Tolerating Architectural Mismatches

Rogério de Lemos
University of Kent at Canterbury, UK

Cristina Gacek, Alexander Romanovsky
University of Newcastle upon Tyne, UK

Motivation

System built from existing components (complex glue-code).

Software architectures describe the structure of systems: components, connectors and configurations.

Architectural mismatches: assumptions on the services provided and required do not match.

Analysis and removal. But impossible to localise and correct all architectural mismatches statically.

Motivation

Dependability is a system property.

Faults can cause errors. Errors can cause failures.

Fault tolerance a means to achieve dependability:

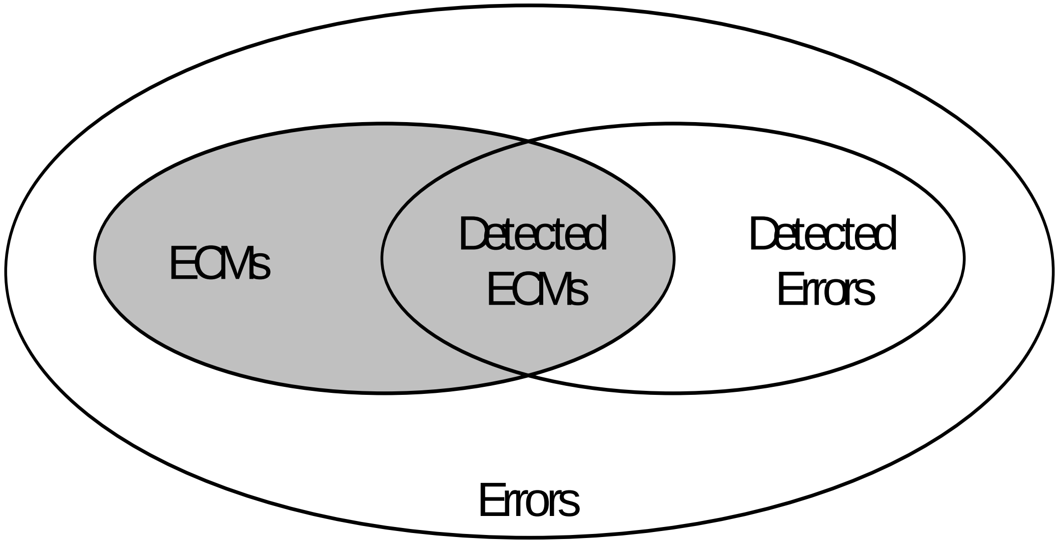
- provision of service in spite of faults;
- error detection, error recovery and fault treatment.

Architectural mismatches are “design faults” at the level of integrated systems.

Architectural Mismatches

Errors caused by architectural mismatches (ECMs):

- latent or detected;
- can cause system failure when ECMs affect the system service.



Mismatch Tolerance

Mismatch prevention, removal, tolerance.

In tolerating mismatches there are two abstraction levels:

- architectural level where the mismatches are introduced;
- execution level where the ECMs are detected and recovered from.

Redundancy (e.g. additional information, time) is needed to detect an ECM, to associate an ECM with a mismatch (cf fault diagnosis) and to tolerate it.

Examples

Integration of two complex large-grain (COTS) components: C1 and C2. Backtracking-related architectural mismatch: C1 backtracks but C2 does not.

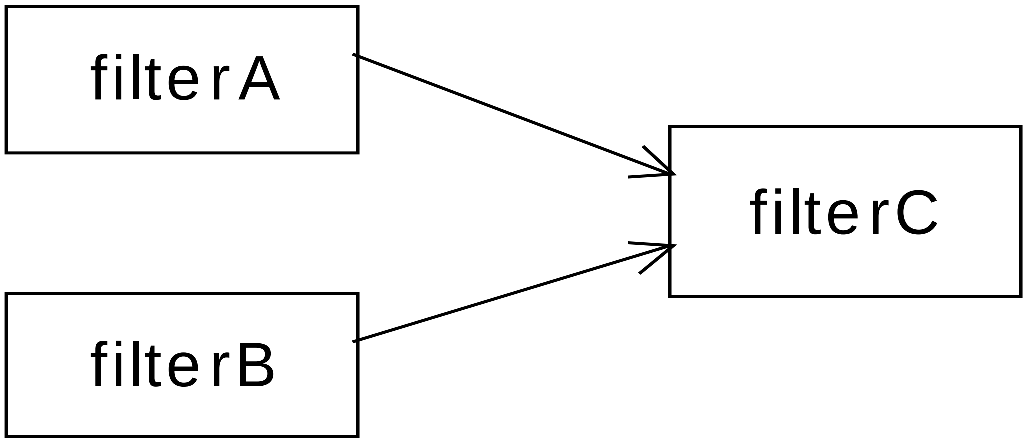
Detection of ECM: need additional information (observer, reflection, additional channel, etc.). It can be at the style level.

Recover from the ECM: depending on the direction of information two types of buffering can be employed, or an application-level recovery can be used.

Examples

Architectural mismatch: call to a non-re-entrant component.

In the pipe-filter style filters are non-re-entrant. filterC is unable to deal correctly with data from two sources.



Examples

Tolerating this mismatch by extending the style.

Detect the ECM:

- incoming port of filterC has to be made aware of more than one connector.

Recover from the ECM:

- queue all the incoming material until the first connection is over. Dealing with one connection at a time.

Future Work

- Using an existing ADL for describing architectures and for introducing mismatch tolerance.
- Developing typical (re-usable) techniques for tolerating typical mismatches.
- Refining existing styles to come up with a set of mismatch tolerance styles (incorporating ECM detection and recovery).
- Dealing with mismatch tolerance artefacts through several phases of software development.
- Introducing diversity of connectors and components.
- Developing architectures that employ general exception handling for mismatch tolerance.