SW Technology for next generation ESO Control Systems

The Trieste Meeting on Y2K Real-time Controls in Physics
Astronomical Observatory of Trieste, 14-15 of January 1999

G.Chiozzi, ESO

14/01/1999

Contents

- Introduction
- OO Analysis and Design: UML
- Distributed objects: CORBA
- OO programming languages: JAVA
- Conclusion

Introduction

- The VLT Telescope Control Software has been implemented using extensively OO technology:
  - Analysis and Design: Booch methodology
  - Implementation: C++ programming language
- New OO technologies have emerged as industrial standards in the last few years:
  - Analysis and Design: UML
  - Implementation: Java programming language
  - CORBA object distribution protocol

We are now evaluating these technologies for the next components of the VLT/VLTI and as a base for the next generation Control Systems

OO Analysis and Design: UML

- The Unified Modeling Language is a well accepted standard for OO Analysis and Design and it has the Booch methodology as a base.
- Pilot project: VLTI Auxiliary Telescopes Control Software
- Current phase: Requirements Analysis completed with Use Cases:
- Tools under test: Rational Rose, integration with World Wide Web

The first impression has been very positive and Use Cases seem a very effective way of capturing requirements.

Integration with the WWW makes collaboration much more effective

Distributed objects: CORBA

- CORBA is being now accepted as an industrial standard for object distribution (for example Netscape provides support to CORBA apps).
- CORBA provides not only a comunication protocol but also a distribution architecture.
- We want as a first step provide a CORBA interface to our current VLT Control System:
  - We gain experience on CORBA
  - New applications talking to the core of VLT TCS can be based on CORBA and other new technologies
  - We do not need to rewrite our core system
- New systems will be based on a CORBA architecture

What we have seen up to now is very promising, but we do not have yet enough on hands experience to say more.

OO programming languages: JAVA

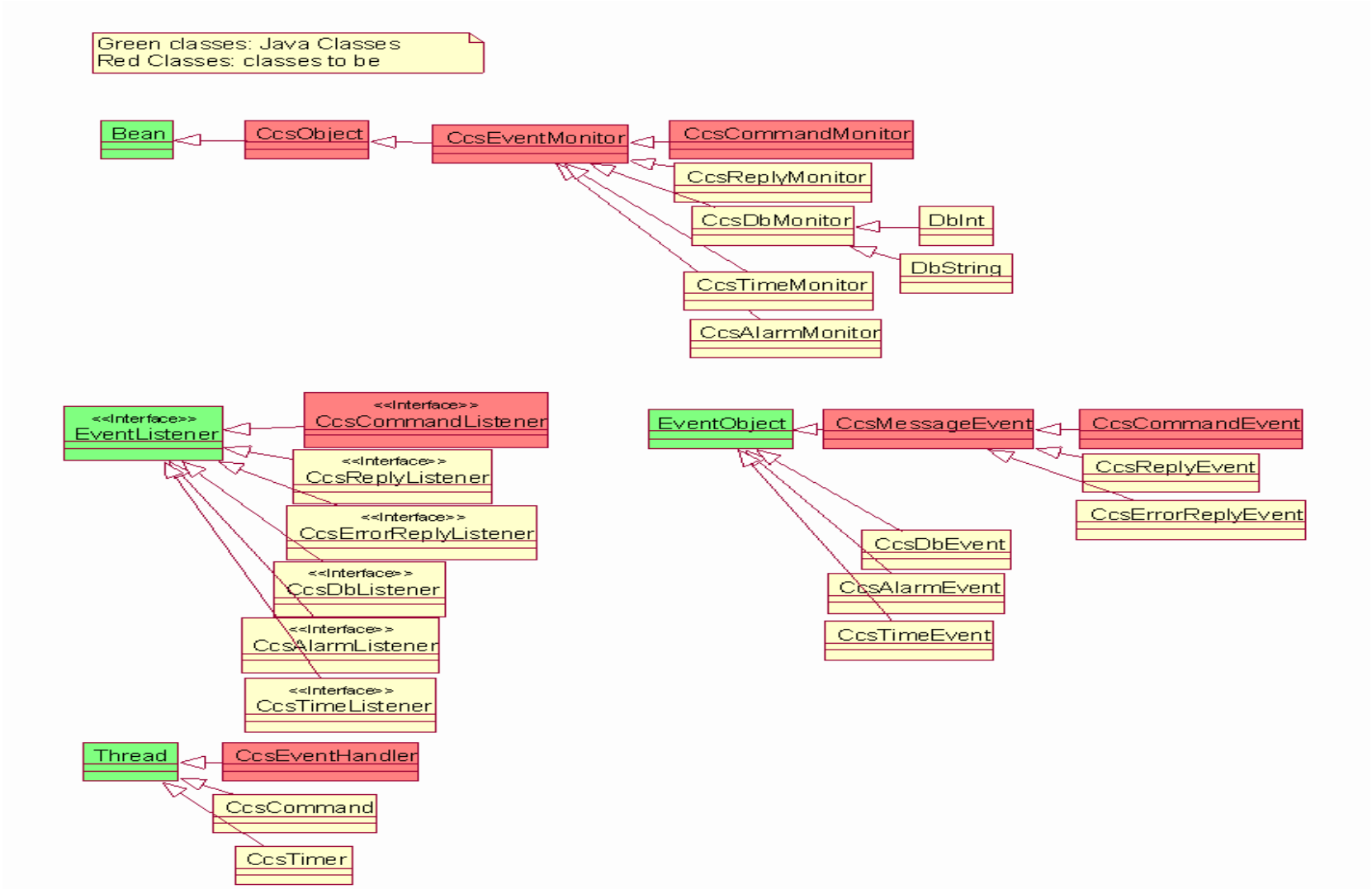
- Java is exploded on the market as a major OO programming language.
- Still Java seems to us not mature enough for control applications (performances, weakness in real-time, HW dependent support).
- Pilot project: Java interface to VLT TCS:
    - Phase 1: Feasibility prototype (almost completed)
    - Phase 2: Implemented with native methods (1999)
    - Phase 3: Implemented based on CORBA (2000)
- Candidate usage areas:
    - AT User Interface
    - VLTI Combined Operation
    - VLT/VLTI remote operation and maintenance SW
    - .....LSA control software

Java First impressions:

- **Pros:**
  - + Programming conventions are very good and comprehensive
  - + The language is easier than C++ and very easy for C++ programmers
  - + Standard class libraries are very extensive

- **Cons:**
  - - Some limitations compared to C++. Difficult access to low-level functions. Many limitations for Real-Time applications.
  - - Many portability problems on non-SUN architectures. VM immature.
  - - Poor development tools on Unix (debugger)
  - - Poor run-time performances. Slow compiler.
  - - Documentation, support and available books of low quality
  - - Language still unstable: too frequent new versions.

Java - CCS interface classes



Conclusion

- UML is a very good choice for analysis and design
- Java and Corba (for what we can say) are where we have to go for new Control Systems.
- They are still immature to base NOW on them the core of our Control Software.