An Infrastructure for Adaptive Systems Development

Kevin Lee, Norman W. Paton, Alvaro A. A. Fernandes, Rizos Sakellariou
School of Computer Science, University of Manchester
Oxford Road, Manchester, M13 9PL, U.K.
{klee, rizos, norm, alvaro}@cs.man.ac.uk

Jim Smith, Paul Watson
School of Computing Science, Newcastle University
Claremont Road, Newcastle upon Tyne, NE1 7RU, U.K.
{Jim.Smith, Paul.Watson}@ncl.ac.uk

Objectives

- Create an infrastructure to support the Systematic Development of Adaptive Systems
- Ease the development of adaptive systems.
- Support the development of better adaptive systems
- Investigate the use of the infrastructure in a number of different domains
- Use the infrastructure to improve the general understanding of adaptive systems

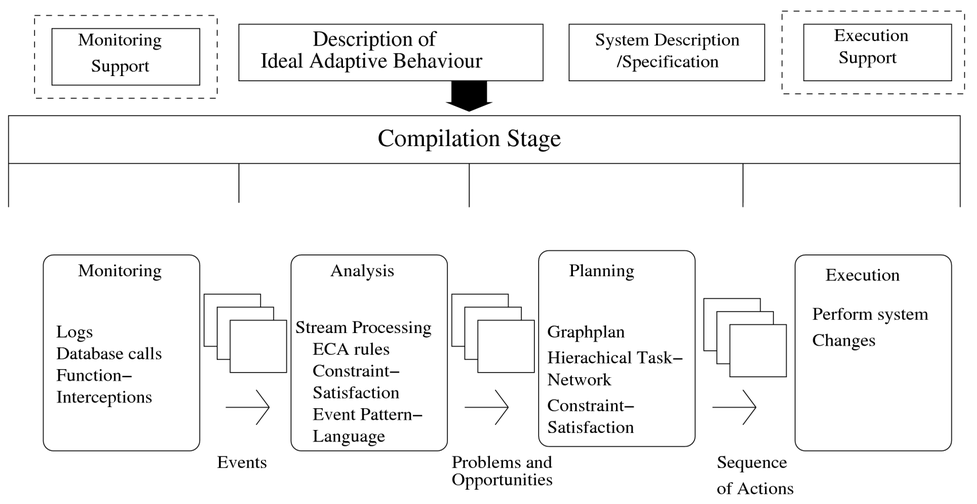
Foundation

- Many systems nowadays face such complex and varying demands that a static configuration is very likely to be wasteful, through over-provisioning of resources.
- Building adaptive systems is hard, and has often been done in an ad-hoc way for a specific purpose.
- The inevitably brittle result from such an approach gives strong motivation for seeking more generic ways to build such systems.
- To this end,adaptive systems are often seen in the Autonomic Systems Community as functionally decomposable into the components:
  - **M**onitor **A**nalyse **P**lan **E**xecute

Methodology

- Implementation of MAPE as loosely coupled publish-subscribe components to allow flexible composition of adaptive systems
- Definition of managed element interfaces for Monitoring and Execution
- Existing technologies can contribute:
  - **Monitoring**: Distributed Monitoring Packages, Log parsing, Event streams
  - **Analysis**: Stream processing, Constraint Satisfaction, Composite Temporal Events
  - **Planning**: Automated planning, Scheduling
  - **Execution**: Deployment protocols
- Using high-level declarative descriptions of adaptive behaviour to enable automated generation of the adaptive infrastructure

Adaptive Infrastructure



Case Studies

Distributed Query Processing

Description

DQP provides many opportunities where adaptation could be beneficial including dealing with load imbalance and bottlenecks. Adaptations include swapping operators and increasing operator parallelism.

Monitoring	Analysis	Planning	Execution
Operator throughput	Look for: Load Imbalance Bottlenecks	Alter: operator Parallelism Data assignment Operator selection	Modify plan, or deploy new plan and restart

Gounaris et al: ICDE 2006, Eurviriyanukul et al: EDBT 2006

Business Workflows

Description

Commercial workflows often run in the context of Service Level Agreements (SLAs), detailing non-functional requirements. Dynamic resource management at multiple levels can benefit both provider and user.

Monitoring	Analysis	Planning	Execution
Various metrics, e.g. throughput, response time, monetary cost incurred	Look for: burst in request flow, budget overrun	Alter: allocation of resources to SLAs, selection between alternate services	Modify invocation mapping and deploy services

DAG Scheduling

Description

Directed Acyclic graphs (DAGs) are often used to organise long-running and CPU intensive computation for execution on distributed clusters. Dynamically adapting to changes in resource availability can reduce the overall execution time.

Monitoring	Analysis	Planning	Execution
Various relevant metrics, e.g. task progress, execution node status, environment status.	Look for: Execution node failure, additional execution capacity, failure to meet QoS time, bottlenecks	Alter: move task to alternate execution node, alter resource allocation, alter task parallelism, change data sources, recalculate task schedule	Modify task-execution schedule, move tasks.