

Demonstration of the Rodin Platform Prototype

Laurent Voisin,
Stefan Hallerstedde,
Farhad Mehta,
Thai Son Hoang,
Francois Terrier,
Jean-Raymond Abrial

Rodin

Aix-en-Provence - 5th April 2006

- Reminder of **main decisions**
- Schematic **view of the system**
- **Scenario** of the platform demo

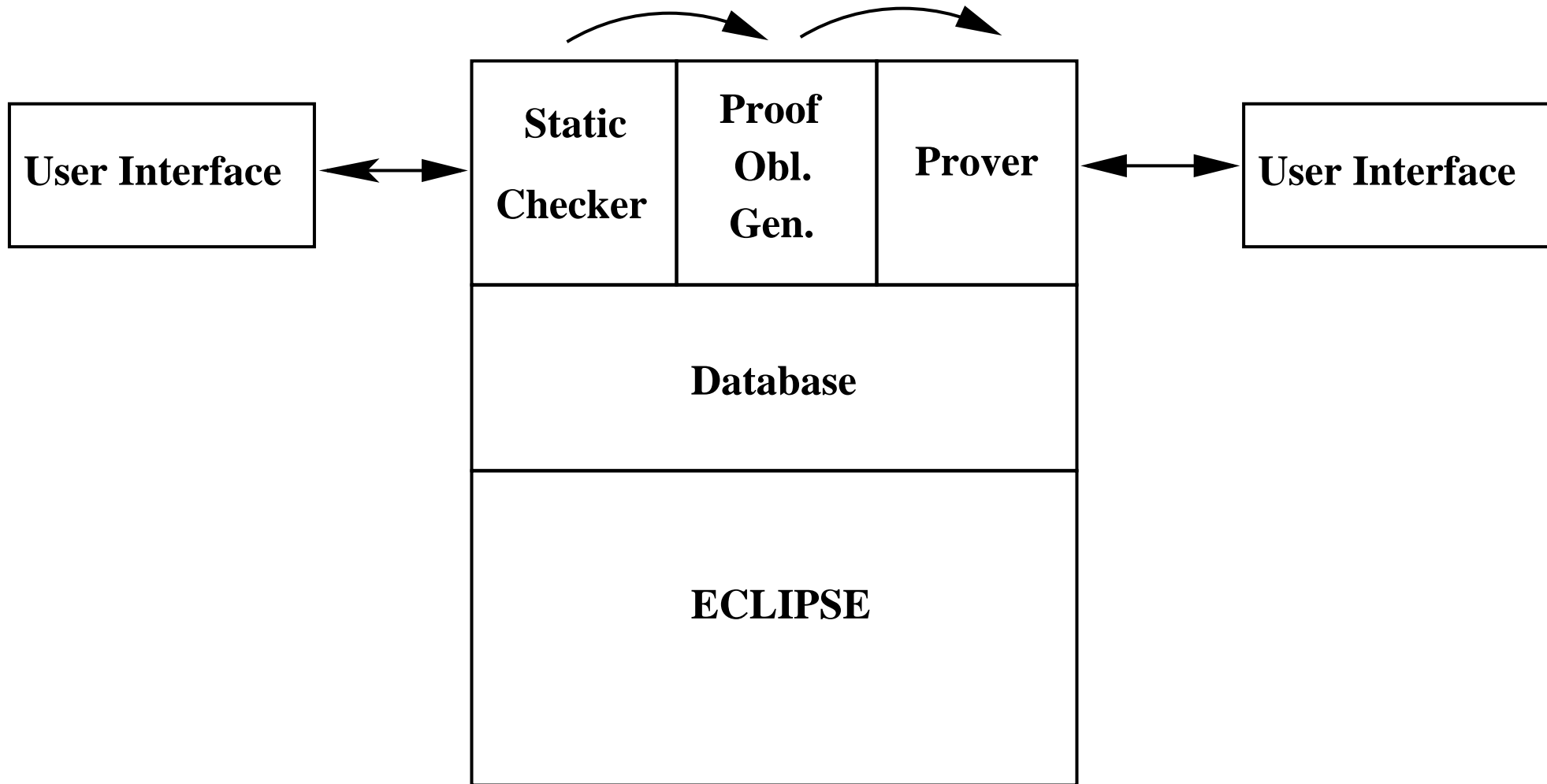
Main Decisions (1)

- **Complex Systems** are developed through **modelling**
- **Modelling** is performed on an **extensible** development **database**
- The **database** is implemented on a **Platform** based on **Eclipse**
- The **persistence** of the database is realized by **XML files**
- The **working representation** of data is made by **Java Objects**.
- **Extensibility** is realized on the XML files and the Java classes.

Main Decisions (2)

- The **platform** supports a **reactive** and **differential** approach
- This approach first deals with **three kernel tools**
- The **kernel tools** are: Static Checker,
Proof Obligation Generator,
Prover
- The kernel tools are **automatically invoked**
- This is performed by an **Eclipse builder**

Schematic View of the Basic System



Deliverable D3.4: the Prototype

- Limited Event-B structure: **Initial model** and **context** only
- Database on these **limited modelling elements**
- **Full** mathematical language
- **Simple** static checker, proof obligation, and prover
- Database and prover **graphical interfaces**

- We have **two demos**:
 - a **very simple** demo
 - a **more elaborate** demo

- **Creating** a project
- **Entering** some modelling elements
- **Showing** the reactive aspect of the tool

Second Demo

- We receive the **archive** of the **incomplete model** of a **bank**
- The archive is loaded in the **Rodin platform**
- We add **more modeling elements** in the database
- We obtain a **new proved model** of the bank
- The new model is **sent back** to the originator (**new archive**)