Layered Programming

A Language Independent Variability Management Approach

Joey De Pauw

Universiteit Antwerpen

Table of Contents

Motivation

- Software product line engineering (SPLE)
- High up-front investment
- Proactive, reactive and extractive SPLE

Motivation

- Software product line engineering (SPLE)
- High up-front investment
- Proactive, reactive and extractive SPLE

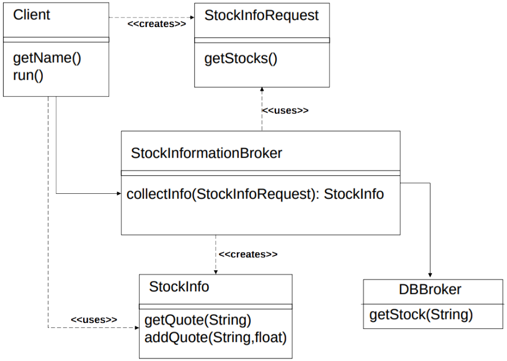
Need for

- Tool support
- Techniques for domain implementation

Related Work Example

```
class Client {
    String getName(){...}
    void run(){...}
}
class StockInformationBroker {
    StockInfo collectInfo(
        StockInfoRequest r){...}
}
class DBBroker {
    Stock getStock() {...}
}
class StockInfoRequest {
    StockInfo getStocks(){...}
}
```

Add pricing functionality



Example

```
class Client {
    String getName(){...}
    void run(){...}
    float balance;
    float balance() { return balance; }
    void charge(StockInfoRequest r) { balance -= r.price(); }
}
class StockInformationBroker {
    StockInfo collectInfo(Client c, StockInfoRequest r){
        ...
        c.charge(r);
    }
}
class DBBroker {
    Stock getStock() {...}
}
class StockInfoRequest {
    StockInfo getStocks(){...}
    float price() { return basicPrice() + calculateTax(); }
    float basicPrice() { return 5 + getStocks().length*0.2; }
    float calculateTax() { ... }
}
```

Example
Preprocessor Directives

```
class Client {
    String getName(){...}
    void run(){...}
#ifdef PRICING
    float balance;
    float balance() { return balance; }
    void charge(StockInfoRequest r) { balance -= r.price(); }
#endif
}
class StockInformationBroker {
    StockInfo collectInfo(Client c, StockInfoRequest r){
        ...
#ifdef PRICING
        c.charge(r);
#endif
    }
}
class DBBroker {
    Stock getStock() {...}
}
...
```

Example Architecture: Mixins

```
class Client {
    String getName(){...}
    void run(){...}
}

template <class Super>
class Pricing: public Super{
    float balance;
    float balance() { return balance; }
    void charge(StockInfoRequest r) { balance -= r.price(); }
};

typedef Pricing<Client>ClientPricing;
...
```

Example
Aspect-oriented Programming

```
aspect Pricing {
    private float Client.balance;
    float Client.balance() { return balance; }
    void Client.charge(StockInfoRequest r) {
        balance -= r.price();
    }
    float StockInfoRequest.price() { ... }
    after( Client c, StockInfoRequest request ):
    (call (StockInfo collectInfo(StockInfoRequest))
    && this(c) && args(request)) {
        c.charge(request);
    }
}
```

Example
Feature-oriented Programming (AHEAD)

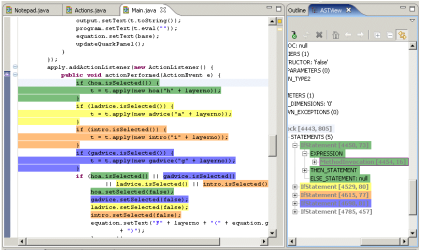
```
refines class Client {
    float balance;
    public float balance() { return balance; }
    void charge(StockInfoRequest r) { balance -= r.price(); }
}
refines class StockInfoRequest {
    float price() { return basicPrice() + calculateTax(); }
    float basicPrice() { return 5 + getStocks().length*0.2; }
    float calculateTax() { ... }
}
refines class StockInformationBroker {
    StockInfo collectInfo(Client c, StockInfoRequest r) {
        super.collectInfo(c, r);
        c.charge(r);
    }
}
```

Example
Delta-oriented Programming

```
delta DClient{
    modifies class Client {
        adds float balance;
        adds float balance() { return balance; }
        adds void charge(StockInfoRequest r) { balance -= r.price(); }
    }
    modifies class StockInformationBroker {
        adds StockInfo collectInfo()(Client c, StockInfoRequest r) {
            collectInfo(r);
            c.charge(r);
        }
    }
}
```

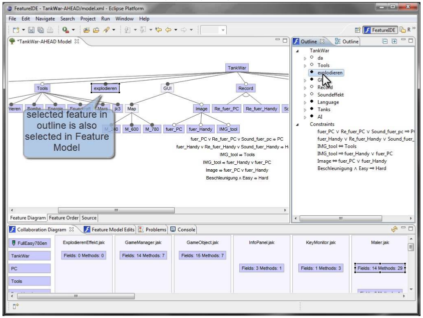
Example
Tool Support

CIDE



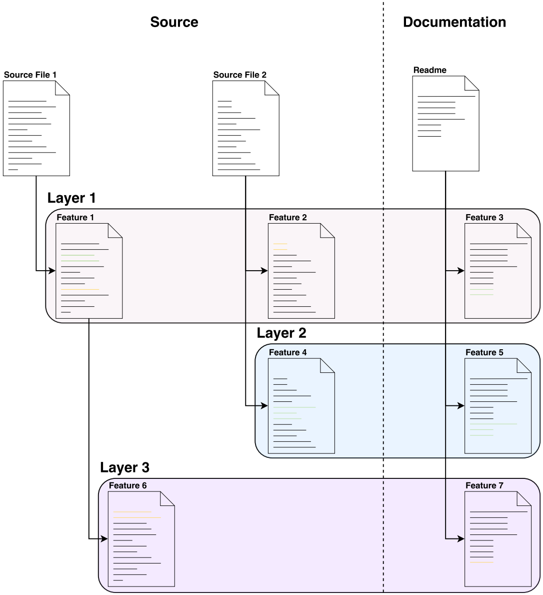
```
                output.setText(t.toString());
                program.setText(t.eval(""));
                equation.setText(base);
                updateQuarkPanel();
            }
        });
        apply.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (hoa.isSelected()) {
                    t = t.apply(new hoa("h" + layerno));
                }
                if (ladvice.isSelected()) {
                    t = t.apply(new advice("a" + layerno));
                }
                if (intro.isSelected()) {
                    t = t.apply(new intro("i" + layerno));
                }
                if (gadvice.isSelected()) {
                    t = t.apply(new gadvice("g" + layerno));
                }
                if (hoa.isSelected() || gadvice.isSelected()
                        || ladvice.isSelected() || intro.isSelected()
                    hoa.setSelected(false);
                    gadvice.setSelected(false);
                    ladvice.setSelected(false);
                    intro.setSelected(false);
                    equation.setText("F" + layerno + "(" + equation.g
                            + ")");
```

FeatureIDE



Layered Programming

- Delta-oriented programming
- Only tool based
  - diff-match-patch
  - Feature model
  - CLI like git
  - Editor


Source
Documentation
Source File 1
Source File 2
Readme
Layer 1
Feature 1
Feature 2
Feature 3
Layer 2
Feature 4
Feature 5
Layer 3
Feature 6
Feature 7

Layered Programming Demo

Layered Programming
Benefits

- System has multiple representations $\rightarrow$ keep them consistent
- Language independent
- Easy to use
  - No new language or complex structure
  - Semantically clear (WYSIWYG)
  - Features added where they are used
  - No intermediary representation
- Robuust (to be verified)
- Proactive, reactive and extractive SPLE

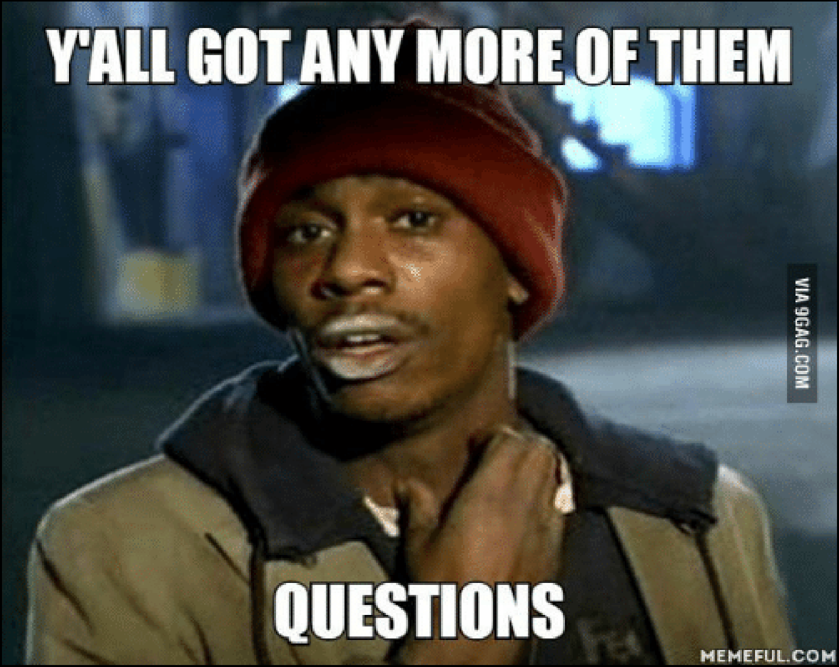
Layered Programming
Risks

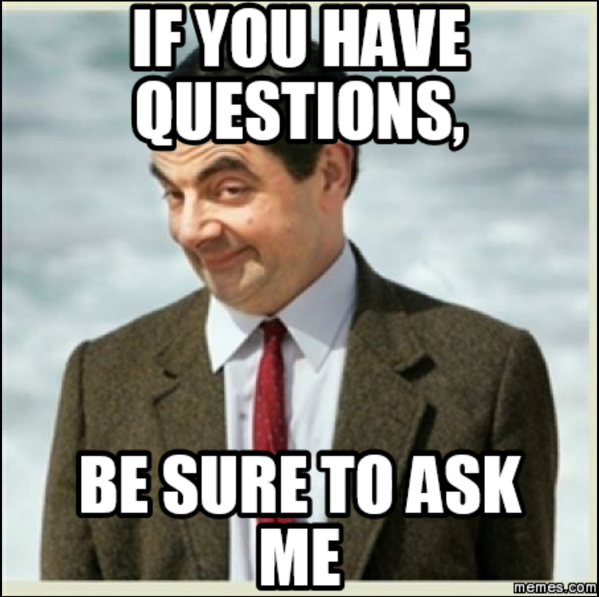
- Interaction with version management (git)
- Is it robust under changes to base
- Can conflicting features be detected?
- What should happen with conflicting features?
- How to deal with optional layer interactions?

Planning

- Analyze risks and propose solutions/workarounds
- Algorithm and tool to support layered programming
- Encoding for layers
- Feature model support

BRACE YOURSELVES
FOR A FEW QUESTIONS
memeshappen.com


Y'ALL GOT ANY MORE OF THEM
QUESTIONS
VIA 9GAG.COM
MEMEFUL.COM


IF YOU HAVE QUESTIONS,
BE SURE TO ASK ME
memes.com