Programming Models Session
1-slide summary

- General acceptance of the value of SOA – but important not to see it as a panacea
- Balance between formal techniques and engineering judgment
- Formal techniques are often inadequate for the “Global Information Utility”
  - Some interesting formal approaches for different styles of interaction
    - Session-based calculi, Committed negotiation, Service invocation as constraint solving, Constraint programming
- Need for engineering judgment in building real systems
  - “Islands” of formality where behaviour is well-understood/ conformance is guaranteed
  - Possible role for Aspect Oriented Programming in composition
  - Move to a smaller number of skilled designers, supported by tools/compilers
- Need different levels of programming, with appropriate features for different users
  - Concerns separated so programmers deal only with decisions needed in a single layer
  - Strong tool support and pre-built skeletons/patterns for common application types
    - modular, reusable and optimised