JACK

Java Applet Correctness Kit

JACK : overview

- Curriculum Vitae / Motivations
- Specificities / Features
- Status
- Demonstration
  - eclipse
  - Verification condition viewer

JACK : overview

- Curriculum Vitae / Motivations
- Specificities / Features
- Status
- Demonstration
  - eclipse
  - Verification condition viewer

From birth to early childhood

- Developed at Gemplus
  - From Jan 2002 to April 2003
- Initial objective : “Bring to developers tools that help them to provide and be accountable of a certain quality level on their code”.
  - Conform to Specification Requirements
  - Documented
  - Without bugs

EVEREST

Motivations

- Develop languages and tools that must get developers agreement.
- Our solution
  - Keep Java as the validation language;
  - Do not change working environment;
  - Provide different validation levels.

Let's developer enjoy the validation phase

EVEREST

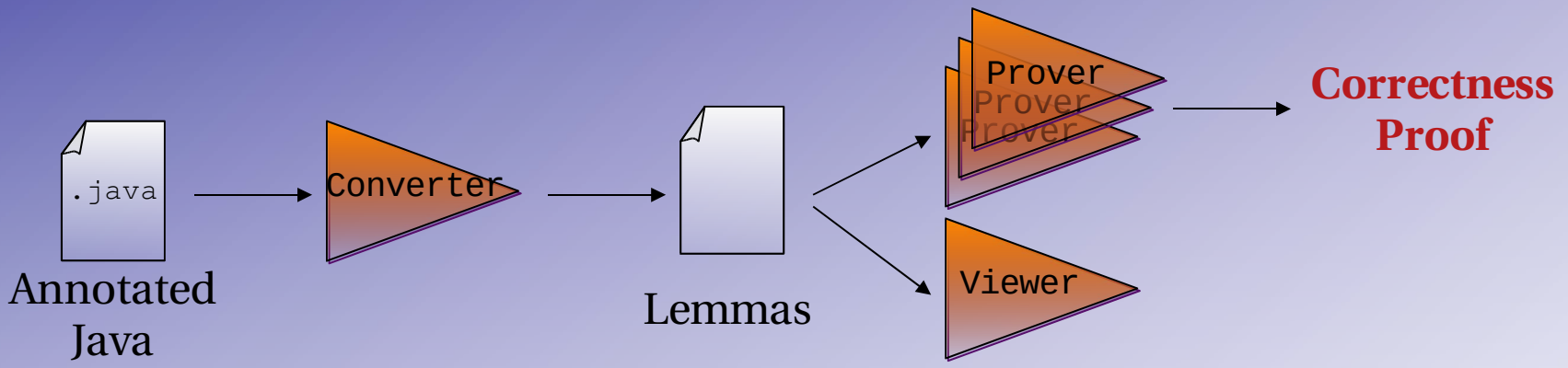
Design choices

- Keep Java as the validation language
  $\Rightarrow$JML
- Do not change working environment
  $\Rightarrow$eclipse
- Provide different validation levels
  $\Rightarrow$Automatic and interactive provers

So JACK is

- A plugin to install in your favorite IDE allowing you to associate a formal documentation to your code and to check that this documentation is correct.
- Allows to
  - Write JML specification;
  - Generate lemmas ensuring that the Java implements the JML specification;
  - Provide a view of the lemmas in the Java language.

JACK Framework



- Converter from JML annotated Java to formal lemmas
- Allows using automatic/interactive provers
- Lemma viewer hiding mathematics material

EVEREST

From childhood to adolescence

- Transfered to INRIA in Sept. 2003
- Enforce the correctness
- More plugins
  - Provers : Simplify, PVS, Coq
  - Annotation generation
  - Annotation propagation
  - Validation at bytecode level

EVEREST

JACK new family

- EVEREST Project
  - aims at ensuring system security for mobile and embedded applications
  - privileged application domains are small, trusted and portable devices, such as mobile phones and smart cards, their operating systems and the applications running on these devices

EVEREST

Correctness / Robustness

- Improve the tool with some test on *big* application:
  - PACAP (JavaCard purse and loyalty)
  - IP Stack, Bytecode Verifier (OS component)
- Work on well definedness lemmas (not achieved)

More plugins 1/2

- Annotation generation
  - Add automatically some annotations: useful when checking for runtime exceptions
- Annotation propagation
  - Mariela Pavlova and Thibault Dupont

More plugins 2/2

- Simplify plugin
- PVS plugin
- Coq plugin
  - Julien Charles and Benjamin Gregoire
- HaRVey plugin
  - Current work with Silvio Ranise in the GECCOO ACI French project

Now, Jack

- Is downloadable on the Jack web site;
- Is currently tested by industrial mainly in the smart card industry;
- Is plugged with 4 theorem provers;
- Is presented in a tutorial at FME 2005...

JACK : overview

- Curriculum Vitae / Motivations
- Specificities / Features
- Status
- Demonstration
  - eclipse
  - Verification condition viewer

Features

- Plugin in eclipse
- Automated Weakest Precondition Calculus
- Java/Jml Proof Obligation Language (JPOL)
- Pluggable with automatic and interactive provers

EVEREST

Plugin in eclipse 1/2

- Benefit from the eclipse IDE and other JML plugin(s):
  http://jmleclipse.projects.cis.ksu.edu
- Bring formal method into the developer world rather than bringing developers to the formal world.

Plugin in eclipse 2/2

- Dedicated perspective
  - Views for verification condition visualization
  - Proof task management
- Different menus and buttons to configure and launch JACK features.

WP Calculus 1/3

- Java language annotated with JML
  - Mono-thread
  - No double, floating point
    - No prover with such feature
  - Some (most of) JML keywords
    - Sufficient to express properties (functional and security)

WP Calculus 2/3

- Fully automated
  - Loops are annotated with invariant and modified variables
    - Specific JML keyword: *loop_modifies*
  - Possibility to annotate block in order to cut the combinatory explosion

WP Calculus 3/3

- Lemmas calculated for each method
  - Conjunctive lemmas are split during the generation
  - Obvious contradictions are detected as soon as possible
  - Control flow informations are stored in manner to help understanding the proof obligation in the lemma viewer.

EVEREST

JPOL 1/2

- Generate lemmas in an internal language
  - Mix between JML, Java and Set Theory
  - Minimal background theory :
    - REFERENCES, instances , null, ...
    - typeof, subtype, ...
    - arraylength, arrayelements, ...
  - Instance fields are arrays

JPOL 2/2

- JPOL lemmas are the entry point for the plugin provers.
- A plugin prover contains a translator from JPOL to its language and can potentially launch automatic proof tasks or interactive proof activities
  - Use the eclipse plugin extension point mechanism.

JACK : overview

- Curriculum Vitae / Motivations
- Specificities / Features
- Status
- Demonstration
  - eclipse
  - Verification condition viewer

Status

- Jack v1.8.0
  - Annotation Propagation
  - ByteCode to JPO
- B plugin
- Coq plugin
- PVS plugin
- Simplify plugin

Status

- License for education and experimentation for Jack and the B plugin
  - Current negotiation with Gemplus
- The Coq, PVS and Simplify plugins are freely available.

JACK : overview

- Curriculum Vitae / Motivations
- Specificities / Features
- Status
- Demonstration
  - eclipse
  - Verification condition viewer