Monitoring Architectural Properties in Dynamic Component-based Systems

Henry Muccini and
Fabiano Ricci
Computer Science Department
Università degli Studi dell'Aquila
L'Aquila, Italy

Andrea Polini and
Antonia Bertolino
Istituto di Scienza e Tecnologie
dell'Informazione - "A.Faedo"
Pisa, Italy

This work has been partly supported by the EU FP6-2005-IST EU Project PLASTIC and by the national FIRB Project ART DECO

Agenda

- The Fact and the Problem
- Our Context
- Our Proposal: MOSAICO
- Experience
- Future Work

The Fact

- Modern systems are increasingly required to be capable to **evolve at run-time:**
  - Changing Requirements
  - Changing Environments
  - Evolving systems

16:00 - 17:30 Working Session

Dynamically Reconfigurable PLAs

Change Management and Arch. Dynamics

… Adaptively Changing Architectues

- **Key Requirement:** *a dynamic system should keep the application at the same **quality level** after a change*
  - We need to avoid service to deteriorate

In Component-based Systems…

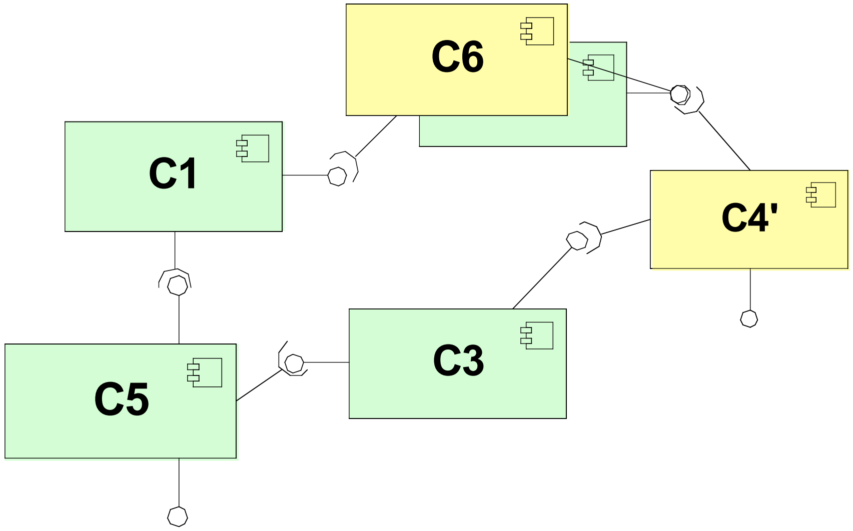
» This problem may even exacerbate in the context of CBSs

- Components are usually black box
- They can be added, replaced, and removed at run-time
- The CBS has some **qualities** it has to satisfy

» The quality of a CBS strongly relies on the

- quality of components
- quality of the CBS **architecture**

The Problem



CBS ├ P

- Model-Checking
- Performance Analysis
- Testing
- Dependendence Analysis
- Security

Evolving CBS ├ P ?

How can we guarantee that the CBS continues satisfying certain properties when evolving?

Static CBS vs Dynamic CBS

- The focus moves from
  - validating the designed architectural **configuration** to
  - validating the **changing over-time** architecture.
- While in static architectures the verification can be done **once and for all** before deployment, for dynamic architectures validation becomes a **perpetual activity** to be performed during system execution too.
- Traditional Analysis techniques need to be complemented by on-line analysis techniques

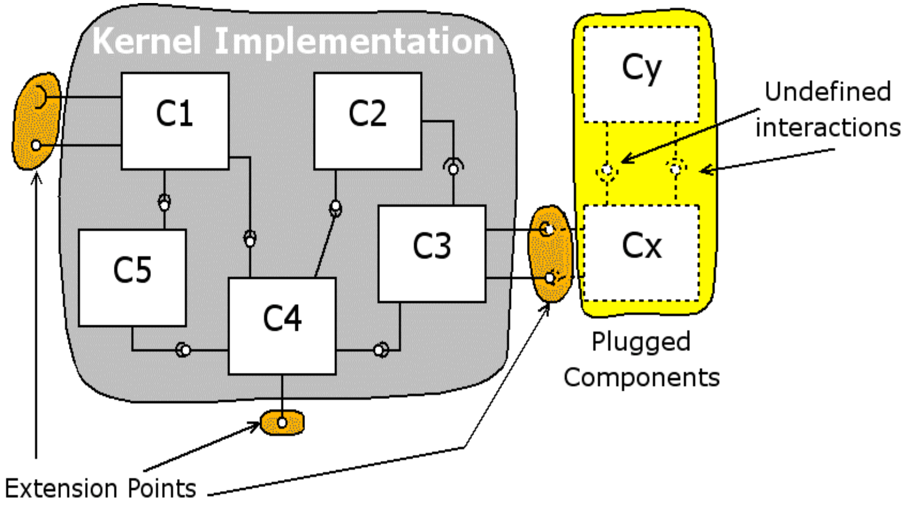
Our Context

» Dynamic CBS

> Black-box Components

> Plugin-based Architecture

- Kernel Architecture
- Pre-defined extension points
- Components can be plugged In at runtime



- Architectural Properties

> Pattern-oriented properties

- Feature-oriented properties

- Assumption: traceability

Our Proposal: MOSAICO

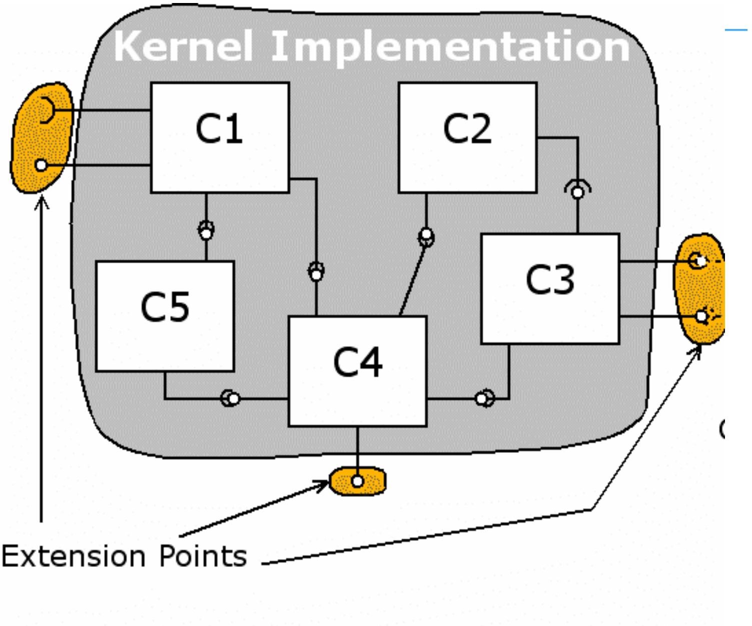
» MOSAICO: MOnitoring SA In COmponents

- Monitoring the CBS implementation
- based on **architecture-level** relevant information
- to collect data on the CBS execution used
  - > to verify that the CBS satisfies certain architectural properties
  - > to verify that the original properties still hold on evolving CBS

Process (1/3)

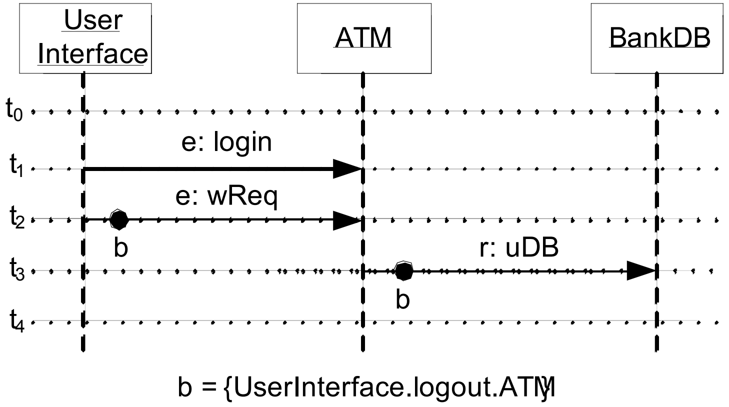
» SA specification of the Kernel Architecture

- Structural Specification
- Interfaces and Extension Points



» Properties Specification

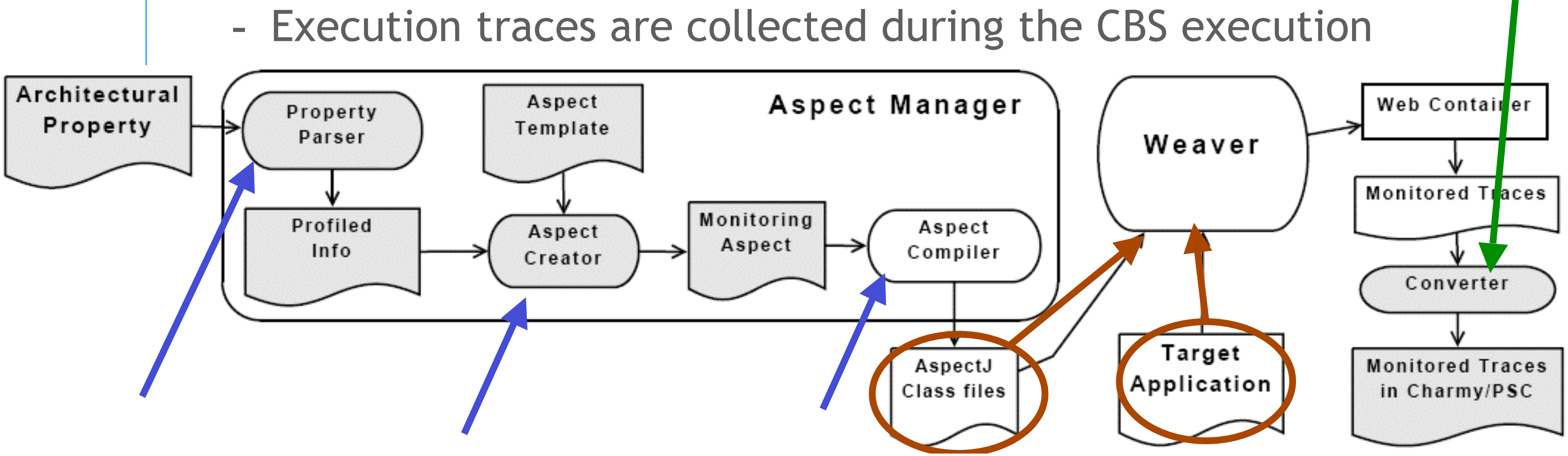
- Using the Property Sequence Charts graphical notation [ASEJournal07]



Process (2/3)

» AOP Instrumentation and Monitoring

- The SA property is parsed and Monitoring Aspects are automatically created
  > Using Aspect Templates
  > For monitoring the desired interactions taking place in the property
- The Monitoring Aspect is weaved into the CBS implementation
- Execution traces are collected during the CBS execution



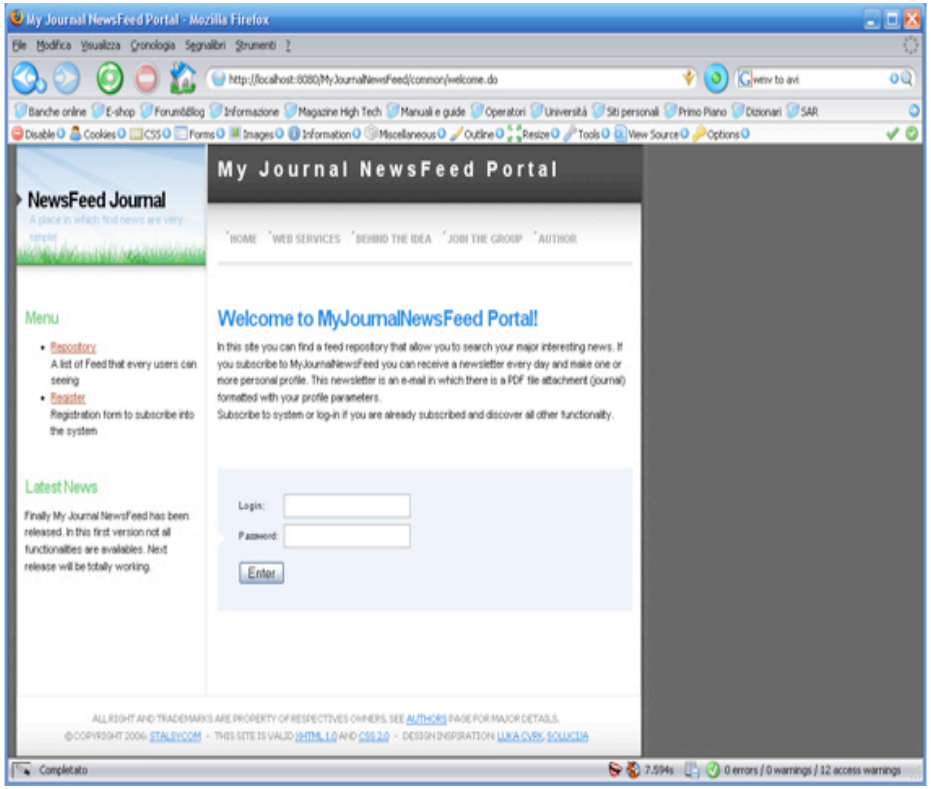
Process (3/3)

» The Analyzer Engine

- To compare the collected execution traces with the properties
  - > the analyzer must be able to check, for each notified event, if it is relevant in the context of any defined property
- Based on “conformance” relations

Experience

- » J2EE NewsFeeder Application
- » To send feed rss in pdf format and via email
- » Black-box system
- » Makes use of design patterns
- » Three tiers:
  - Client, middle, and data tiers


My Journal NewsFeed Portal - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti ?
http://localhost:8080/MyJournalNewsFeed/common/welcome.do
Banche online E-shop Forum&Blog Informazione Magazine High Tech Manuali e guide Operatori Università Siti personali Primo Piano Dizionari SAR
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
HOME
WEB SERVICES
BEHIND THE IDEA
JOIN THE GROUP
AUTHOR
Completato
7.594s
0 errors / 0 warnings / 12 access warnings

My Journal NewsFeed Portal

NewsFeed Journal

A place in which find news are very simple!

Menu

- Repository
  A list of Feed that every users can seeing
- Register
  Registration form to subscribe into the system

Latest News

Finally My Journal NewsFeed has been released. In this first version not all functionalities are availables. Next release will be totally working.

Welcome to MyJournalNewsFeed Portal!

In this site you can find a feed repository that allow you to search your major interesting news. If you subscribe to MyJournalNewsFeed you can receive a newsletter every day and make one or more personal profile. This newsletter is an e-mail in which there is a PDF file attachment (journal) formatted with your profile parameters.
Subscribe to system or log-in if you are already subscribed and discover all other functionality.

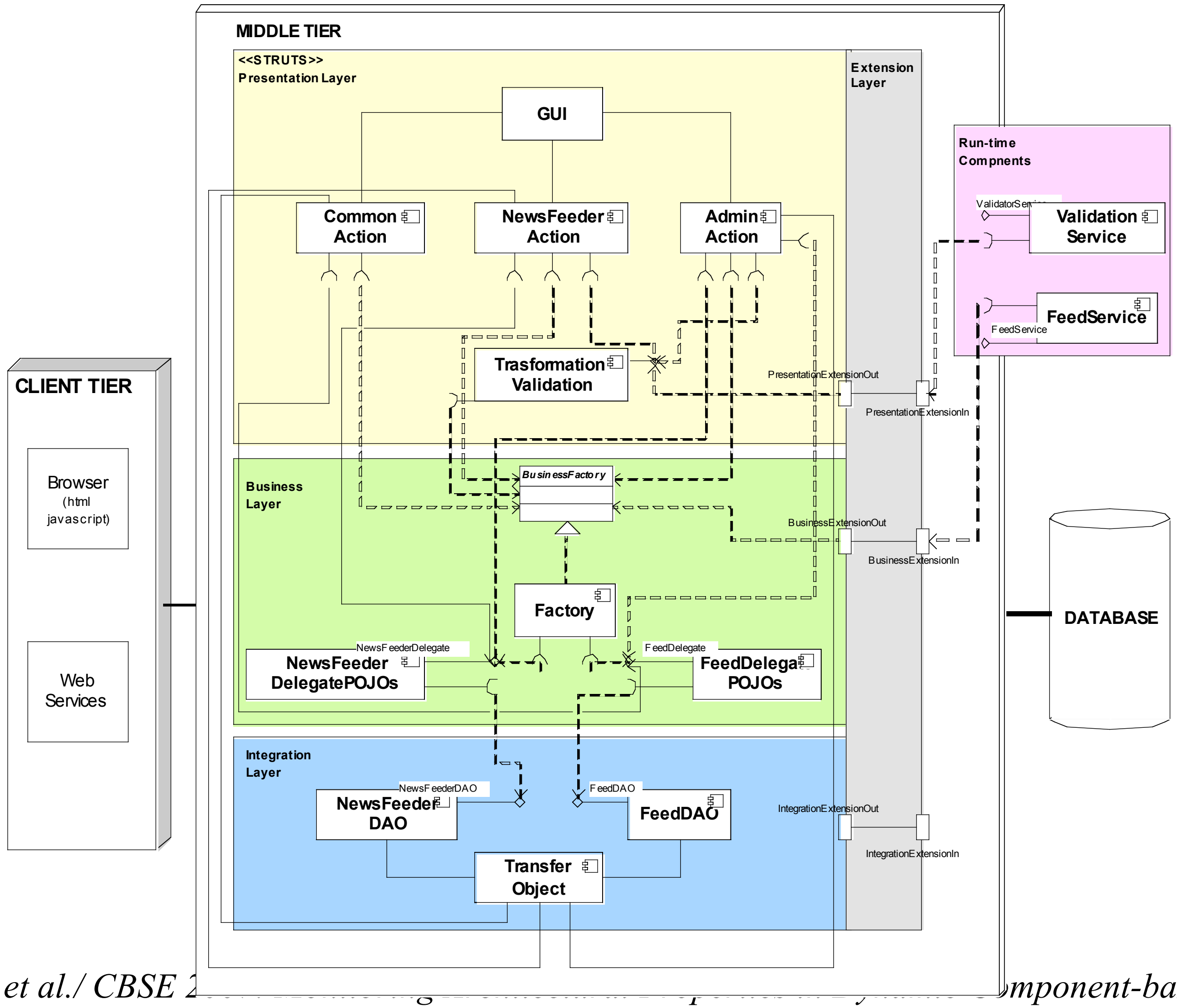
Login:

Password:

Enter

ALL RIGHT AND TRADEMARKS ARE PROPERTY OF RESPECTIVES OWNERS. SEE AUTHORS PAGE FOR MAJOR DETAILS.
©COPYRIGHT 2006: STALSYCOM · THIS SITE IS VALID XHTML 1.0 AND CSS 2.0 · DESIGN INSPIRATION: LUKA CVRK, SOLUCIJA

The NewsFeeder Architecture



Properties

» J2EE Patterns

- A J2EE application can make use of several patterns: Model View Controller, Singleton, Factory, Business delegate, Transfer Object, and Data Transfer Object

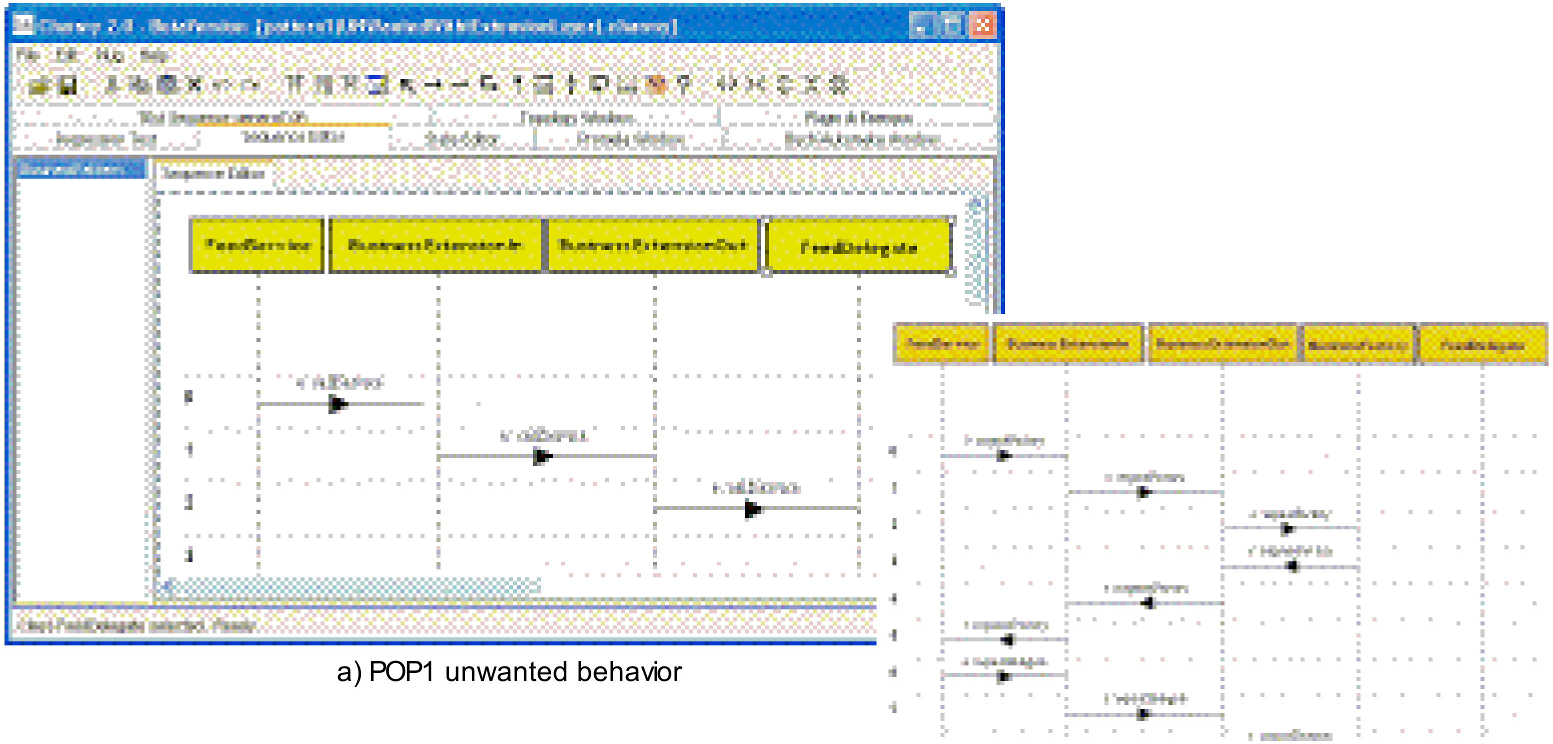
» Our Focus

- *Factory pattern:* the business layer can be accessed only by calling an instance of the business factory
- *Data Transfer Object pattern:* manages the connection to the database and creates the "transfer" serializable object to maintain data received by the database

» Properties:

- POP1 *Direct access to a POJO object (unwanted)*
- POP2 *DAO object creation (unwanted)*
- FOP1 *Business service creation feature*
- FOP2 *Web service request features*
- FOP3 *Remote validation service feature*

Properties



a) POP1 unwanted behavior

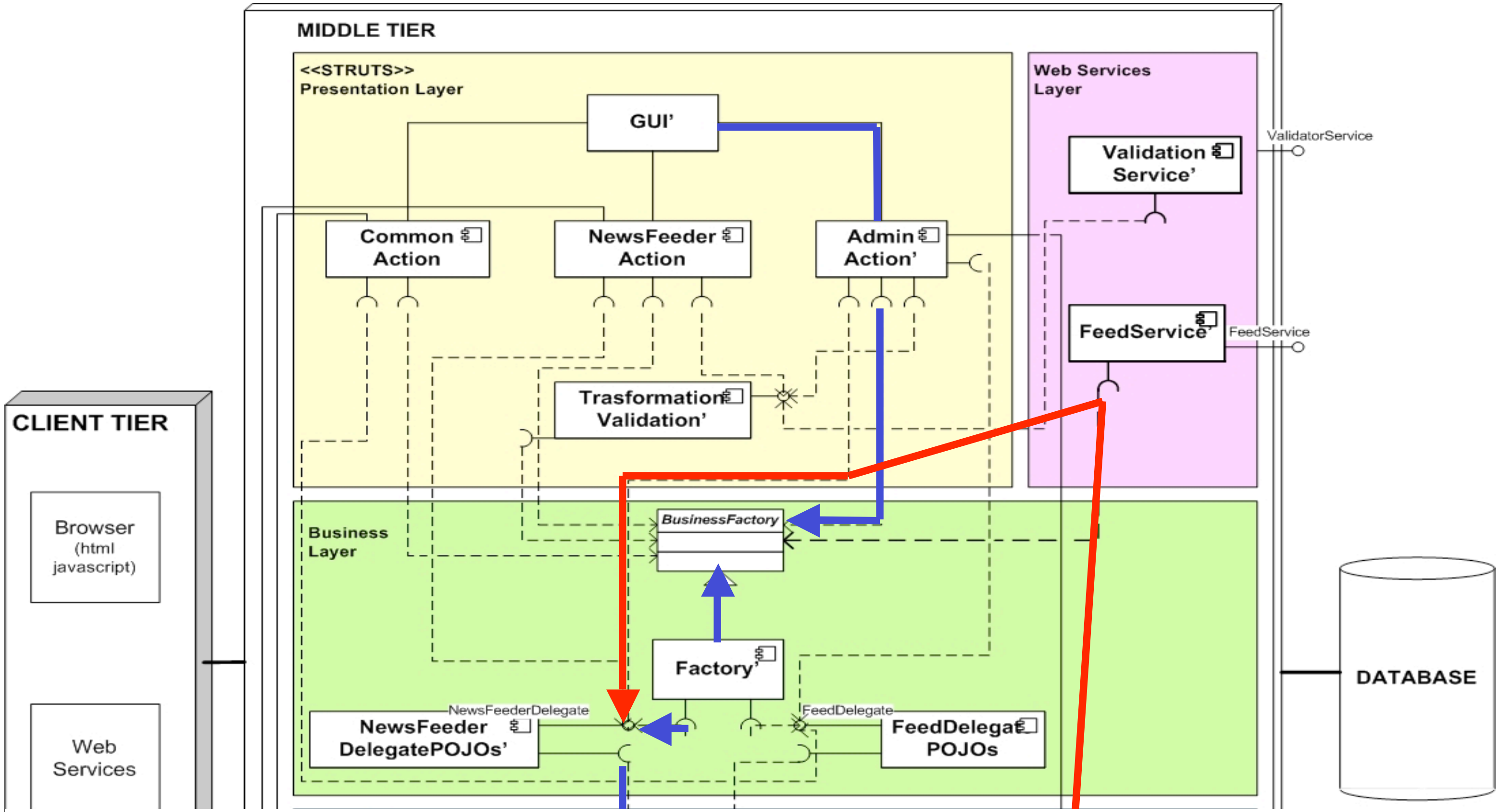
b) POP1 expected behavior

POP1 *Direct access to a POJO object*

Monitoring

- **Mutated** components with interactions breaking the rules defined by the various patterns have been plugged in
- With reference to POP1 we inserted a *FeedService* component that directly accessed an existing POJO instance without first calling the *BussinessFactory* component instance.

Faulty Behavior



»MOSAICO correctly reported the violation.

-Clearly, though the approach seems to provide promising results, its validation as exposed here is still preliminary, and we plan to carry on more formal experimentation.

Future Work

- Validation
  - Coverage
  - Performance overhead analysis
- Service-Oriented CBS
  - A standard configuration does not exist
- New Properties
  - Security
- Run-time Weaving:
  - This is unfeasible so far. To monitor a new property, we have to re-start the application

Monitoring Architectural Properties in Dynamic Component-based Systems

Henry Muccini and
Fabiano Ricci
Computer Science Department
Università degli Studi dell'Aquila
L'Aquila, Italy

Andrea Polini and
Antonia Bertolino
Istituto di Scienza e Tecnologie
dell'Informazione - "A.Faedo"
Pisa, Italy

This work has been partly supported by the EU FP6-2005-IST EU Project PLASTIC and by the national FIRB Project ART DECO