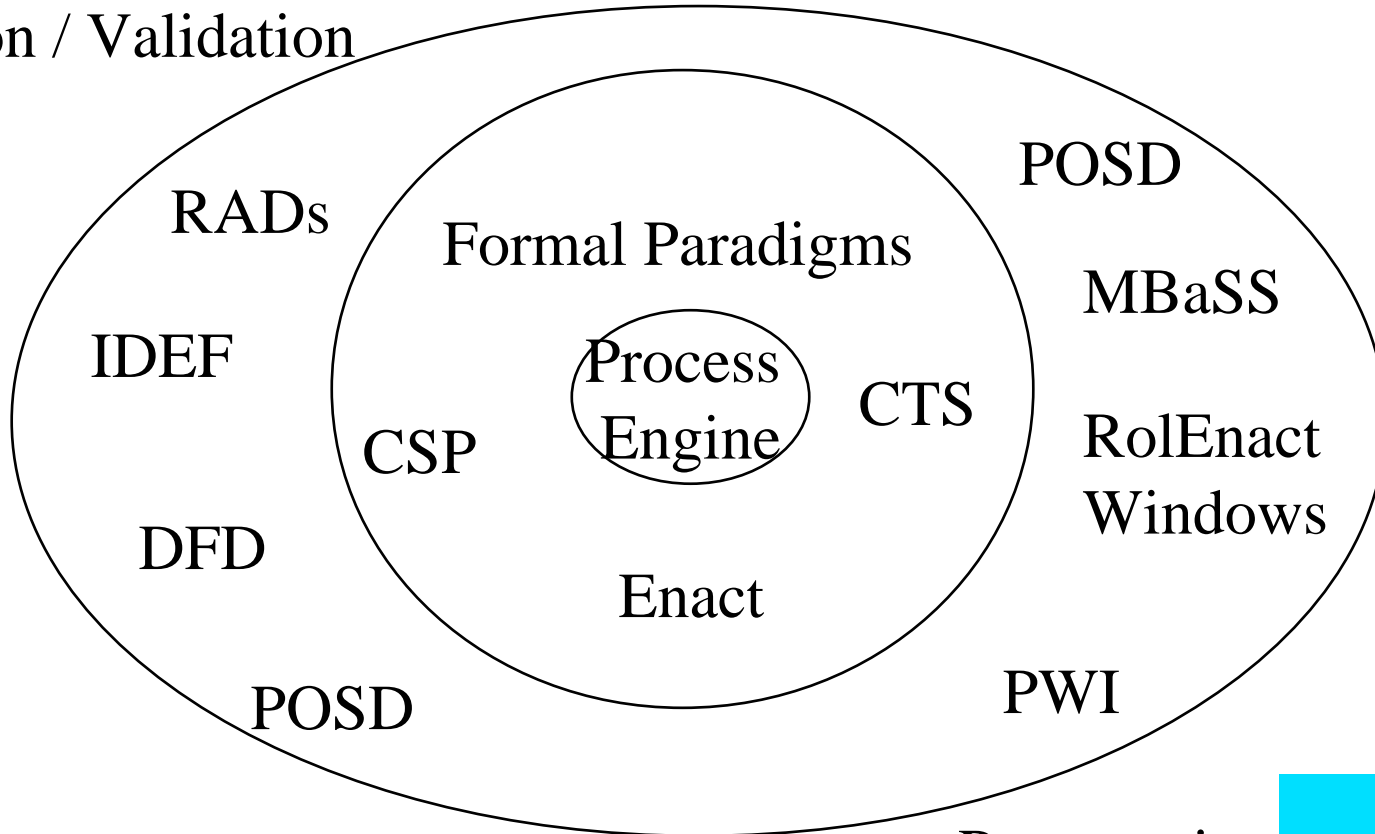


- Models of the Software Development Process.
  - Key ideas: Process programming, process maturity, study of people, tools, measurement, notations.
- Models of Client (business) processes.
  - Business process modelling.
  - Strategic modelling, requirements and analysis.
  - Alignment of IT with strategy and operation.
- Software development as a business process.

- Start with an easy to understand approach (diagrams).
- Even simple diagrammatic notations lead to complex models
  - Which users find difficult to comprehend.
  - Mechanisms to add structure to detailed models can help.
- Enactable notations do help
  - but users need to be shielded from them.

# Layers and Validation

Elicitation / Validation



Presentation  
Enaction /



- Mostly concerned with Capture.
  - However, the notation should allow us to discuss, validate, and experiment with process changes. (Hence, capability to present to users, and to carry out some qualitative analysis).
- What makes a good model?
- What are our constraints?

- Primitives correspond to features found in the *real-world*.
- Models *essential* properties.
- Intuitive & appropriate.
- Accessible by all model users. (Audience).
- Sound underlying *semantics*.
- Model can handle complexity.

- Goals - Those things the organisation / business is trying to achieve.
- Business rules - Limits (constraints) the business places on people.
  - Legacy: constraints of existing working (not under consideration for change).
- What people *do* (their actions) to achieve goals.
- How people *interact* (work collaboratively).
- *Of course, as with objects, similarity to the 'real world' is a dangerous argument.*



## Manifest themselves in:

- Responsibilities.
- Delegation of tasks and authority.
- *Ordering of activities, pre conditions, pre-actions and post conditions.*
- Criteria for decisions.
- Reporting on progress and giving approval.

# Typical actions

- produce
- distribute
- find
- check
- send
- receive
- identify
- create
- wait
- monitor
- choose
- write
- organise
- plan

# Typical interactions

- agree
- approve
- delegate work (task)
- pass (information, work).
- give authority for
- report on status
- wait (for something)

Interactions are  
points of  
synchronisation.

# Don't need?

- Activities where people act completely on their own.
- Mechanisms to support the process (e.g., to support interaction we use e-mail).
- Non-essential data.
- *Of course all modelling is abstraction.*
- *The trick is to decide what to throw away.*

- Need to consider mapping to Spec and Design, specifically to UML models.
  - A use case or class description.
  - Aid identification of some business objects.
- Need to use terms acceptable to the business audience.
- Need to have fairly rigorous semantics.

# What do process actors need to know?

‘For an individual (or group) in the organisation to carry out their activities, they need to know what activities they must take part in, in what order those activities must take place, what other individuals or groups they must interact with, and which actions are dependent upon those interactions’.

Handy, C. (1976), ‘Understanding Organisations’, Penguin.

- ‘Role based models satisfy these requirements by grouping activities into ‘roles’, which describe the desired behaviour of individual groups, or systems’.

Ould, M.A. (1992), An introduction to Process Modelling using RADs, in IOPTCLUB Practical Process Modelling, Mountbatten Hotel, Monmouth Street, Covent Garden, London.

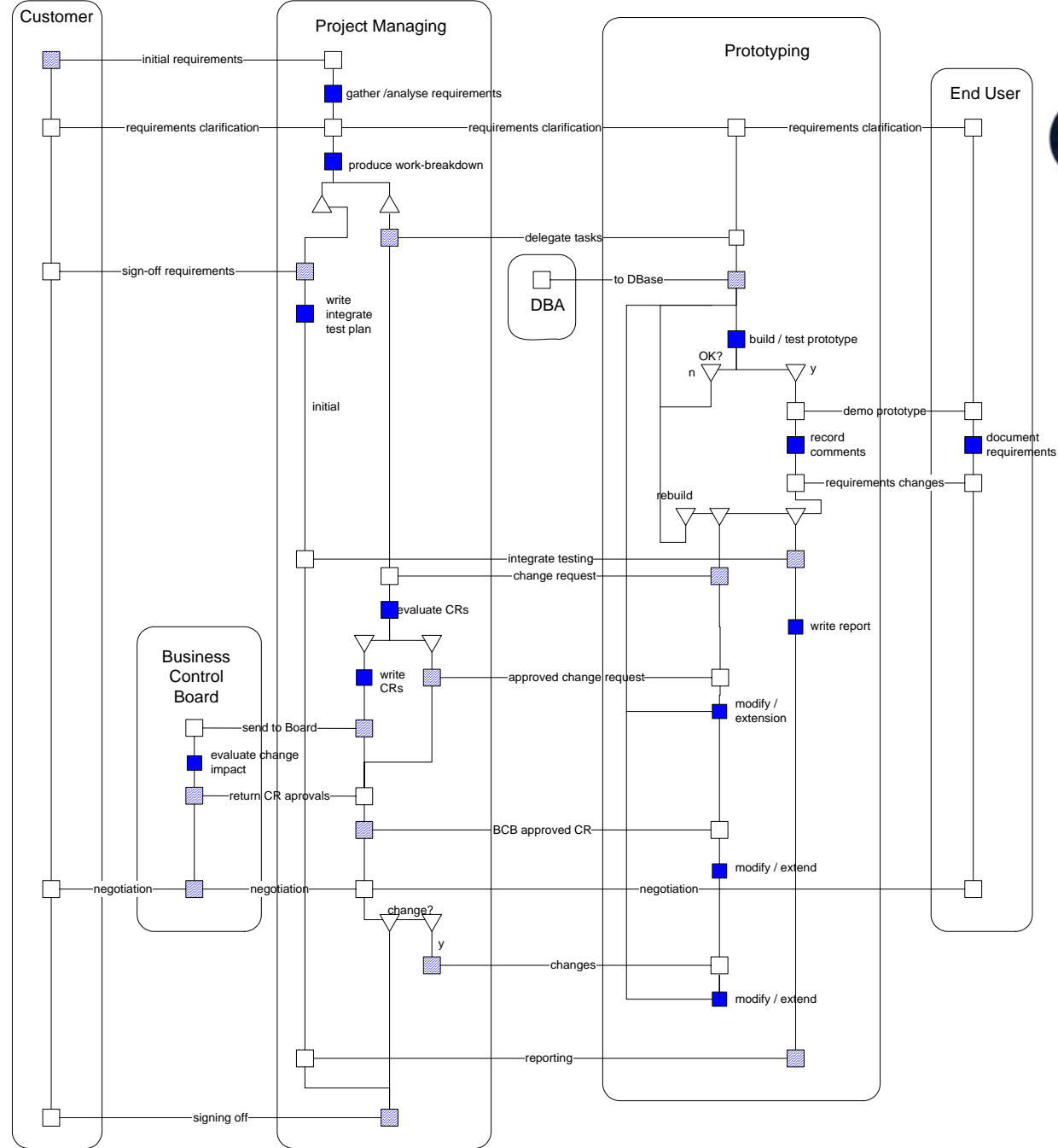
- ‘A role involves a set of activities which, taken together, carry out a particular responsibility or set of responsibilities’.

Ould, M.A. (1995), Business Processes modelling and Analysis for Re-engineering and Improvement, John Wiley & Sons.

- Other top contender for process description is a procedural model. But:
- Difficult to abstract away from the detail (mechanism) of current process. This inhibits change.
- Models tend to have a decomposition related to function, hence activities which are to be carried out by individuals are often spread around the model.



- Original paper Ould & Roberts (1986).
- Formal semantics. Similar to Petri Nets. Can be mapped to other formal notations.
- Widely used. Promoted by Praxis (Ould, Huckvale & others) & Coordination Systems (Roberts).
- Applied to a number of domains, e.g., Software Engineering, finance, Retail and Construction.



- Business depicted in terms of roles.
- Roles are types - e.g., they describe the behaviour of a class of individuals.
- A Role is independent of other roles, but communicates through interactions.
- Instances of roles therefore act in parallel, with the interaction between roles being their only synchronisation mechanism.

## An Example: New projects

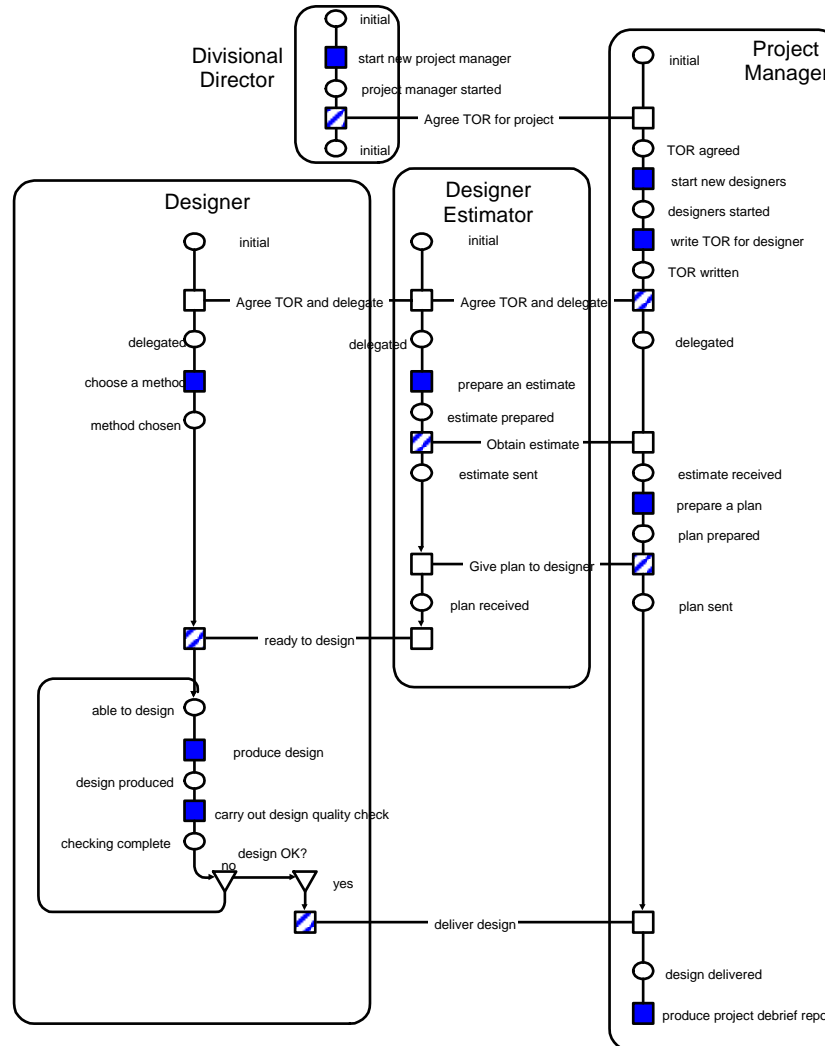
- An external event signifies approval for a new project. As a result the ‘Divisional Director’ is able to start a new project.
- This is followed by the creation of a new ‘Project Manager’. Now the Project Manager and Divisional Director are able to agree a Terms of Reference (TOR) for the project.
- The Project Manager is now able to start a new ‘Designer’. The Project Manager must also write a TOR and then agree this TOR with the Designer.

## Example Continued

- The Designer must ‘prepare an estimate’ and agree the estimate with the Project Manager.
  - (Since this takes place only after the Designer has prepared the estimate the Designer drives the interaction).
- Only then can the Project Manager go ahead and ‘prepare a plan’.
- On completion of the plan the Project Manager gives the plan to the Designer.

- The Designer is ready to design when they have the plan and have chosen the method.
- Having produced a design, the Designer must ‘carry out a design quality check’.
- If the design is not OK, then the designer must go back to design.
- Once the design is acceptable its is given to the Project Manager.
- Finally the Project Manger produces a project debrief report.

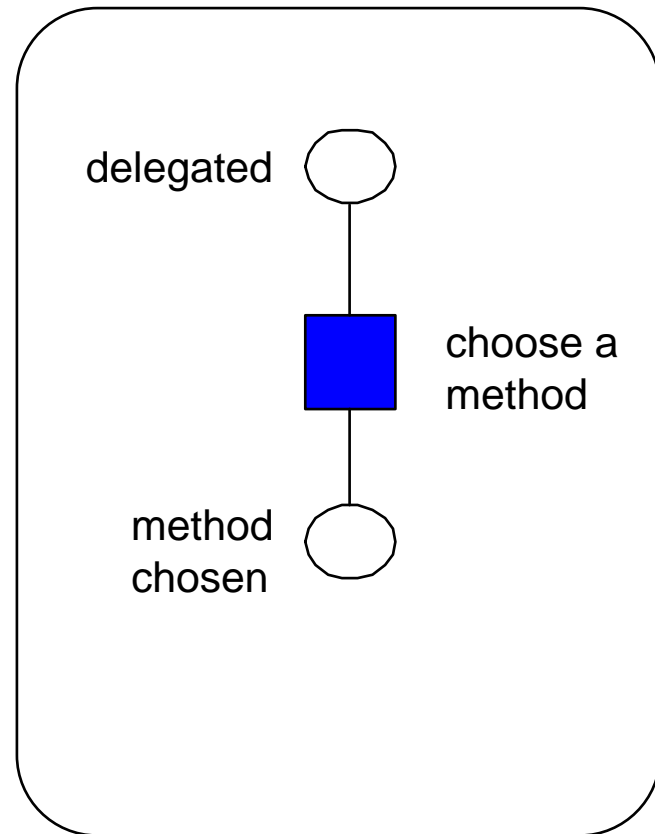
# Adding States



- Roles are:
  - The ‘Designer’, The ‘Project Manager’ and the ‘Divisional Director’.
- What are some of the actions?
- What are some of the interactions?
- Are there any choices?
- Are there any parallel or concurrent threads?



- An action is an activity which the role carries out in isolation.
- Carrying out an action moves the role from its present state to the next state.



- An activity carried out at the same point as another activity (or other activities) in another role (or roles). A shared event.
- The consequence of an interaction is that all of the roles involved move from their current state to their next state.
- Interaction must be initiated by some (driving) role.
- Interactions are synchronous.

- Are interactions in the real world synchronous?
- Are interactions in the real asynchronous?
  - Conceptually some are, though many hidden interactions. Also depends on perspective.
  - As an example (email).
- If interactions are often asynchronous, then why would we have a synchronous model?
- How can we use synchronous models to model both sorts of interactions?

# Types of Interaction

- Demonstration of synchronous and asynchronous interactions.
- We used a buffer.
- In process modelling, one often finds many people who seem to act as buffers!

- Iteration is where a state may be revisited.  
Shown by:
  - Drawing a loop back to a previous point on the role.
  - Having the post-state of an action as a previously named state.
- Typically used when there is some checking or control mechanism to be modelled.

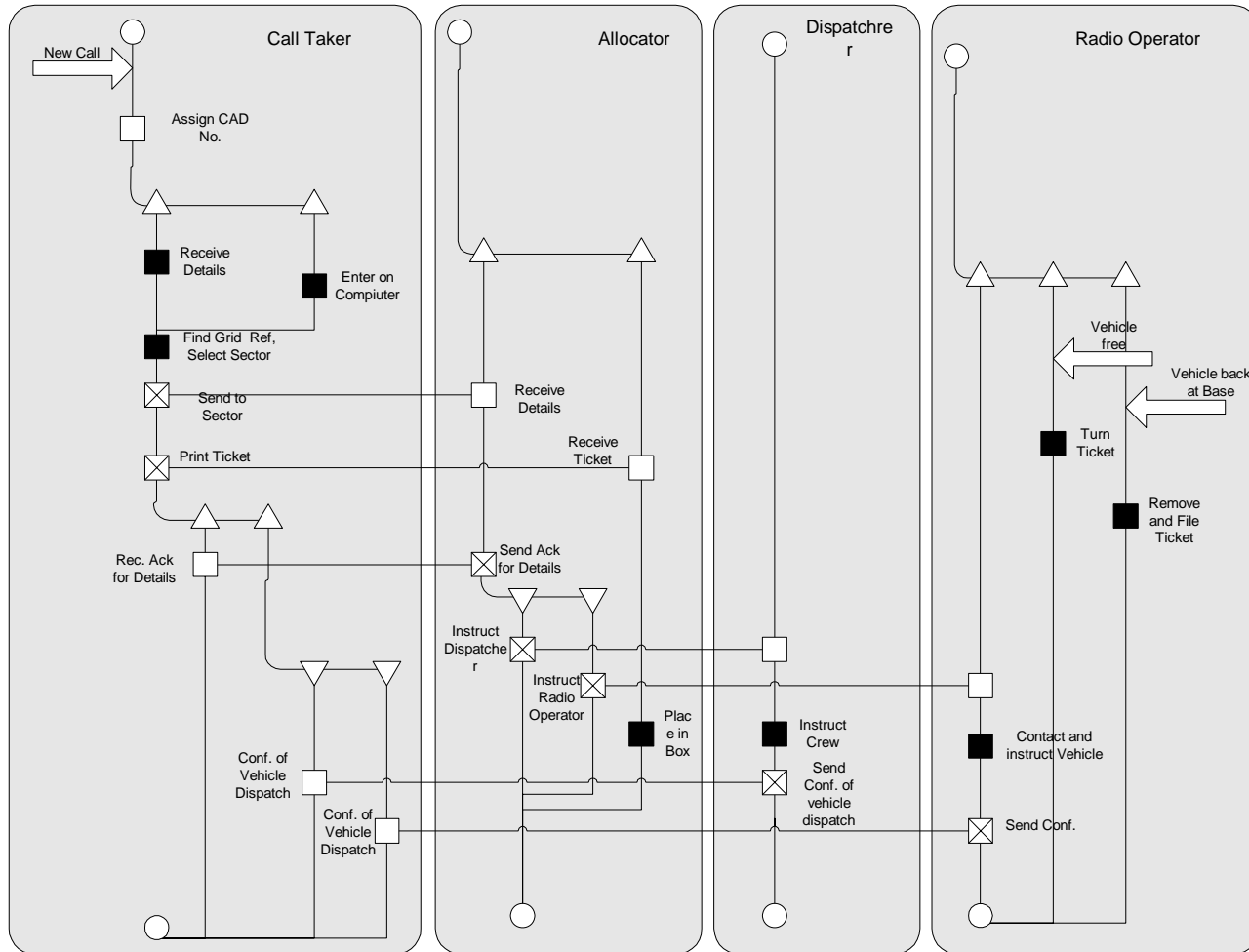
- Thread of control in a role need not proceed sequentially.
- Choice or case-refinement. There may be any number of alternative threads but only one of the threads (or cases) may be chosen.
- Concurrent threads or part-refinement. Each thread represents part of the path. The threads all join together again after the split denoting that all paths have been completed.

- Not required to explicitly label the states of a role, though some authors prefer to do so.
- Labelling states (with circles or ellipses) helps the semantics of the role become clearer.
  - Labels make explicit the pre-conditions, pre-actions and consequences (post-conditions) of each activity.
  - Sometimes need to separate parallel threads into separate (or main and sub) roles..
- Diagram becomes larger and this may hamper understanding.

- Given a RAD of a business process.
- Produce a textual description.
  - Note roles, actions & interactions.
  - Consider whether states act as pre or post conditions. Describe this too.
  - List any problems or errors within the RAD .
  - List any assumptions you have made.
- *I know: It's the reverse of what you expect.*



# To Describe



- Given a description of a business process.
- Produce a Role Activity Diagram.
  - List roles, actions & interactions.
  - Decide where there are part and case refinements.
  - List any assumptions you have made.
  - Try to describe the essential business process, and not the current implementation detail.
- The Example follows:

# Constructing RADs

- *Taken from Martin Ould's IOPT Presentation of RADs*
- *A day where I “saw the light”, and stopped trying to model everything with DFDs.*
- **The Process Scenario**
- The main steps from receipt of proposal to dispatch of policy papers.
- In the Mail Room the Mail Room Clerk opens incoming mail and enters basic details into the computer, such as customer name, agency reference number, type of business.
- The system automatically allocates a reference number and creates a work-item record for the application. The work-item is automatically routed to one of several Supervisors, depending on the agency reference number.
- A supervisor allocates a work-item to one of their Clerks. They can also allocate the work to themselves if all of their clerks are busy.
- The Supervisor may check the workload of the clerk at any time.
- The Clerk enters details from the application (address, age, health-checks,...).
- The Clerk continues processing an application. They check a database to see if there has been any previous business with the Customer. If not, a credit check is requested from an external agency. When the reply is received, the Clerk accepts or rejects the application. The Clerk may also ask the system to remind them when a reply is overdue.
- The underwriter also assesses the application, either working from a different site or from home. They decide whether a medical examination is needed, and if so, send a letter requesting one. The Underwriter may also ask the system to remind them when a reply is overdue.
- If both the Clerk and the Underwriter accept the application, the Mail Room Clerk dispatches policy documents to the Customer. Otherwise, a rejection is sent.

# Summary

- Process modelling can be used to inform requirements
- Role based models are a useful approach for many scenarios.
- Introduced Role Activity Diagrams
- Simple Exercises
- Still to consider how process models can fit with other RE aspects (in the document).