

An Evaluation of Component- based Software Design Approaches

Diego Puppini, Fabrizio Silvestri,
Domenico Laforenza

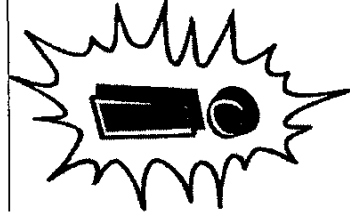
ISTI - CNR, Pisa, Italy
diego.puppini@isti.cnr.it

Introduction

- Growing attention for a component-oriented software design of Grid applications
 - Business and scientific problems
- Goal: applications by assembling together independently developed-software components
- Components: independently developed, composable, re-usable, substitutable software solutions, with clearly defined interface and behavior

Today approaches to components

- 1) based on Interface Description Language (IDL)
 - CCA, CORBA
- 2) Based on introspection and design conventions
 - Java Beans
- 3) NEW: Automatic extraction of signature information and bridging



IDL usage: CCAffeine

- Usual process:
 - Definition of the component interface using Babel;
 - Implementation of the component in the Babel-generated Impl file (in any supported language);
 - Definition of the CCA ports in the Impl file;
 - Creation of a wrapper and a CCA definition file;
 - Creation of a suitable makefile.
- Problems of SW engineering

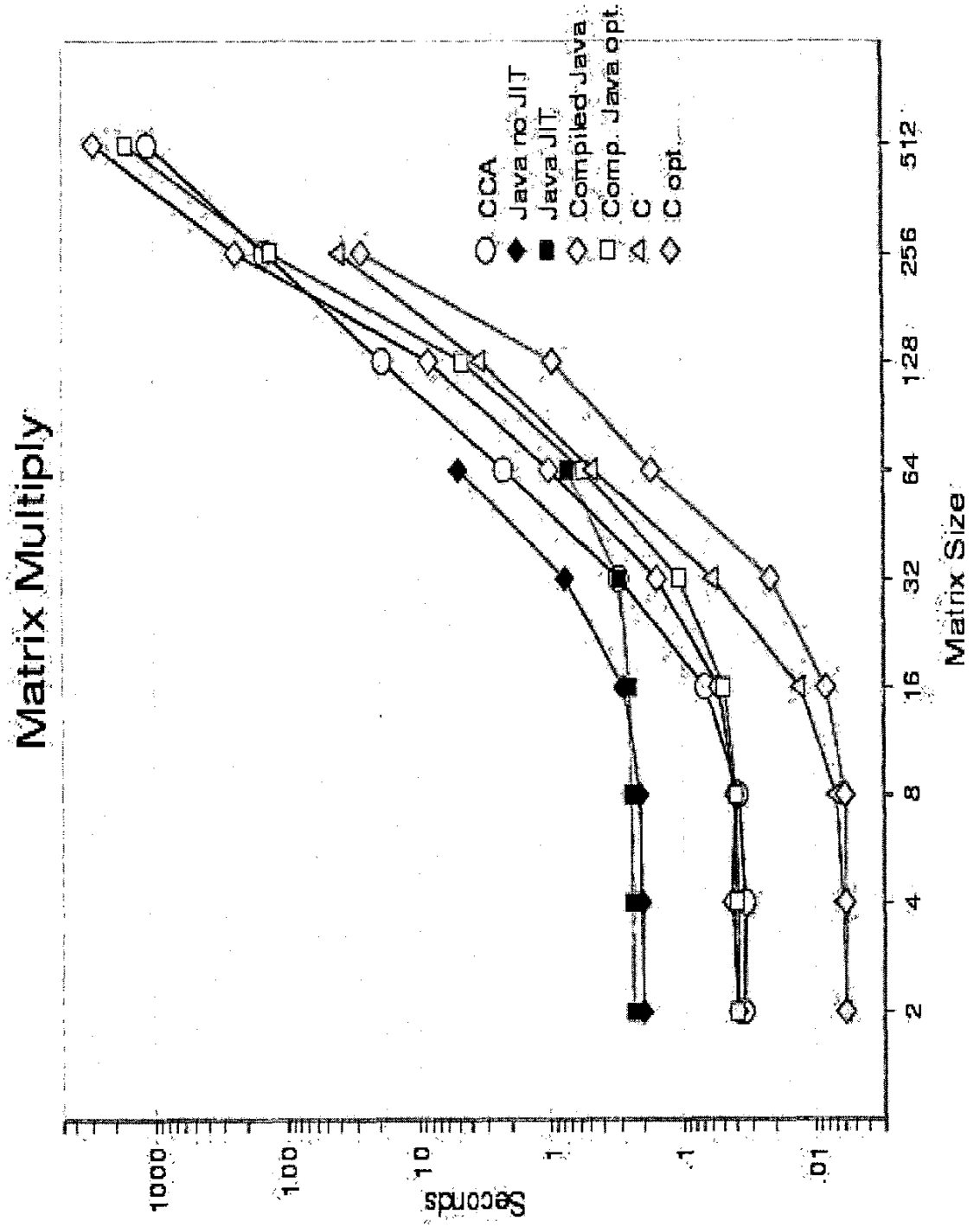
Introspection: Java Beans

- Usual process:
 - Extension of a Java class into a Bean
 - Bean queried by introspections
 - Design conventions are recognized and matched
 - The framework shows properties and interface
 - Graphic manipulation in the environment

Comparison

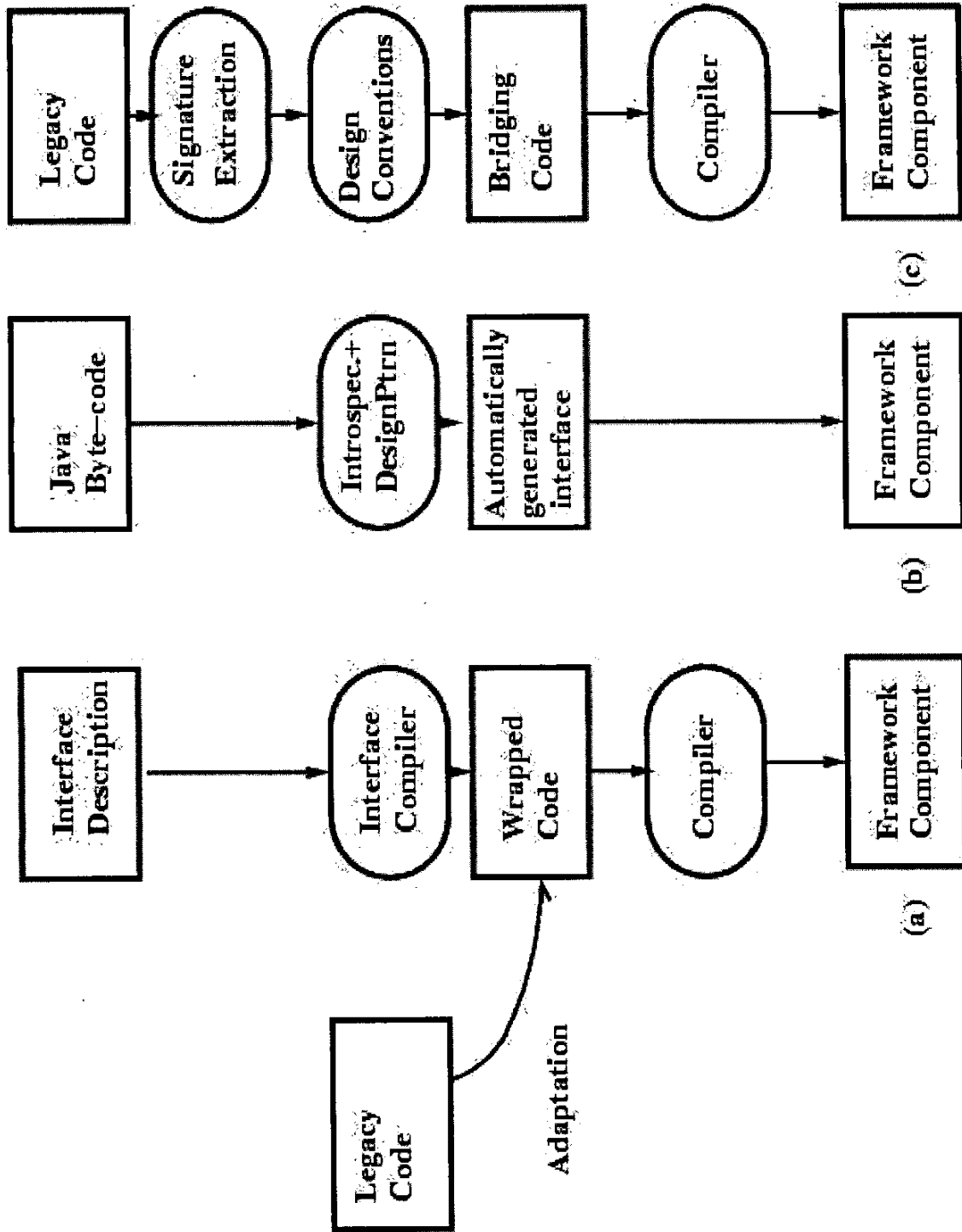
- CCAffeine/IDL
 - Interfaces needs to be passed around
 - Cross-language portability
 - Emerging standard
 - Limited manipulation
 - Overhead due to data abstraction
- JavaBeans
 - Component are packaged as JAR
 - Based on portable virtual machine
 - Recognized standard
 - Visual manipulation
 - Overhead due to virtual machine

Performance comparison



A new approach

- Legacy code is developed with the tool of choice
- Signature is extracted using headers, pattern matching, binary analysis
- Signature is analyzed according to standard and design conventions
- Bridging code to frameworks is generated, with ad-hoc conversions



(A) IDL (B) Java Beans (C) automatic extraction of signature and bridging

Related work

- JACAW: C to Java wrapper
- Li et al.: wrapping C/MPI to CORBA/Java
- Stuer et al.: H2O pluglets can be exported as WSDL Web-Services
- Mocha, GDB to analyze binary code