XE: Bridging the Aspect-Oriented Programming Usability Gap

Wiwat Ruengmee, Roberto S. Silva Filho, Sushil Bajracharya, David Redmiles, and Cristina Lopes

ISR Institute for Software Research, UNIVERSITY OF CALIFORNIA, IRVINE

UCIrvine University of California, Irvine

(Introduction)

AOP developers lack adequate support for:

- visualizing and identifying the exact points in the code where aspects are woven
- preventing aspect-base code inconsistencies
- evolving aspect-oriented code in a coherent way

(Prototype)

XE was developed as an extension to PLT Scheme and DrScheme IDEs. XE IDE integrates components around a common relational model, providing meta-interpreters that are responsible for combining aspect and base code in a single view.

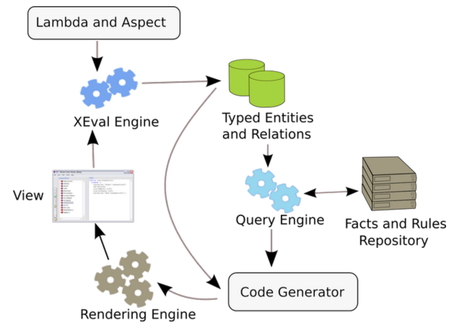

Figure 1: XE architecture

(Future Work)

Future work includes extending XE program model and approach to other programming paradigms, e.g. Java and AspectJ, as an Eclipse plug-in. We also aim at supporting parallel development, when different programmers build a common piece of AOP software.

(Example: Banking API)

Base code

```
(define (deposit)
    (lambda ()
      (run-deposit 2000)))
(define (withdraw)
    (lambda ()
      (run-withdraw 1000)))
(define (balance)
    (lambda ()
      (run-show-balance)))
```

Evolve

```
(define (balance)
    (lambda ()
      (run-show-balance)
      (run-print-balance)))
```

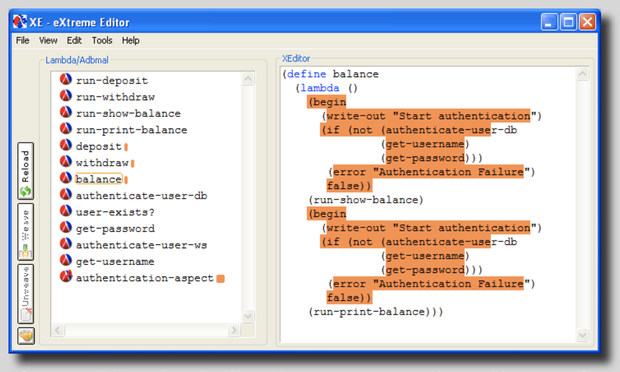

Figure 2: XE main window shows woven code and base code

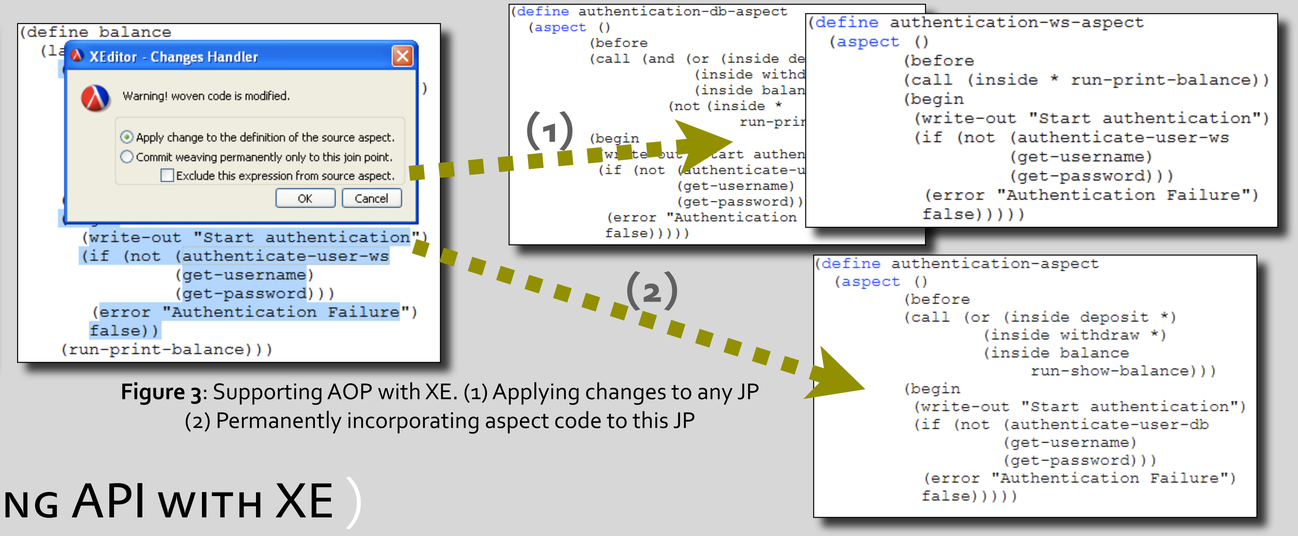

Figure 3: Supporting AOP with XE. (1) Applying changes to any JP (2) Permanently incorporating aspect code to this JP

(Evolving banking API with XE)

Aspect code

```
(define authentication-db-aspect
  (aspect ()
    ((before
      (call
        (or
          (inside deposit *)
          (inside withdraw *)
          (inside balance *)))
      (begin
        (if (not (authenticate-user-db
                   (get-username)
                   (get-password)))
            (error “Authentication Failure”)
             false))))))
```

Modify

```
(define authentication-db-aspect
  (aspect ()
    ((before
      (call
        (or
          (inside deposit *)
          (inside withdraw *)
          (inside balance run-show-balance)))
      (begin
        (if (not (authenticate-user-db
                   (get-username)
                   (get-password)))
            (error “Authentication Failure”)
             false))))))
```

Create by copy-and-paste

```
(define authentication-ws-aspect
  (aspect ()
    ((before
      (call
        (or
          (inside deposit *)
          (inside withdraw *)
          (inside balance run-print-balance)))
      (begin
        (if (not (authenticate-user-ws
                   (get-username)
                   (get-password)))
            (error “Authentication Failure”)
             false))))))
```

For more information about XE and its prototype, please contact one of the authors at **{wruengme, rsilvafi, sbajrach, redmiles, lopes}@ics.uci.edu**