Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Architecture Analysis of Evolving Complex Systems of Systems (CI07)

Executive Status Report
Software Assurance Symposium
2007

Principal Investigator (PI): Dr. Mikael Lindvall, FC-MD
NASA POC: Sally Godfrey, GSFC
Team members: Chris Ackermann[1], Arnab Ray[1], Lyly Yonkwa[1]
William C. Stratton[2], Deane E. Sibol[2]

1: Fraunhofer Center for Experimental Software Engineering Maryland
2:Johns Hopkins University Applied Physics Laboratory Space Department Ground Applications Group

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Project Goal

- Goal
  - To research and develop a tool for architecture analysis of dynamic (run-time) and static data
- The new tool, Dyn-SAVE,
  - Will extend the already existing *static* Software Architecture Visualization and Evaluation (SAVE) tool
- Background
  - SAVE successfully applied to JHU/APL's Common Ground System in 2006 NASA Research Infusion project
  - Architecture = structure + behavior
  - Need for dynamic architecture analysis was identified

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Motivation

- Systems are often difficult to understand
  - Static and dynamic architecture very different
  - Distributed systems of systems hard to understand
- System verification is difficult, e.g.
  - Interface Control Documents interpreted differently
  - Changes of COTS behavior make upgrading risky

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

The (static) SAVE Tool

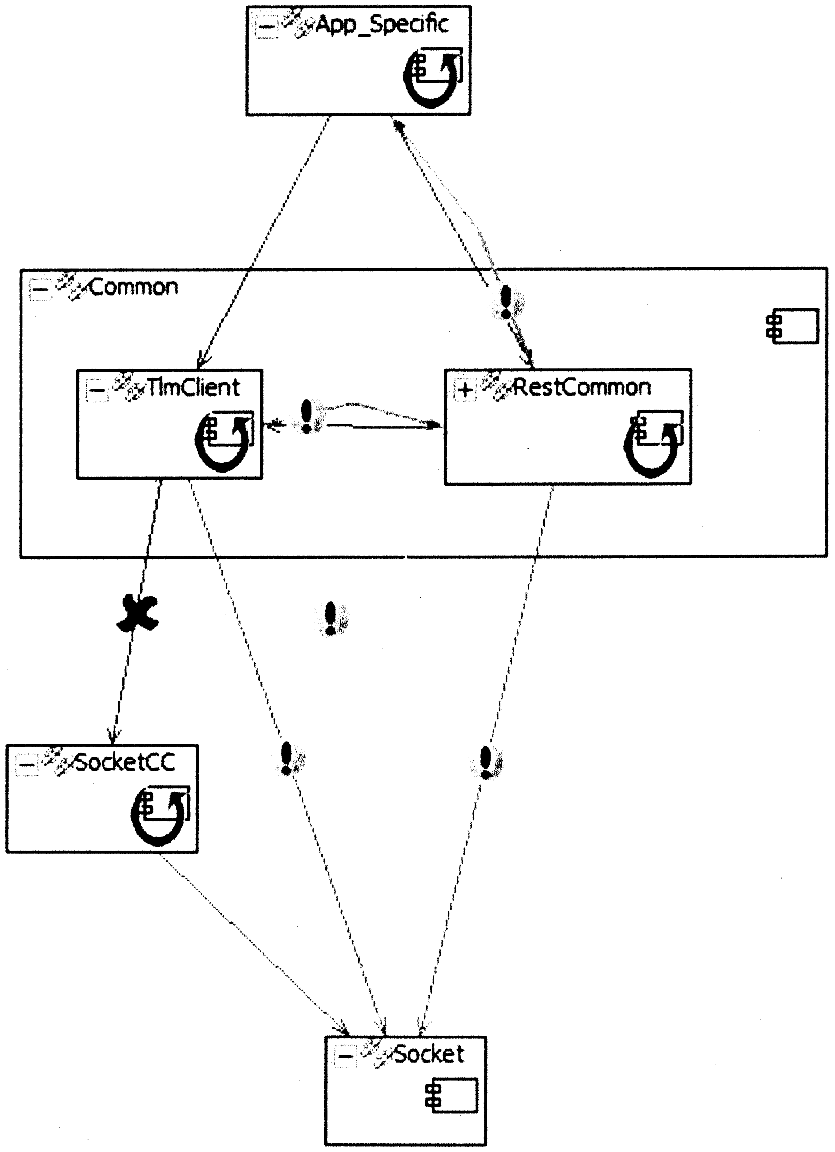
- Objective: Make Architecture/Design specifications alive!

- Helps answer: Does the implementation match the plan?
  - Define a *planned* (and/or target) architecture (using rules etc);
  - Create an *actual* architecture from source code;
  - Compare planned architecture w/ actual, identifying architectural violations

- Features for Zooming, Filtering, Refactoring
- Language independent: C/C++, Java, Delphi, Ada, Simulink, Fortran

- Conclusion after applying SAVE at APL and to many other systems:
  - The SAVE approach is useful and practical
  - One can quickly model and analyze software architectures
  - But has some weaknesses since it's based on static analysis

Current (static) SAVE Capabilities



Using static SAVE, we can identify some violations, but

1. Are these couplings harmful?
2. In what order do the couplings occur?
3. Who does socket communicate with?
4. Is that communication correct?
5. What components are responsible for that communication?

Let's see how these issues **could** be analyzed in the **future** using Dyn-SAVE!

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Dyn-SAVE Capabilities (Vision)

Specify Planned Behavior

Compare Planned and Actual Behavior

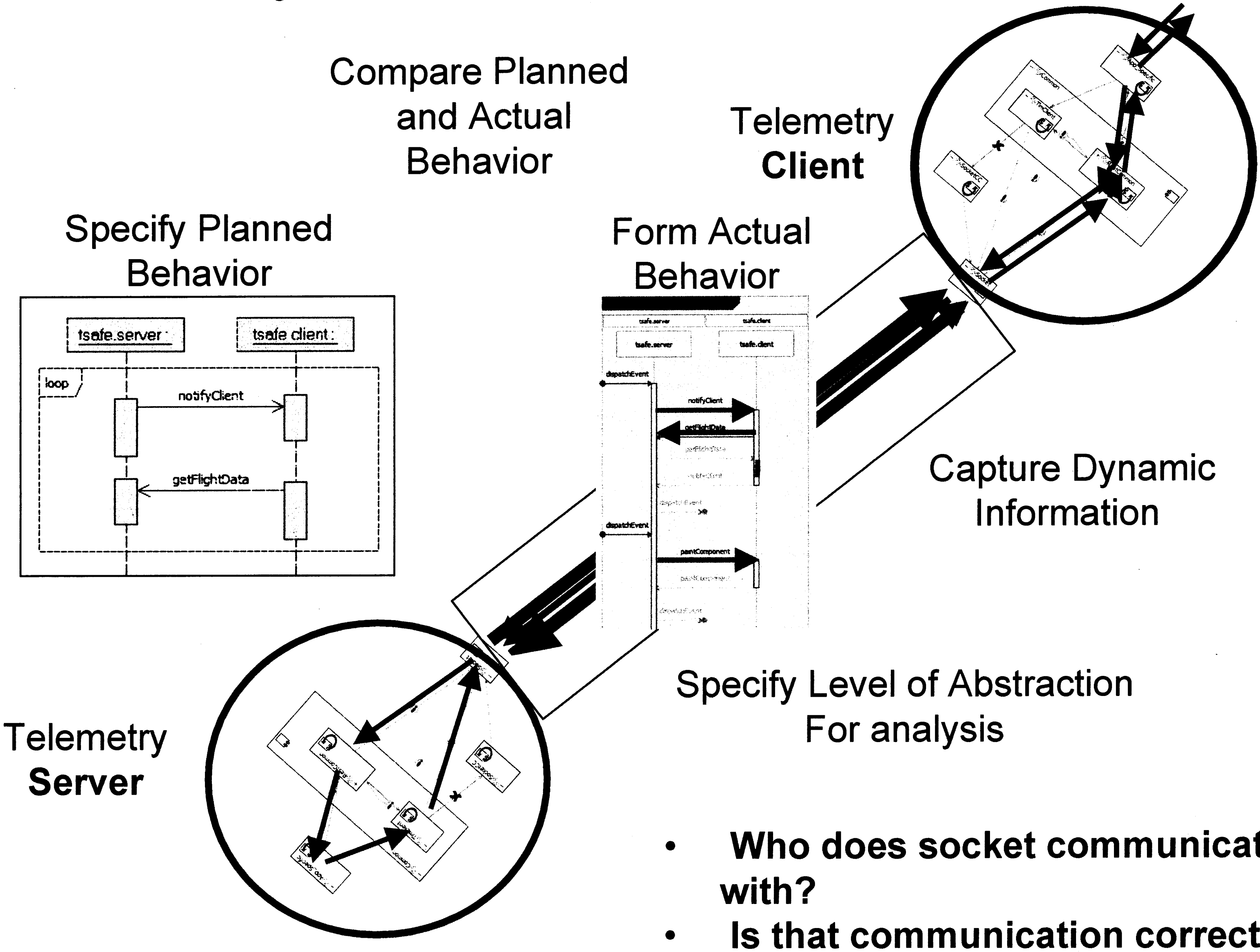
Form Actual Behavior

Telemetry **Client**

Capture Dynamic Information

Specify Level of Abstraction For analysis

Telemetry **Server**



- **Who does socket communicate with?**
- **Is that communication correct?**

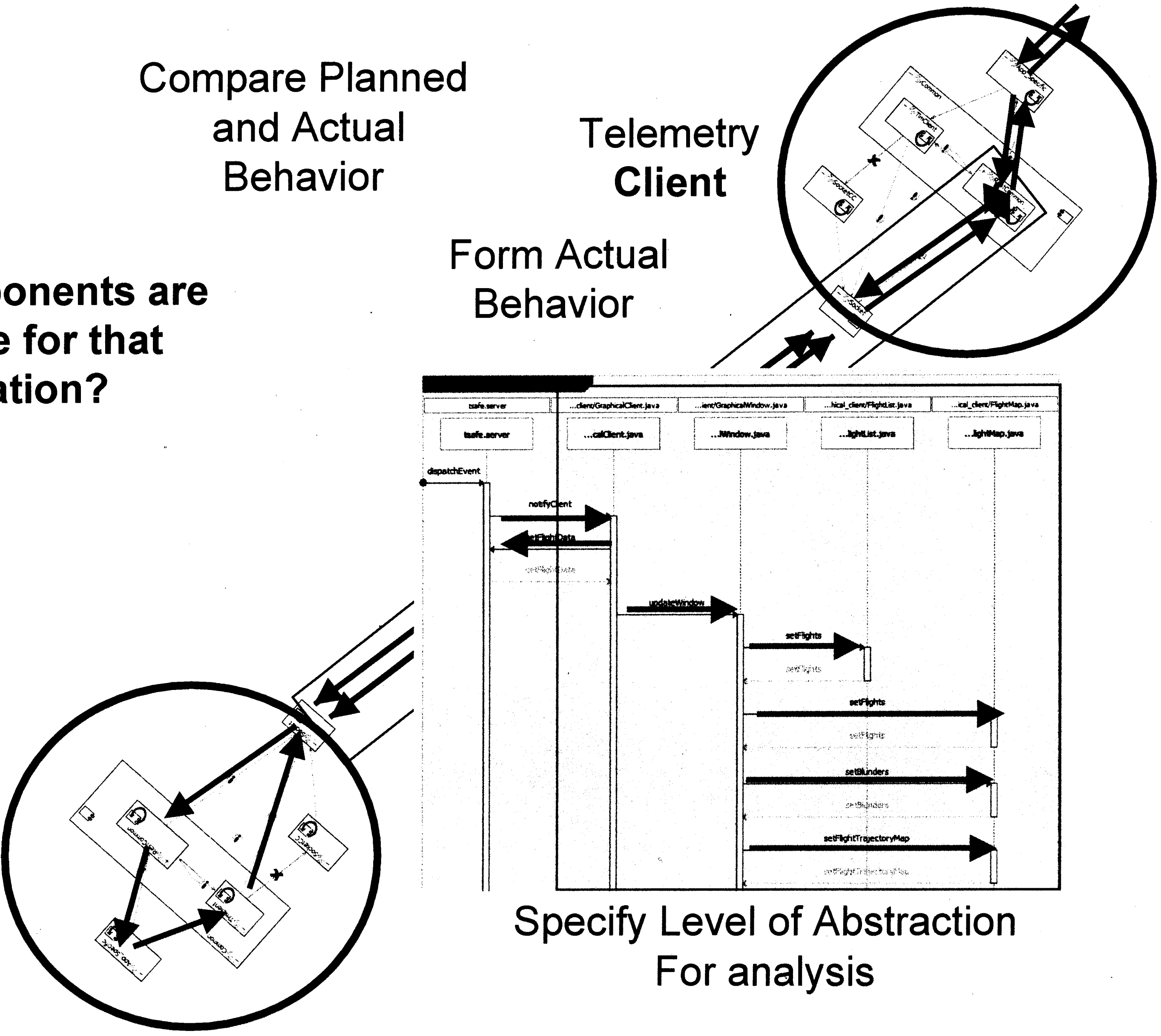
Dyn-SAVE Capabilities (Vision)

Compare Planned and Actual Behavior

Telemetry **Client**

Form Actual Behavior

What components are responsible for that communication?



Telemetry **Server**

Specify Level of Abstraction For analysis

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Approach

- Work as one team with problem-owners at APL
- Experiment with new technology; apply to FC-MD testbed
- Evaluate new technology; apply it at APL
- Improve technology based on feedback, results
- Repeat

- When technology is mature, extend to NASA projects
  - e.g. through Research Infusion projects

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Summary

- Approach is to apply Visualization and Evaluation concepts to Dynamic Analysis
- Combining static and dynamic information
- Experimentation using TSAFE testbed
- Evaluation on APL's Common Ground System