

Могућности примене напредне модуларизације софтвера

Валентино Вранић

Slovak University of Technology, Bratislava, Slovakia

Пословни факултет у Ваљеву, Универзитет Сингидунум

vranic@fiit.stuba.sk

<http://fiit.stuba.sk/~vranic/>

Мрежа 2010 – Употреба савремених интернет технологија

Ваљево

4. јун 2010.

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

Раздвајање интереса

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

- ▶ Separation of concerns¹

focussing one's attention on some aspect

- ▶ Интерес (concern) је ствар којој посвећујемо пажњу
- ▶ Раздвајање интереса значи да се у једном тренутку посвећује сва пажња једном интересу
- ▶ У програмирању се тежи што бољој **локализацији** интереса
- ▶ Један од начина је **модуларна декомпозиција**

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

¹Е. W. Dijkstra. On the role of scientific thought, 1974 (EWD 447).
<http://www.cs.utexas.edu/users/EWD/ewd04xx/EWD447.PDF>

Модуларна декомпозиција

- ▶ Модуларне јединице: модули, компоненте, објекти, функције/процедуре итд.
- ▶ Хијерархијски организоване
- ▶ Циљ је висока кохезија унутар модуларне јединице, а слаба спрега с осталим модуларним јединицама

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

Пресецајући интереси

- ▶ Софтвер моделира стварни свет
- ▶ А у стварном свету су ретке потпуне хијерархије²
- ▶ Долази до спреге међу модуларним јединицама које припадају различитим хијерархијама
- ▶ Испољавају се интереси које није могуће модуларизовати: *пресецајући интереси* (crosscutting concerns)
- ▶ Последица је преплетен и разбацан код
- ▶ Нпр. методе апликационе логике често садрже и код који се тиче других интереса, као што је логирање, безбедност, перзистенција, аутентификација. . .
- ▶ Имплементација ових интереса је разбацана у много метода уз понављање кода

²T. Reenskaug. Working With Object: The OOram Software Engineering Method. Prentice Hall, 1995.

Аспектно-оријентисано програмирање

- ▶ Две линије: прагматична и истраживачка
- ▶ Прагматична линија – преовлађујуће асиметричан приступ
 - ▶ Посебне конструкције – *аспекти* – утичу на главни, објектно-оријентисани или процедурални код
 - ▶ Типичан пример: обухватање позива метода које одговарају одређеној сигнатури и извршење одређеног кода пре, после или уместо ових позива

```
aspect RangeControl {  
    void around(int x): call(void Point.setX(..)) && args(x) {  
        if (x < 0)  
            proceed(640 + x % 640);  
        else if (x > 639)  
            proceed(x % 640);  
        else  
            proceed(x);  
    }  
    ...  
}
```

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

- ▶ Најзаступљенији аспектно-оријентисани језик је AspectJ³
- ▶ Заснован је на језику Java и има подршку у радним окружењима Eclipse, JBuilder и NetBeans
- ▶ Стабилан је и користи се у пракси – нпр. обезбеђује изолацију функција различитих издања у IBM WAS⁴
- ▶ Низ других језика опонаша AspectJ: AspectC++, CeasarJ, AspectS. . .
- ▶ Подршка и на нивоу софтверских оквира: AspectWerkz, Spring, JBoss, Seasar. . .

³<http://eclipse.org/aspectj/>

⁴http://gateway.comp.lancs.ac.uk:8080/c/portal/layout?p_1_id=1394

Симетричан приступ

- ▶ У симетричном приступу програм представља композицију аспеката
- ▶ Не тиче се нужно само модуларизације пресецајућих интереса
- ▶ Аспекти представљају различите погледе на исте ентитете
- ▶ Правила спајања су дефинисана изван аспеката
- ▶ Главни репрезентант: језик Hyper/J – прототип који је развио IBM
- ▶ Утицај и на AspectJ: аспекти могу да додају нове методе, атрибуте и односе наслеђивања
- ▶ Симетричан приступ подесан у анализи и дизајну (Theme/UML, али и OOram)

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

Очување случајева коришћења

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

- ▶ Случајеви коришћења су врло погодни као јединица функционалности софтвера у комуникацији с клијентом
- ▶ Нажалост, у имплементацији се губе
- ▶ Могу се очувати ако се имплементирају у виду аспеката⁵

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

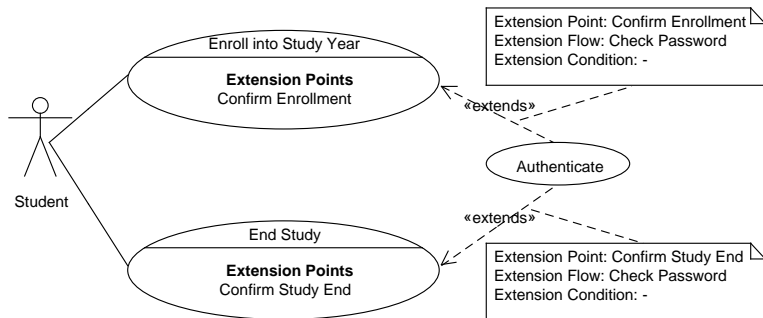
Реализација
измена
аспектима

Сумаризација

⁵ I. Jacobson and P. W. Ng. *Aspect-Oriented Software Development with Use Cases*. Addison-Wesley, 2004.

Проширујући случајеви коришћења

- ▶ Проширујући случајеви коришћења пресецају основну (другу) функционалност – асиметрична декомпозиција



Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

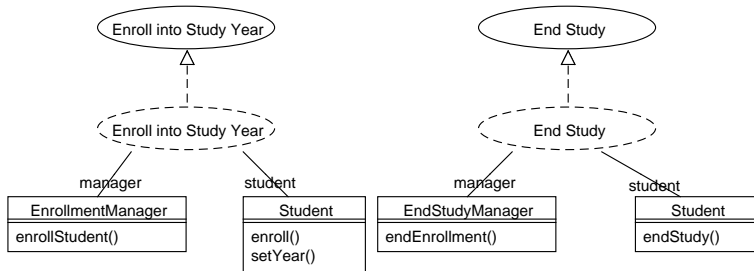
Сумаризација

Случајеви коришћења на истом нивоу

Могућности примене напредне модуларизације софтвера

Валентино Вранић

- ▶ Случајеви коришћења на истом нивоу се реализују над истим класама – симетрична декомпозиција



Проблем традиционалне модуларизације

Подршка на нивоу језика

Очување случајева коришћења

Аспекти и варијабилност

Реализација измена аспекта

Сумаризација

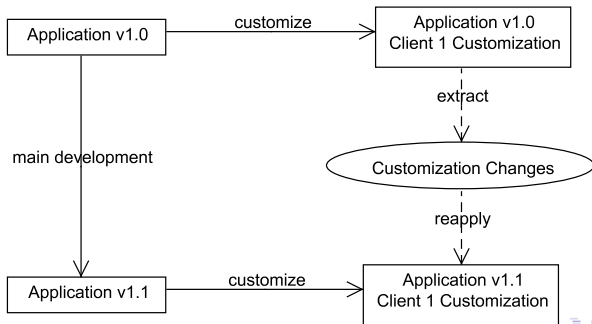
Аспекти и варијабилност

- ▶ Прикључење и искључење функција дефинисаних аспекта се реализује самим њиховим присуством или одсуством – без потребе модификације других делова софтвера
- ▶ То је посебно подесно за имплементацију варијабилних делова софтвера
- ▶ Могуће је лако реализовати више верзија софтвера и вршити реконфигурацију по потреби
- ▶ Примена у редовима софтверских производа (software product lines)⁶

⁶ J. Kohut and V. Vranić. Guidelines for Using Aspects in Product Lines. In Proc. of 8th International Symposium on Applied Machine Intelligence and Informatics, SAMI 2010, January 2010, Herlany, Slovakia, IEEE.

Реализација измена аспектима

- ▶ Одржавање софтвера представља његову најдужу и најскупљу фазу
- ▶ Велики део измена је некорективне природе: захтеви за проширење функционалности или прилагођавање новом контексту
- ▶ Понекад је потребно издвојити измене или их применити на другу верзију истог софтвера



Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

Проблем
традиционалне
модуларизације

Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

Реализација измена аспектима

Могућности
примене
напредне
модуларизације
софтвера

Валентино
Вранић

- ▶ Развијен је приступ за аспектно-оријентисану реализацију измена⁷
- ▶ Заснован је на разликовању два нивоа типова реализације измена:
 - ▶ Спецификациони типови измена – идентификација типа реализације измене у самом захтеву за измену
 - ▶ нпр. увођење резервног ресурса
 - ▶ Реализациони типови измена – схеме програмског кода
 - ▶ нпр. замена класе

Проблем
традиционалне
модуларизације


Подршка на
нивоу језика

Очување
случајева
коришћења

Аспекти и
варијабилност

Реализација
измена
аспектима

Сумаризација

⁷V. Vranić, R. Menkyna, M. Bebjak, and P. Dolog. Aspect-Oriented Change Realizations and Their Interaction. *e-Informatica Software Engineering Journal*, 3(1):43-58, 2009. 

Сумаризација

- ▶ Реална модуларизација се измиче представи потпуне хијерархије
- ▶ Аспектно-оријентисани развој софтвера може да помогне
- ▶ Модуларизација пресецајућих интереса – асиметричан приступ
- ▶ Композиција различитих погледа – симетричан приступ
- ▶ Примена у пракси – AspectJ, AspectWerkz, Spring, JBoss. . .
- ▶ Проблеми: губитак прегледа над кодом, нарушавање енкапсулације и интеракција аспеката
- ▶ Истраживање – AOSD-Europe⁸

<http://fiit.stuba.sk/~vranic/>

⁸ <http://www.aosd-europe.net/>