

Kopitiam: Modular Incremental Interactive Full Functional Static Verification of Java Code

Hannes Mehnert
IT University of Copenhagen
hame@itu.dk

18th April 2011
3rd NASA Formal Methods Symposium, Pasadena

Motivation

- Develop Java code and Coq proofs side-by-side
- Separation logic is intuitively usable for OO programs
 - Enables sharing and local reasoning
- Research project at ITU Copenhagen
 - Higher-order separation logic (for both specifications and assertions) formalized in Coq
 - Tools and methods for scalable software verification
 - Lars Birkedal
 - Peter Sestoft
 - Jonas, Filip
 - Jesper, Jakob

Motivation

```
Interface ICell {  
    int get ()  
    void set (int x)  
}
```

```
class Cell implements ICell {  
    int value  
    Cell () {}  
    int get () {  
        return value  
    }  
  
    void set (int x) {  
        value = x  
    }  
}
```

Specification of ICell

Specification of ICell

- Parametrized with class

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get() } \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get() } \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$
- $\{ \text{Repr this } t \} \text{ set}(x) \{ \text{Repr this } (\text{fset } t \ x) \}$

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get}() \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$
- $\{ \text{Repr this } t \} \text{ set}(x) \{ \text{Repr this } (\text{fset } t \ x) \}$
- t is mathematical model (valid over $\forall t : T$)

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get() } \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$
- $\{ \text{Repr this } t \} \text{ set}(x) \{ \text{Repr this } (\text{fset } t \ x) \}$
- t is mathematical model (valid over $\forall t : T$)
- Repr is representation predicate: $\text{val} \rightarrow T \rightarrow \text{Pred}(\text{heap})$

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get}() \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$
- $\{ \text{Repr this } t \} \text{ set}(x) \{ \text{Repr this } (\text{fset } t \ x) \}$
- t is mathematical model (valid over $\forall t : T$)
- Repr is representation predicate: $\text{val} \rightarrow T \rightarrow \text{Pred}(\text{heap})$
- fget and fset are mathematical functions

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get}() \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$
- $\{ \text{Repr this } t \} \text{ set}(x) \{ \text{Repr this } (\text{fset } t \ x) \}$
- t is mathematical model (valid over $\forall t : T$)
- Repr is representation predicate: $\text{val} \rightarrow T \rightarrow \text{Pred}(\text{heap})$
- fget and fset are mathematical functions
 - $\text{fget}: T \rightarrow \text{val}$ $\text{fset}: T \rightarrow \text{val} \rightarrow T$

Specification of ICell

- Parametrized with class
- $\{ \text{Repr this } t \} \text{ get() } \{ \text{ret}, \text{Repr this } t \wedge \text{ret} = \text{fget } t \}$
- $\{ \text{Repr this } t \} \text{ set}(x) \{ \text{Repr this } (\text{fset } t \ x) \}$
- t is mathematical model (valid over $\forall t : T$)
- Repr is representation predicate: $\text{val} \rightarrow T \rightarrow \text{Pred}(\text{heap})$
- fget and fset are mathematical functions
 - $\text{fget}: T \rightarrow \text{val}$ $\text{fset}: T \rightarrow \text{val} \rightarrow T$
- model must satisfy: $\text{fget } (\text{fset } t \ x) = x$

Instantiation for Class Cell

- $T = \text{val}$ $\text{Repr} = \lambda c, v. c.\text{value} \mapsto v$
- $\text{fget} = \lambda v. v$ $\text{fset} = \lambda _, v. v$
- $\{ \text{this.value} \mapsto t \} \text{get}() \{ \text{ret}, \text{this.value} \mapsto t \wedge \text{ret} = t \}$

Instantiation for Class Cell

- $T = \text{val}$ $\text{Repr} = \lambda c, v. c.\text{value} \mapsto v$
- $\text{fget} = \lambda v. v$ $\text{fset} = \lambda _, v. v$
- $\{ \text{this.value} \mapsto t \} \text{get}() \{ \text{ret}, \text{this.value} \mapsto t \wedge \text{ret} = t \}$
- $\{ \text{this.value} \mapsto t \} \text{set}(x) \{ \text{this.value} \mapsto x \}$

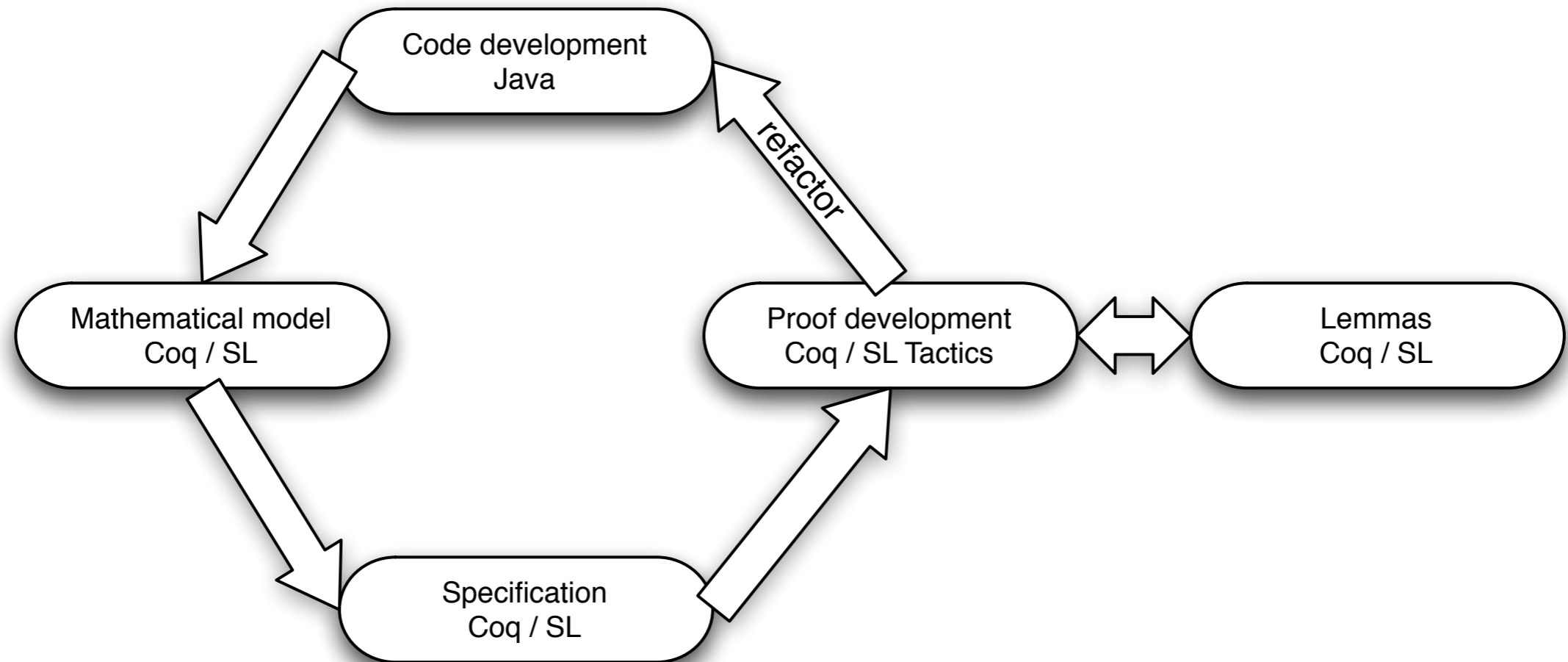
Instantiation for Class Cell

- $T = \text{val}$ $\text{Repr} = \lambda c, v. c.\text{value} \mapsto v$
- $\text{fget} = \lambda v. v$ $\text{fset} = \lambda _, v. v$
- $\{ \text{this.value} \mapsto t \} \text{get}() \{ \text{ret}, \text{this.value} \mapsto t \wedge \text{ret} = t \}$
- $\{ \text{this.value} \mapsto t \} \text{set}(x) \{ \text{this.value} \mapsto x \}$
- $\{ \text{true} \} \text{new Cell}() \{ \text{ret}, \text{ret.value} \mapsto 0 \}$

Proof tools

- Either generate source stubs from specification
- Or proof on bytecode level
- No support for refactoring
- Rarely support for incremental proofs

Kopitiam: workflow



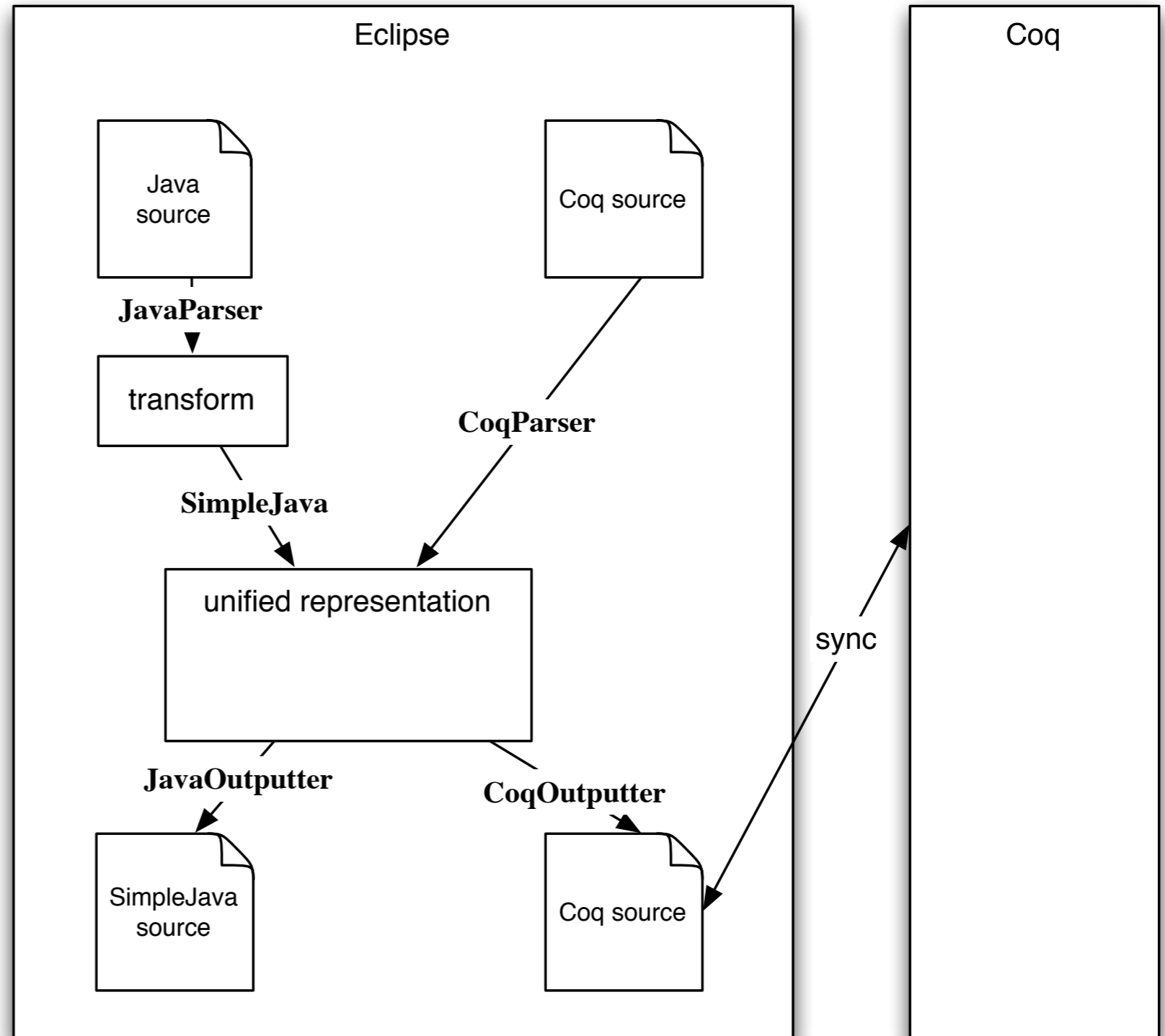
- Develop Java code and proof correctness side-by-side
- Enables refactoring of Java while proofing
- Slightly modified code might be easier to proof

Screenshots

- Demonstration

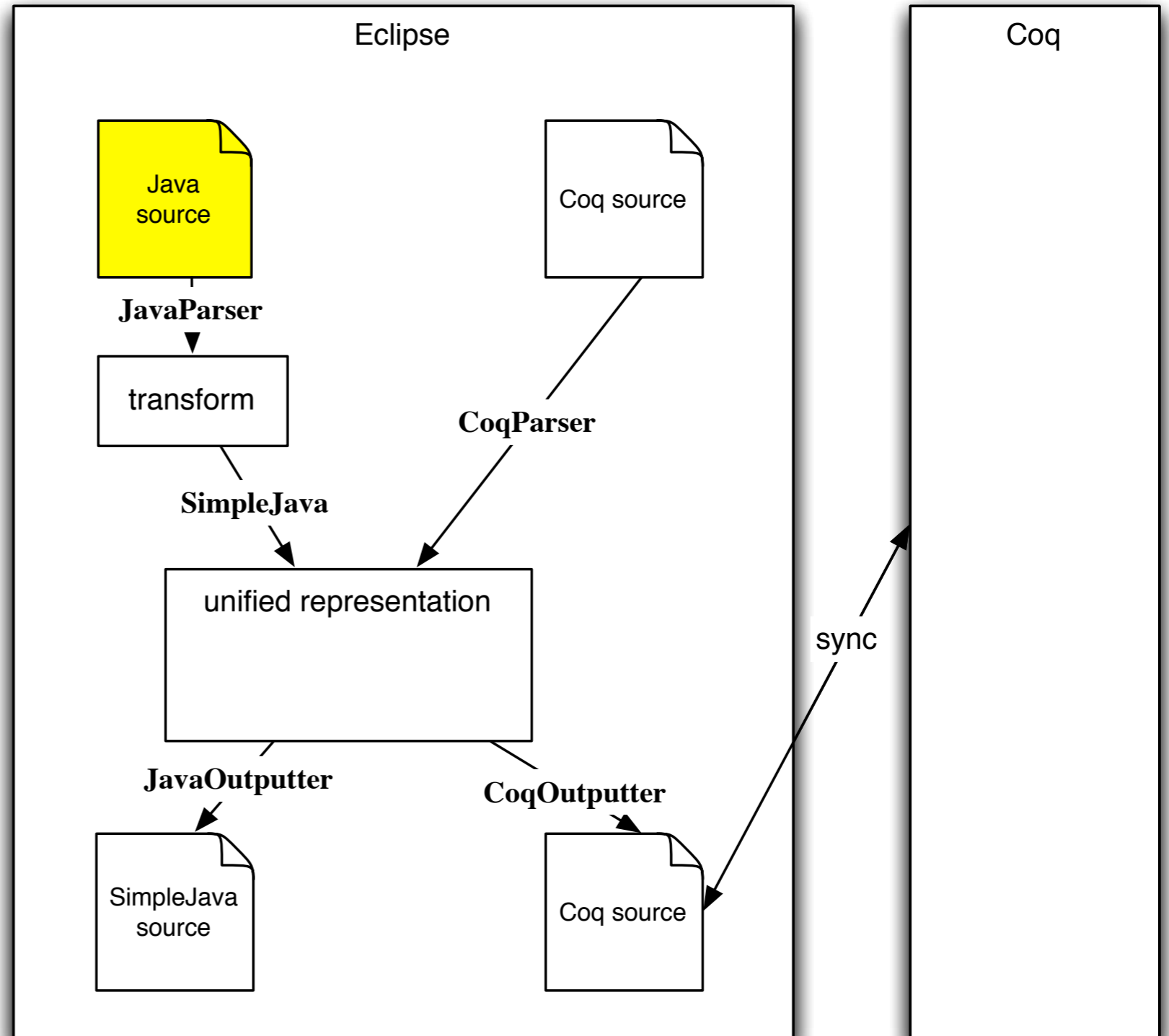
Kopitiam challenges

- Keep state in sync
- Modification can imply retract of Coq proof



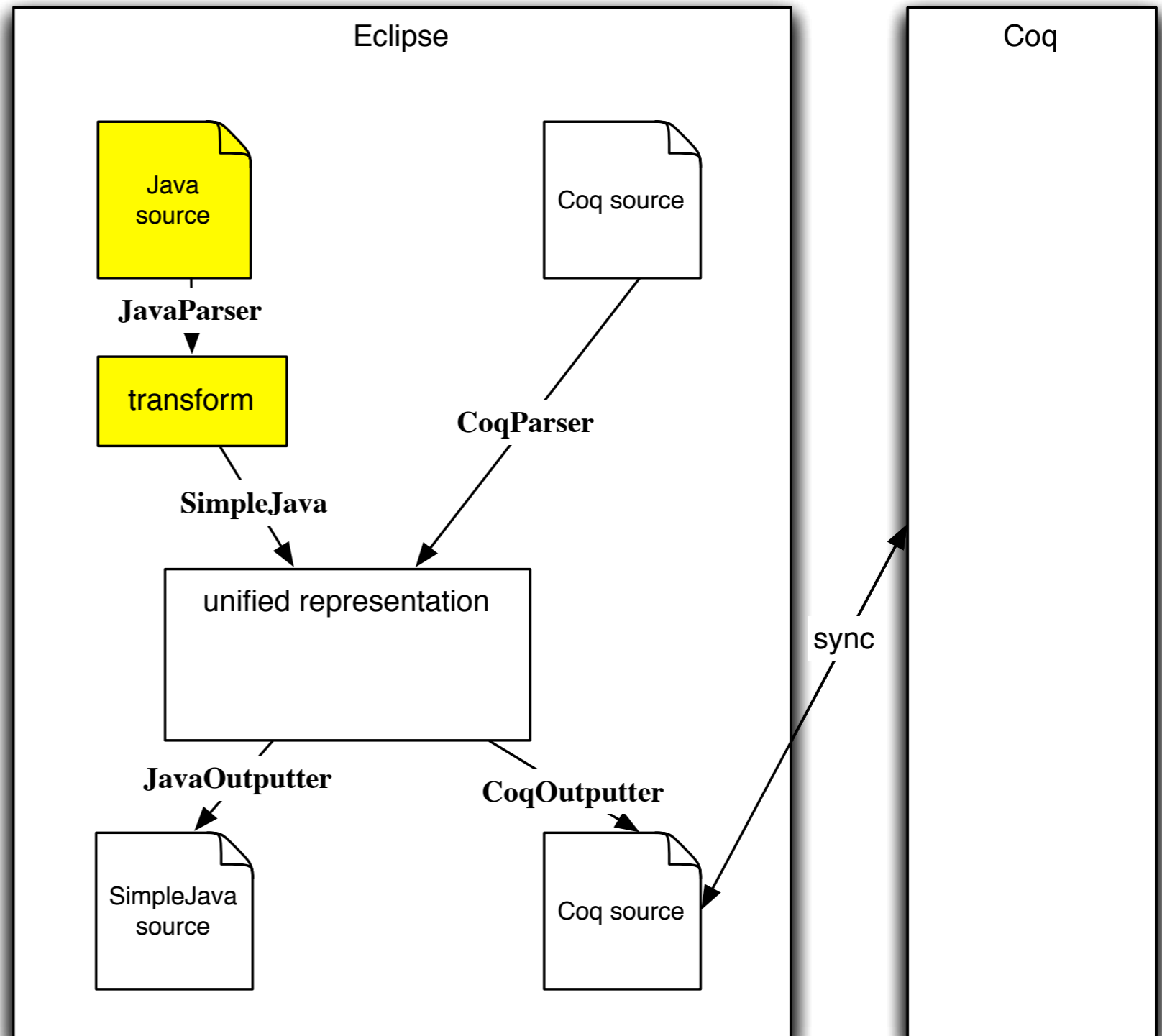
Kopitiam challenges

- Keep state in sync
- Modification can imply retract of Coq proof



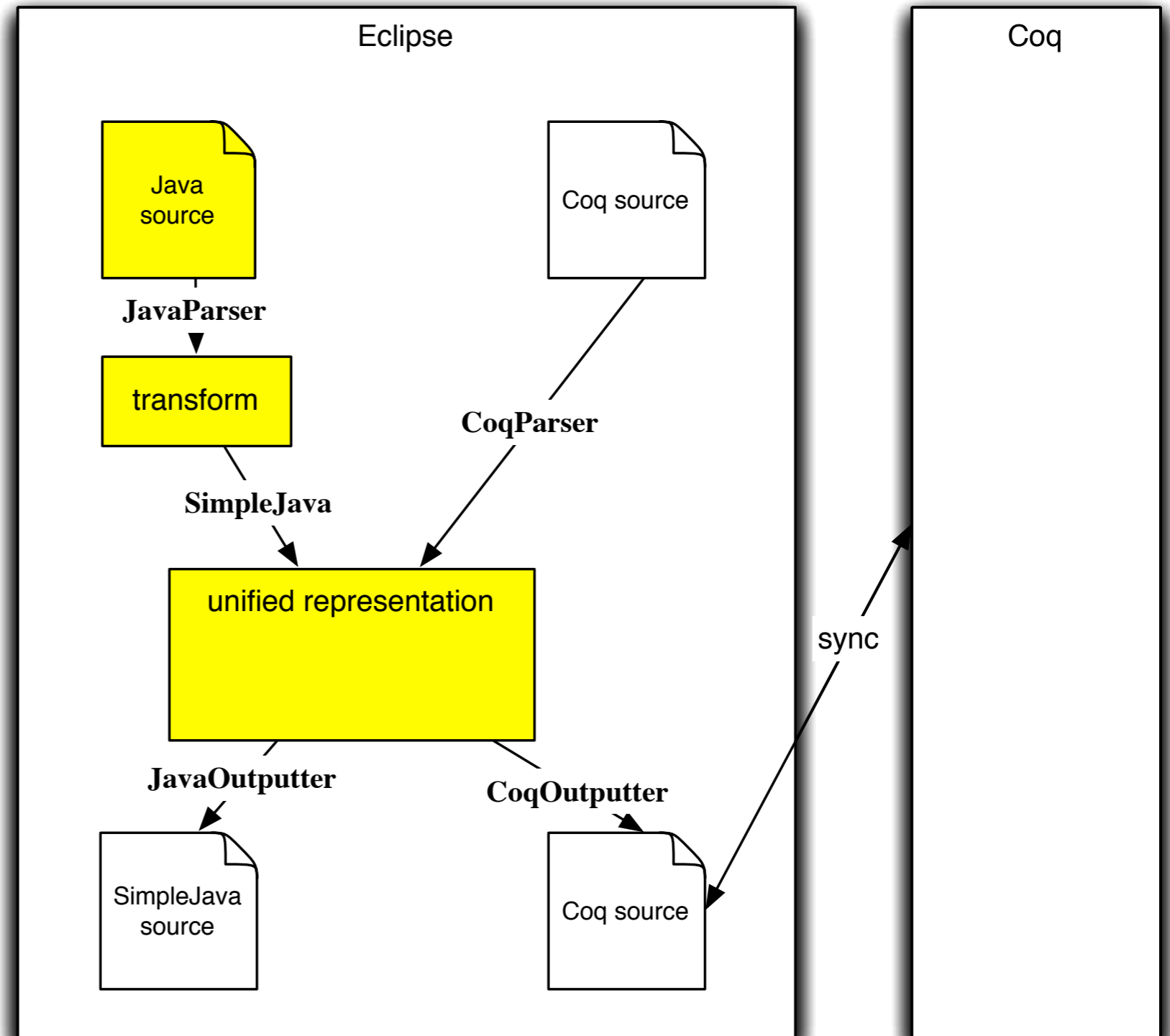
Kopitiam challenges

- Keep state in sync
- Modification can imply retract of Coq proof



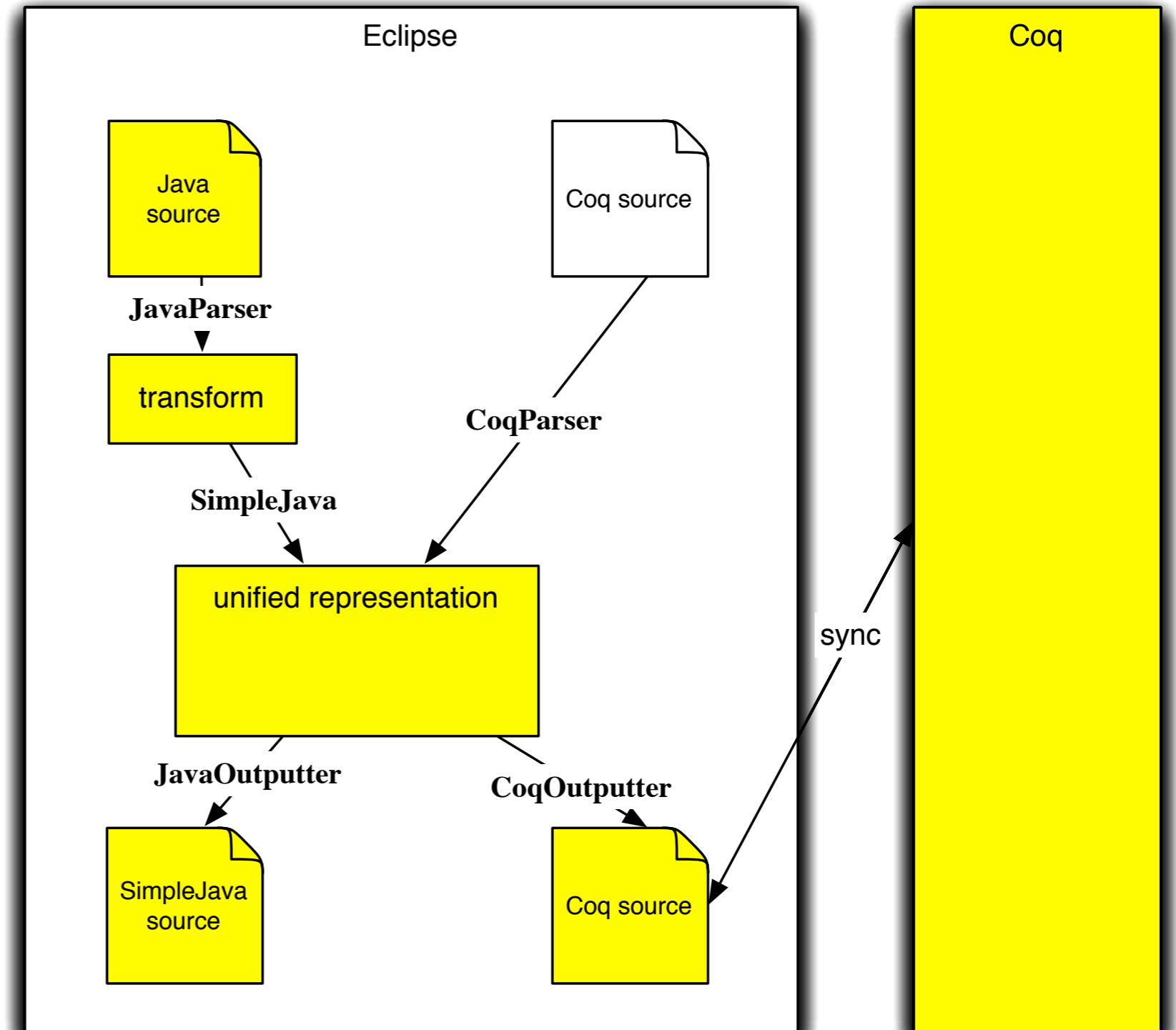
Kopitiam challenges

- Keep state in sync
- Modification can imply retract of Coq proof



Kopitiam challenges

- Keep state in sync
- Modification can imply retract of Coq proof



Future Work

- Integration into a single (JDT) editor
- Case studies
 - Snapshottable Trees

Thanks

- Development <http://github.com/hannesm/Kopitiam>
- Eclipse update-site <http://www.itu.dk/~hame/kopitiam/>
- Tools and Methods for Scalable Software Verification
[http://www.itu.dk/research/pls/wiki/index.php/
Tools_and_Methods_for_Scalable_Software_Verification_\(TOMESO\)](http://www.itu.dk/research/pls/wiki/index.php/Tools_and_Methods_for_Scalable_Software_Verification_(TOMESO))