Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (0223) 323010**
INTERNATIONAL: **+44 223 323010**
FAX: **+44 223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

Dependability Model TC Presentation

Nigel Edwards, Owen Rees

Abstract

These are the slides for the presentation of the work in progress on the dependability model.

This presentation was prepared for the ANSA TC meeting of 1st November 1993.

APM.1080.00.03 **Draft** 28 October 1993

Request for Comments (confidential to ANSA consortium for 2 years)

Distribution:
Supersedes:
Superseded by:

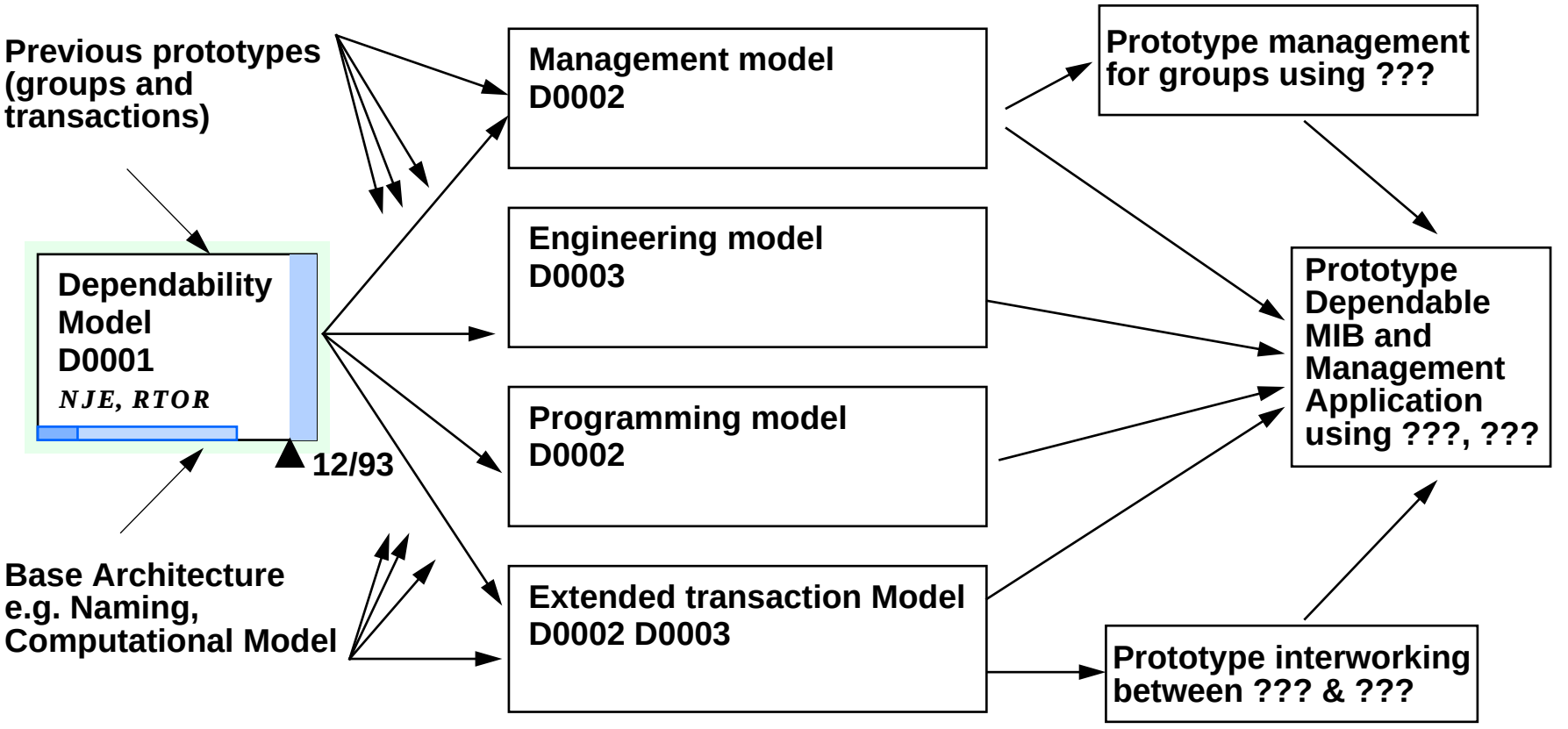
Dependability Model

Work in Progress

Nigel Edwards, Owen Rees

Dependability Group

Dependability outline plan



Objective of this talk

- **The vision we are working towards**
  - **How you can use the failure model in the future**
- **The basic concepts of the failure model**
- **How you can use this model now**
  - **see APM1046**

The ANSA Dependability Vision

Business-critical applications ⇒ Liability ⇒ need dependability

Guaranteed delivery of service to customers; ideal is measurable exposure

Dependability is hard & complex ⇒ Transparencies

Hide as much of this complexity as possible from programmers

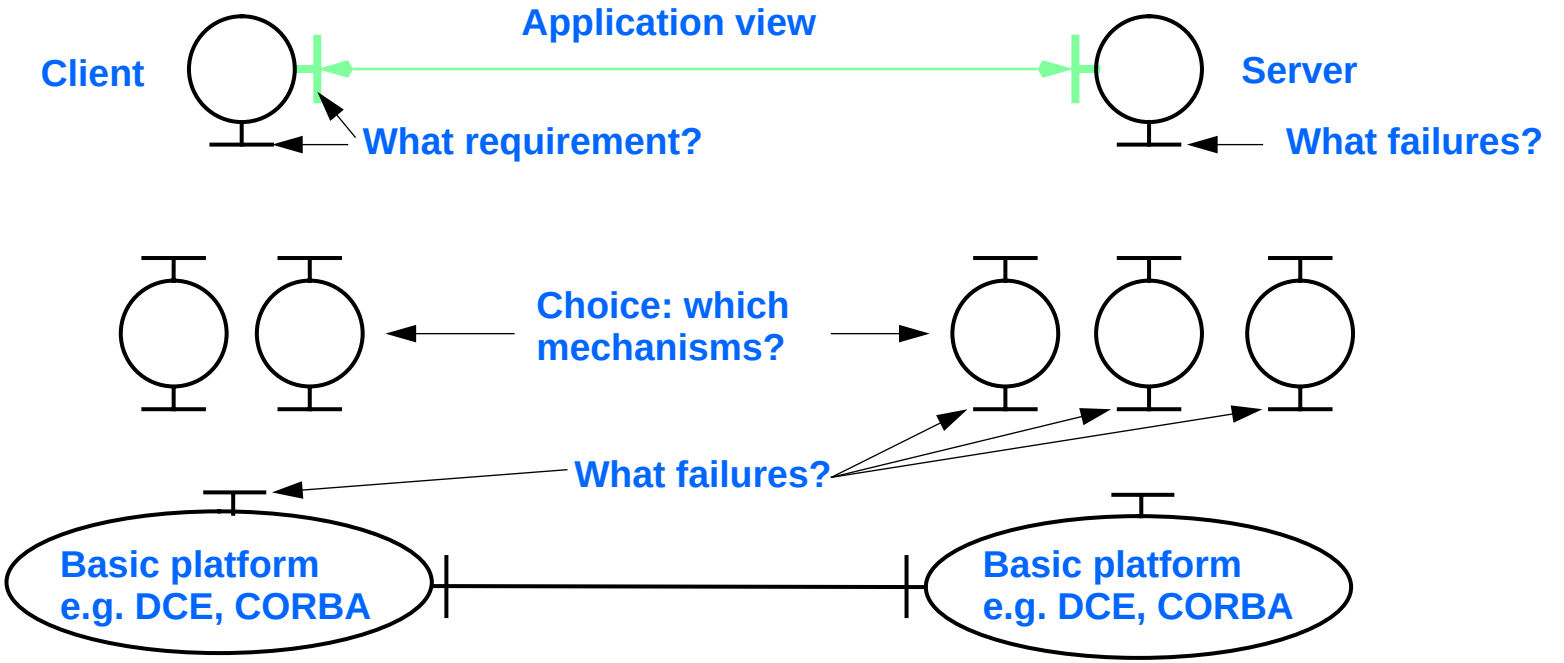
No universal requirements or solution ⇒ Selective transparency

The programming model & advanced transactions provide the programmer with concepts to state requirements & select transparencies

Deploy solutions quickly ⇒ Automated transparency for dependability

automate the configuration of existing engineering components based on the rules, recipes and guidelines which constitute the engineering and management models

How the model fits into the vision

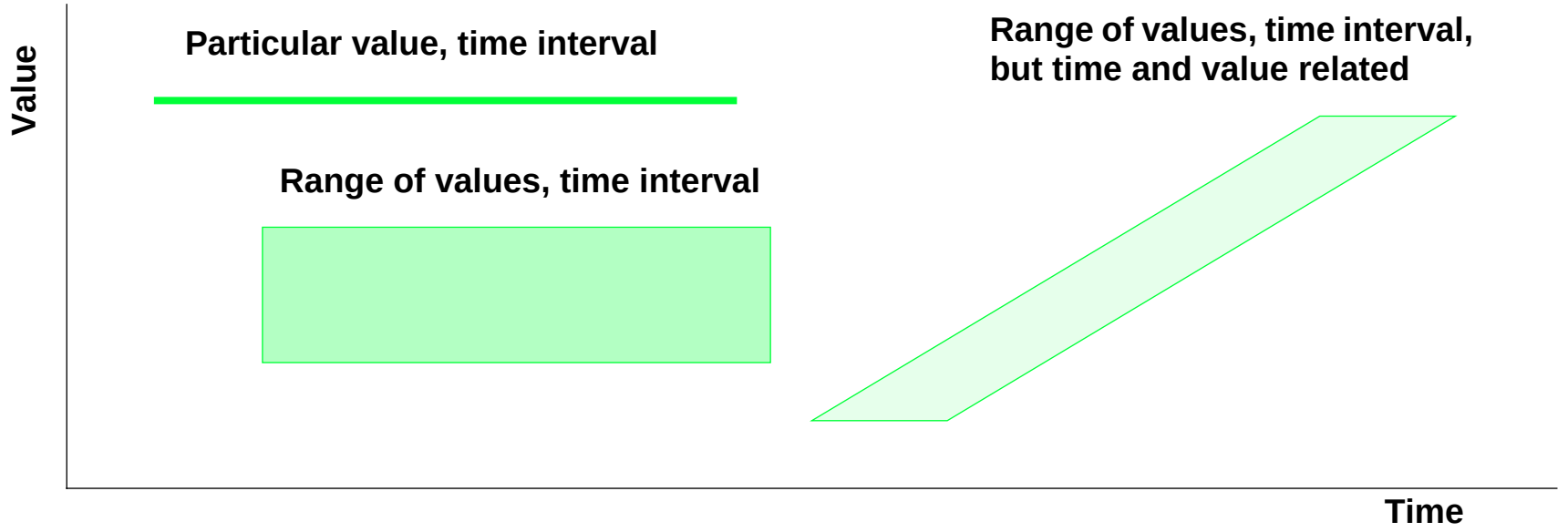


- **Failure model – Language for describing failures**
- **Describe effect of mechanisms – selection criteria**

Overview of the Failure Model

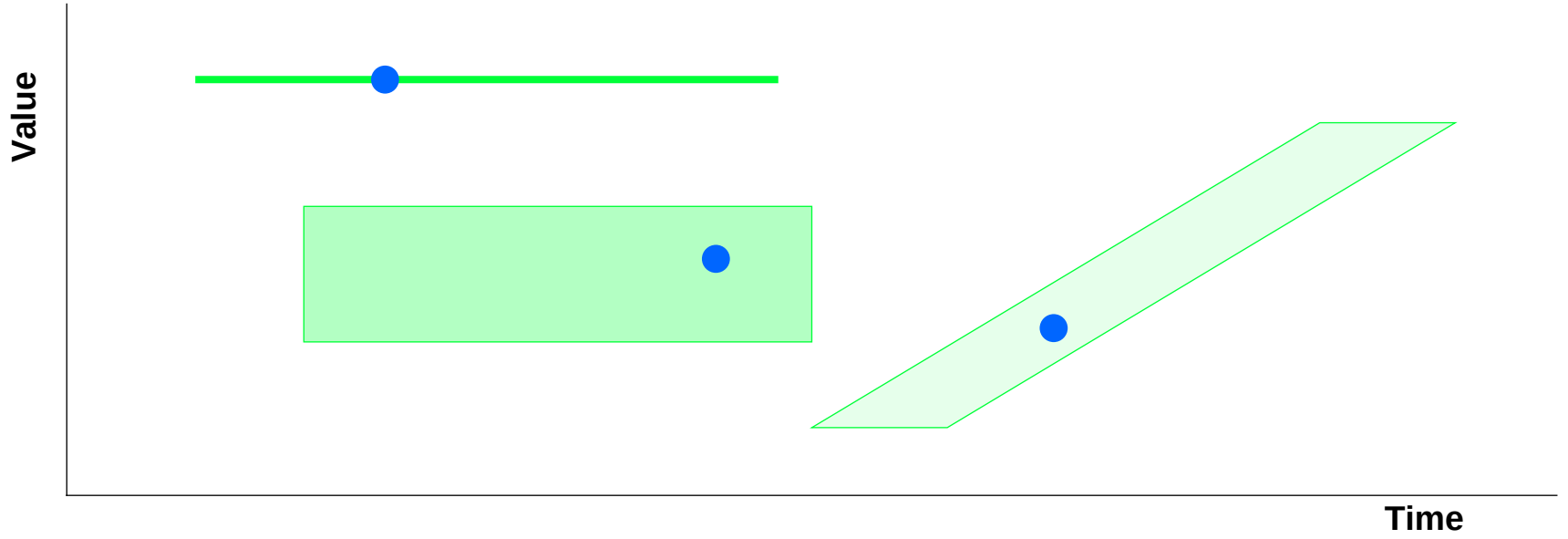
- **Correctness of interaction between objects**
  - **Based on events**
  - **procedure call – potentially remote**
- **Different observers with different expectations**
  - **expectation is observer's view of what is correct**
- **Expectations and occurrences**
  - **Graphical representation**
  - **Set based model used when more rigor is needed**
- **Correctness: occurrence, or non-occurrence, matches expectation**
- **Failure: occurrence, or non-occurrence, does not match expectation**

Expectations



- **Structured values**
  - **map to single value**
  - **slice or project onto single value**

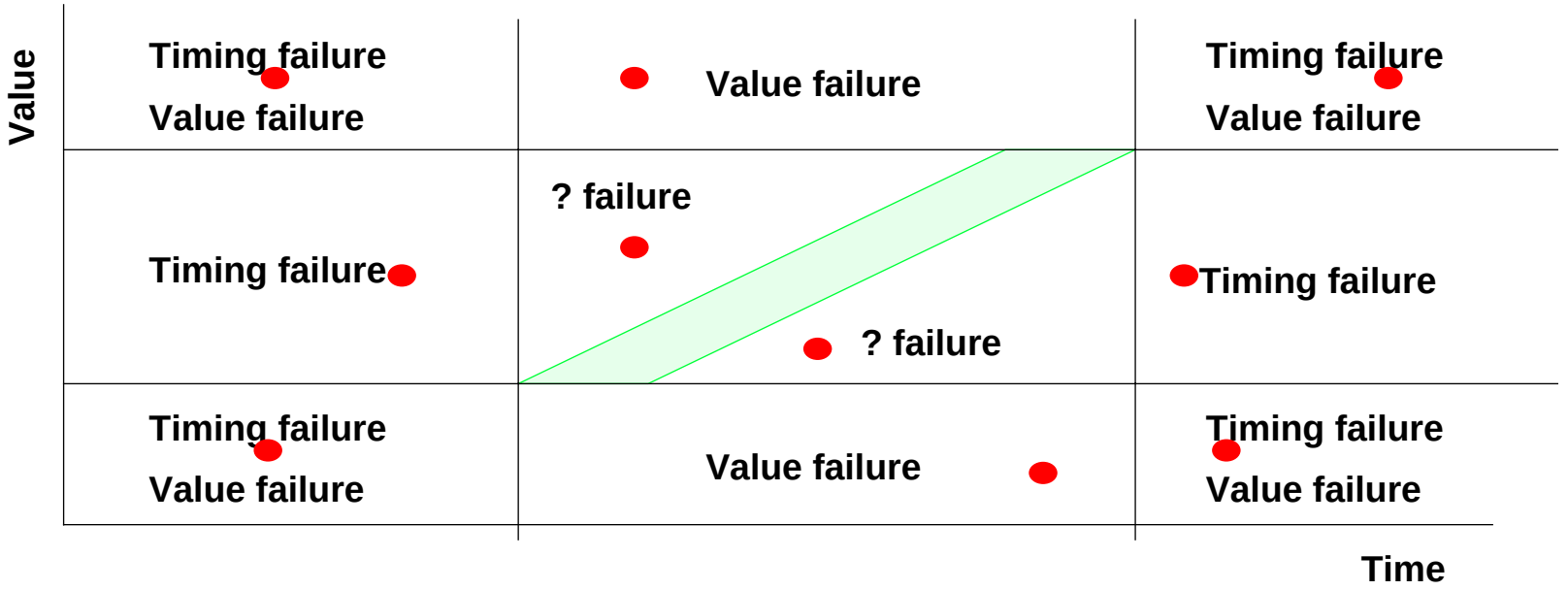
Correct cases



- **Correct occurrence – something happens and it was expected**
- **Correct non-occurrence – nothing expected, nothing happens**

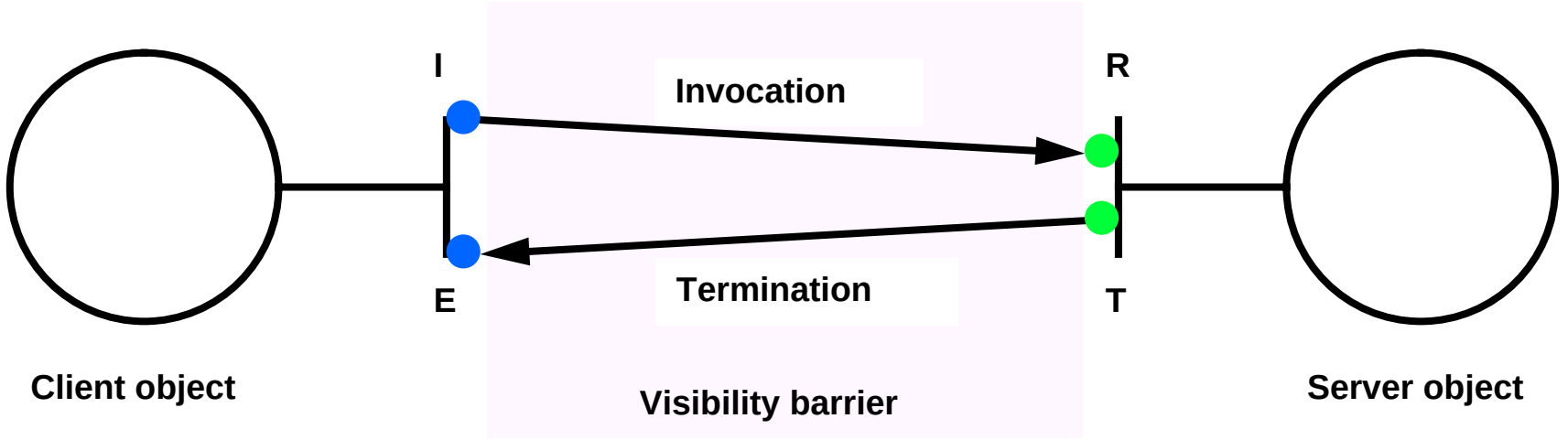
Failure cases

- **Something expected, something happens: incorrect occurrence**



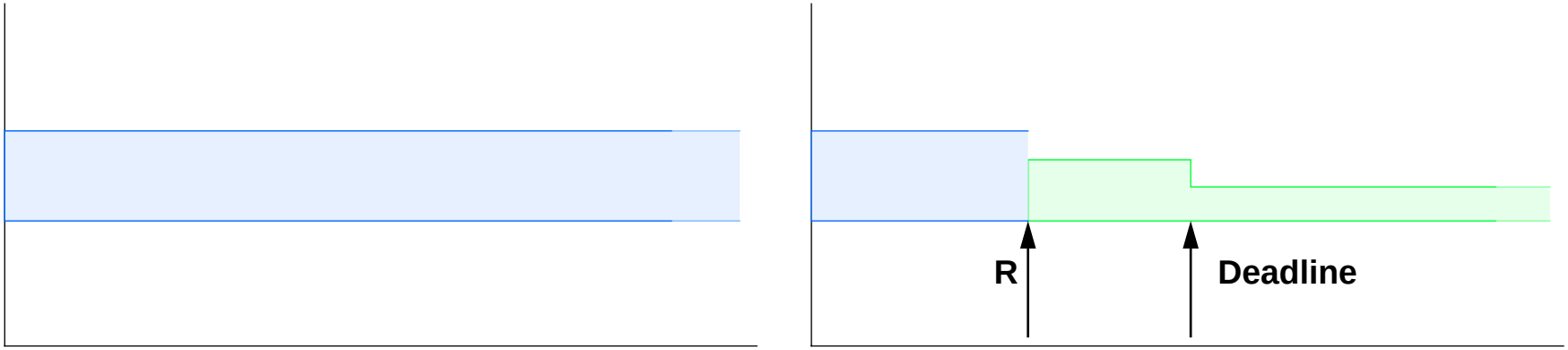
- **Omission failure – something expected, nothing happens**
- **Unexpected occurrence – nothing expected, something happens**

Interaction Model



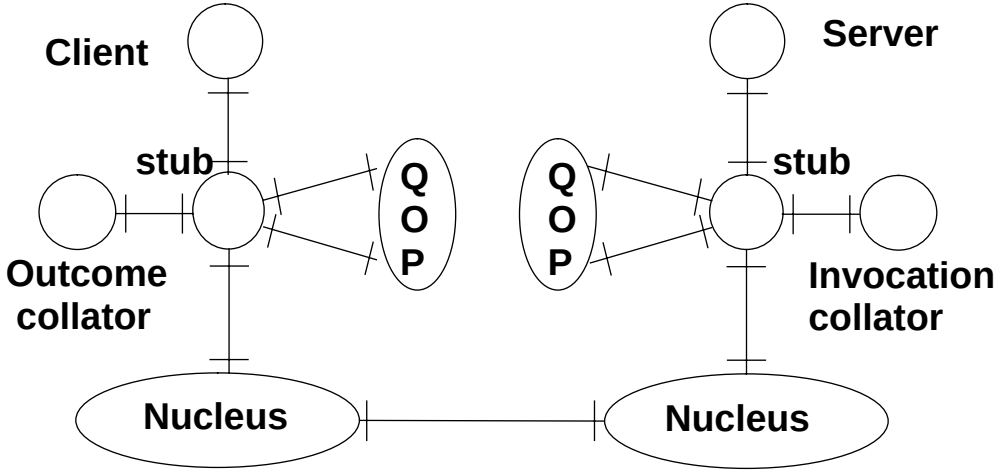
- Client expectation – of Initiate and End
- Server expectation – of Request and Terminate
- Computational model expectation – v(I) = v(R) & v(T) = v(E), causal order
- Engineering expected to deliver this – requirement for mechanism

A Servers Expectation



- **Typically no deadline for client to call**
- **Expectation changes when request R observed**
  - **This is the expectation of the engineering supporting a server**
  - **Request from group member – other member requests expected before deadline**
  - **Not expecting retransmission with different value**

– Using the model to analyse GEX (APM1046)



- Identify pairwise expectations of engineering objects
- Identify application expectations of client and server
- expectations given the structure
- analyse propagation for each failure mode of server against client expectation
- analyse for client failures and server expectation
- repeat for QOP interaction

Benefits of the analysis of GEX

- **We now understand what kinds of failures can be detected and what kinds of failures can be tolerated — this was absent from AR2**

- **Gained considerable insight into how to improve the engineering**
  - **More failures can be detected than tolerated**
  - **Better exploitation of detection capability**

Work in progress on “Dependability Requirements”

- **APM1062: “Building Dependable Systems”**
  - **Defines basic concepts: reliability, availability, safety etc.**
  - **Describes some of the important problems: Fault propagation, Fault assignment**
  - **The basic techniques for dependability: avoidance and tolerance**
  - **Dependability Management: configuring, measuring and evaluating dependability**
  - **The role of the work on a programming model, advanced transactions and an engineering model**

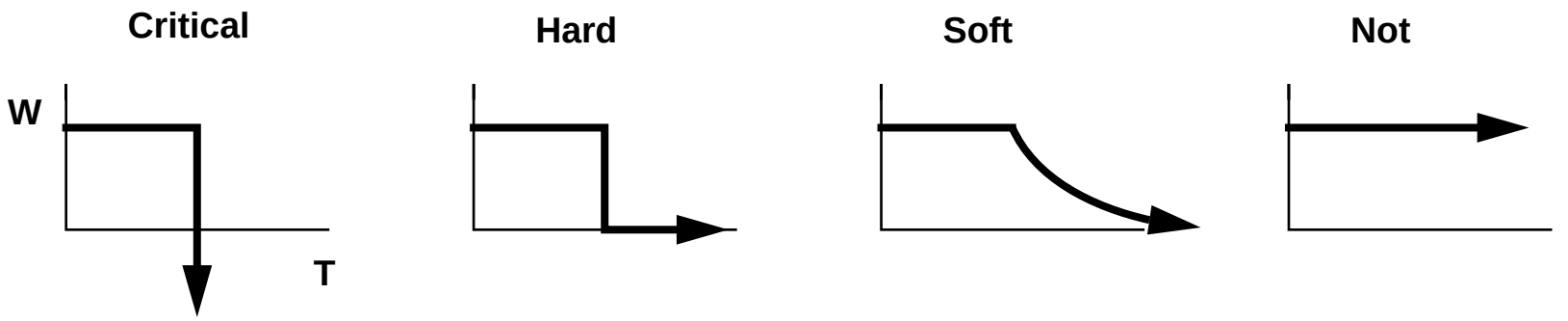
Future Directions

- **Use the model to analyse other work**
  - **Other ANSA work – e.g. the Atomic Activity Model**
  - **Any other work proposed for inclusion in ANSA**

- **Further developments of the model**
  - **assigning “worth” to occurrences**

Adding “worth” – an extra dimension

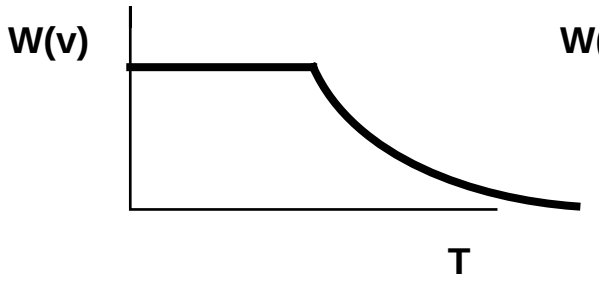
Graphical taxonomy of real-time [Stankovic, PDCS2 1st Open Workshop]



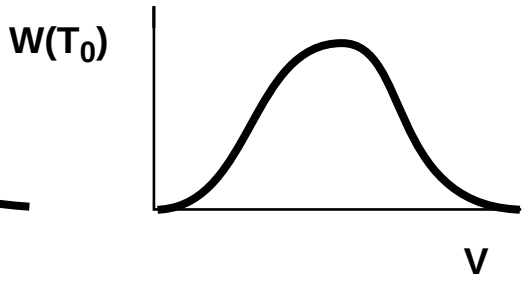
- **What is it worth to get the answer at time t?**
- **Extend failure model to include this idea**
  - **assign a ‘worth’ to each value, time pair**
  - **(v,t) expected ⇒ worth(v,t)>0**

Extended Model - example

Slices through Worth×Value×Time space

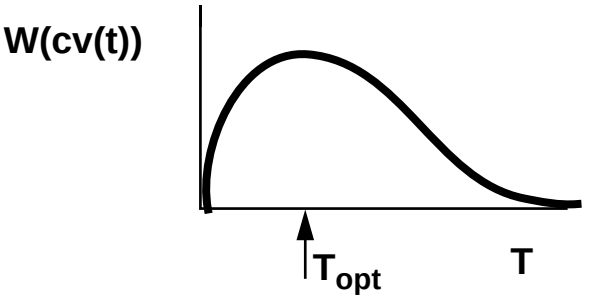


Worth of a value decays



Worth increases with accuracy

Computation 'surface' slice



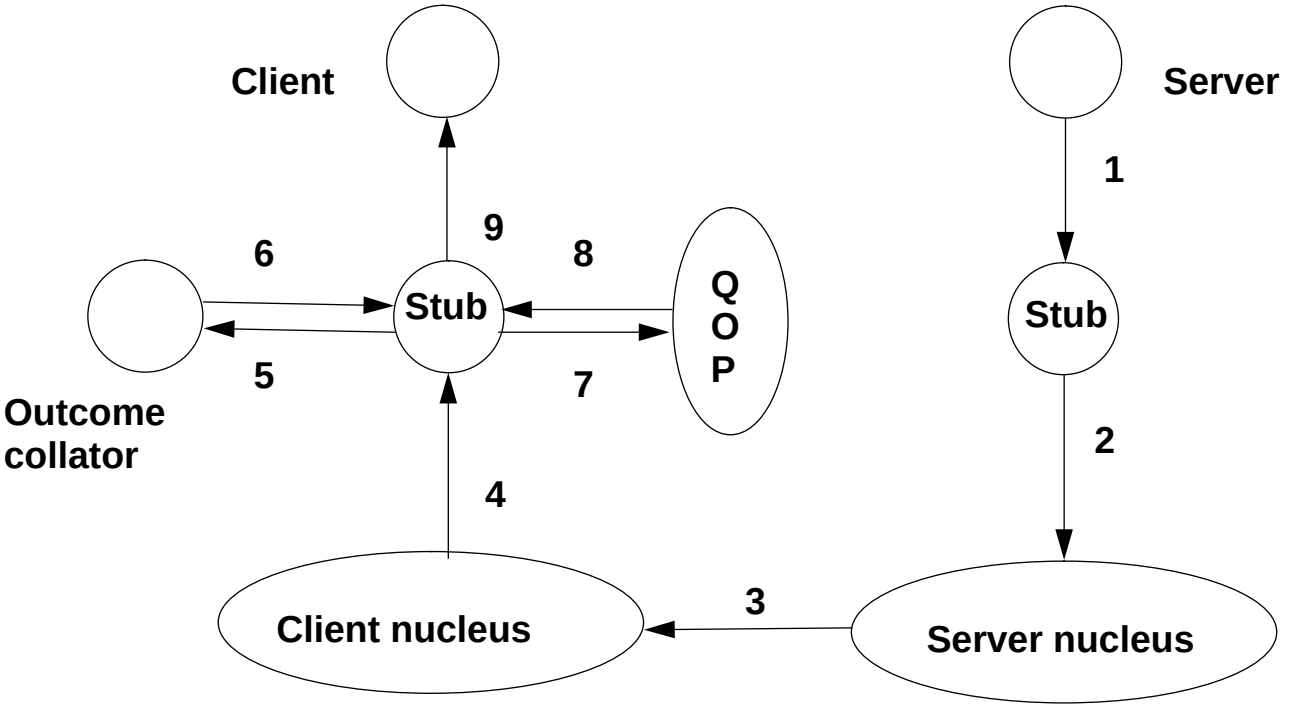
Optimum compromise

- **Suppose it takes longer to compute a more accurate answer**
- **Value computed at time t is a 'surface' in the Worth×Value×Time space**
- **Maximum worth under these constraints**

Summary

- **Vision**
  - **tools to help build open dependable distributed systems**
  - **selective transparency with selection criteria**
  - **rapid deployment**
- **Model – applied now to an application problem**
  - **better understanding of the requirement**
  - **better understanding of the effect of transparency mechanisms**
  - **choice of transparency mechanism based on that understanding**
- **Model – applied to infrastructure**
  - **improved transparency mechanisms**
  - **improved tools for configuring transparency mechanisms**
  - **realising the vision**

Propagation analysis for server failure



- Response follows numbered path; failure propagation route
- Each object is possible detector; which failures could it detect?