

What I have Learned this Morning: A Jigsaw of Pieces?

Gordon S. Blair
(*gordon@comp.lancs.ac.uk*)



This is a Very Hard Problem

Difficult to solve in the general case

Most successful approaches narrow the scope

- By focusing on a given domain
 - E.G. user interfaces [Sottet]
 - Role of *domain specific languages*?
- By focusing on a given development methodology
 - Assume a component approach such a fractal [Dubus]
 - Assume an AOP approach such as hyper-slices [Chitchyan]
 - Or indeed a hybrid model!



The Reality of Distributed Systems

You cannot ignore the hard issues relating to:

- Scalability
- Extensibility
- Performance
- Etc!



Importance of the SE Process

Software engineering methodologies for adaptive and autonomic systems [Cheng, Normand]

Systems architectures for adaptive and autonomic systems

Role of models throughout

- Traceability
- Verification and validation
- Etc.



Towards models@run.time

Need to move from models *for* run-time (or for adaptation) to models@run-time

- This needs work on:
 - Models
 - Systems
 - The relationship between them
[Bezevin, Boskovic]



This is really a workshop about reflection!

Defines the relationship between the model and the system [Costa]

- Maintains the causally connected *self-representation*
- Lots of experience in defining appropriate (canonical) systems level models

N.B.

◆ Reflective systems are by definition recursive

- From models to meta-models and beyond



Shaping the Discussion

Key Questions

- What should a run-time model look like?
- How can the models be maintained at run-time?
- What is their role in system validation?
- What are the best overall model-driven approaches for adaptive and autonomous systems?

Meta-questions

- What do we know (useful building blocks)?
- What do we not know (towards a roadmap)?
- And of course, what should we do next!

