OSIF: A Framework To Instrument, Validate, and Analyze Simulations

Judicaël Ribault - Olivier Dalle
INRIA - CRISAM
University of Nice Sophia Antipolis
I3S-UMR CNRS 6070
BP93 - 06903 Sophia Antipolis, France
judicael.ribault@sophia.inria.fr
olivier.dalle@sophia.inria.fr

Denis Conan - Sébastien Leriche
Institut Télécom, Télécom SudParis
UMR CNRS SAMOVAR
9 rue Charles Fourier
91011 Évry, France
denis.conan@it-sudparis.eu
sebastien.leriche@it-sudparis.eu

ABSTRACT

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIMUTools 2010 March 15–19, Torremolinos, Malaga, Spain.

Categories and Subject Descriptors
I.6.6 Simulation And Modeling
I.6.4 Simulation And Modeling
D.2.13 Software

General Terms
Software

Keywords
Simulation, Instrumentation, Validation, and Analysis

1. INTRODUCTION
2. MOTIVATIONS AND OBJECTIVES

2.1 Separation of Concerns

We propose to use the Aspect-Oriented Programming (AOP) paradigm because it allows such a separation of concerns, and in the end, favor reuse of use cases and components. OIF is a tool for OSA while OSIF aims at being a generic instrumentation and data processing framework.

Figure 1: Simulation workflow focusing on instruments tasks.
2.2 From Real to Virtual System

2.3 From Live to Post-mortem analysis

2.4 Data Processing Composition

3. CASE STUDY

http://www-sop.inria.fr/mascotte/Contrats/spreads
4. OPEN SIMULATION INSTRUMENTATION FRAMEWORK

4.1 COSMOS

4.2 Separation of Concerns

Aspect-Oriented Programming. AOP promotes three principles. Firstly, functional or extra-
up being intertwined, called the "spaghetti" code problem. Applications evolve and become more complex, concerns end
always be cleanly separated from each other and, when ap-
dural or object orientations, these different concerns cannot
pro
tations have to address different concerns such as data man-
agement, security, GUI, data integrity. Using only proce-

2. Figure 2: Separation of concerns using AOP.
Listing 1: Peer Java class without separation of concerns.

```java
public class Peer{
    Sampler sampler;
    String peerName;

    public void boot()
    {
        //modeling code snipped
        sampler.write(peerName+"\boot");
    }

    public void halt()
    {
        //modeling code snipped
        sampler.write(peerName+"\halt");
    }
}
```

Listing 2: AspectJ aspect to observe Peer class.

```java
public aspect peer_instrumentation {
    Sampler Peer.sampler;

    after(Peer peer): execution(void Peer.boot())
    && this(peer)
    {
        sampler.write(peerName+"\boot");
    }

    after(Peer peer): execution(void Peer.halt())
    && this(peer)
    {
        sampler.write(peerName+"\halt");
    }
}
```

Listing 3: Java class with separation of concerns.

```java
public class Peer{
    String name;

    public void boot()
    {
        //modeling code snipped
    }

    public void halt()
    {
        //modeling code snipped
    }
}
```

4.3 From Live to Post Analysis

**COSMOS Collector.**

- Passive Vs. active.
- Observation Vs. notification.
- Blocking or not.
4.4 Composition of instrumentations

COSMOS instrumentation policy.

Figure 3: Graphical representations of data processing in a distributed simulation.
Component-based Architecture.

Fractal ADL.

Architecture Description Language.

Fractal ADL.

Listing 4: Fractal ADL definition of a live analysis of a peer lifetime.

```
<definition name="PeerLifetime" arguments=""
  peername">
  <component name="OutputPolicy">
    ...
  </component>
  <component name="AverageLifetime">
    ...
  </component>
  <component name="CollectorOf{peername}">
    ...
  </component>
</definition>
```
4.5 From Real to Virtual System

5. RELATED WORKS

6. CONCLUSIONS AND PERSPECTIVES
7. ACKNOWLEDGMENTS

This work was partially funded by the European project IST/FET AEOLUS, the ANR project SPREADS and the ARC INRIA project BROCCOLI.

8. REFERENCES


