HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Towards the Design and Implementation of Aspect-Oriented Programming for Spreadsheets

Pedro Maia, Jorge Mendes, Jácome Cunha,
Henrique Rebêlo, João Saraiva

HASLab / INESC TEC, Portugal
Universidade do Minho, Portugal
Universidade Nova de Lisboa, Portugal
Universidade Federal de Pernambuco, Brazil

SEMS 15 / ICSE 2015

May 18, 2015

INESCTEC
TECHNOLOGY & SCIENCE
ASSOCIATE LABORATORY
PORTUGAL

Universidade do Minho

Spreadsheet Usage

Spreadsheet Usage

Problems

- Spreadsheets lack modularity.
- A user may work on different features[1].
- Different users may need distinct features.
- Not all the features may be needed/wanted by a user.
- Some features may be repeated in different places:
  - of the same spreadsheet; and/or,
  - in distinct spreadsheets.

[1] Feature: data and/or computation.

Spreadsheet Usage

Features

- Placed in a specific position (or point) of the spreadsheet.
- Contain data and/or computation.
- Portrait aspects of the spreadsheet.

Aspect-Oriented Programming!

Aspect-Oriented Programming

AOP

- Enables modularity of cross-cutting concerns.
  - E.g., error handling, certain design patterns, tracing, design by contract.
- Obtained through:

  join points parts of the spreadsheet that need some feature; and,

  advice the content of that feature.
- Strong relation to Object-Oriented Programming.
  - Implementations for Java, C++, and other languages.

Aspect-Oriented Programming

Main Characteristics

Join Point Model – the parts of the language that can be altered.

- E.g., call or execution of methods, exception handler execution.

Kinds of Advice – ways to alter the join point.

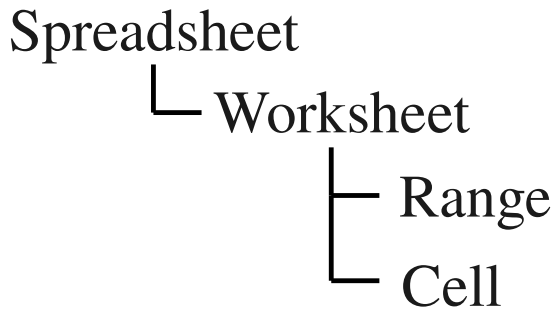
- Usual alterations: **before**, **after**, **around**.

```
public aspect MyAspect {
  pointcut setMethods() : execution(void set(..));
  before() : setMethods() {
    System.out.println("before set method");
  }
}
```

Aspect-Oriented Programming

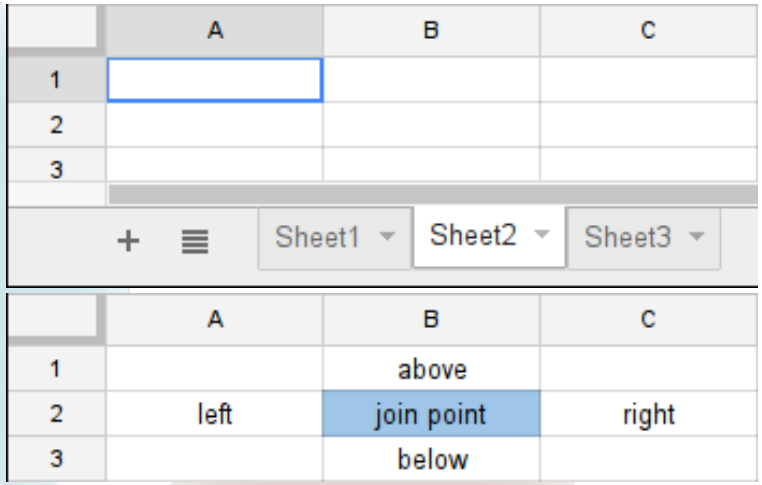
For Spreadsheets

Join Point Model – the parts of the language that can be altered.



Kinds of Advice – ways to alter the join point.

- Worksheet:
  before, **after**, **around**.
- Cell:
  left, **above**, **right**, **below**, **around**.
- Range:
  left, **above**, **right**, **below**, **around**.



HASLab HIGH-ASSURANCE SOFTWARE LABORATORY

Example

	A	B	C	D	E
1		Exam 1	Exam 2	Essay	Final Mark
2	Shaquille Solomon	9.5	9.5	9.5	9.5
3	Grayson Boyd	4.5	3.2	7	4.9
4	Joss Esmond	8	6	9	7.7
5	Callahan Galen	5	4	3	4.0
6	Averill Cal	6	7	6	6.3

	A	B	C	D	E
1		Exam 1	Exam 2	Essay	Final Mark
2	Shaquille Solomon	9.5	9.5	9.5	9.5
3	Grayson Boyd	4.5	3.2	7	5.0
4	Joss Esmond	8	6	9	7.7
5	Callahan Galen	5	4	3	4.0
6	Averill Cal	6	7	6	6.3

Example

	A	B	C	D	E
1		Exam 1	Exam 2	Essay	Final Mark
2	Shaquille Solomon	9.5	9.5	9.5	9.5
3	Grayson Boyd	4.5	3.2	7	4.9
4	Joss Esmond	8	6	9	7.7
5	Callahan Galen	5	4	3	4.0
6	Averill Cal	6	7	6	6.3

How are border line cases handled?

- Add new column with the logic?
  - `=IF(AND(E2>=4.8;E2<5); 5; E2)`

    Do we want to share the logic?
- Substitute the formula for the wanted result?
  - Who remembers the criteria for the substitution?
  - What was the original value?

Example

Aspect

```
aspect BorderlineCase
finalmark : select sheet{*}.column{*}.cell{*}
around finalmark {
    #{cell.result >= 4.8 && cell.result < 5
        ? 5
        : cell.value }
  } when {
   cell.column[0].value = "Final Mark"
  }
end
```

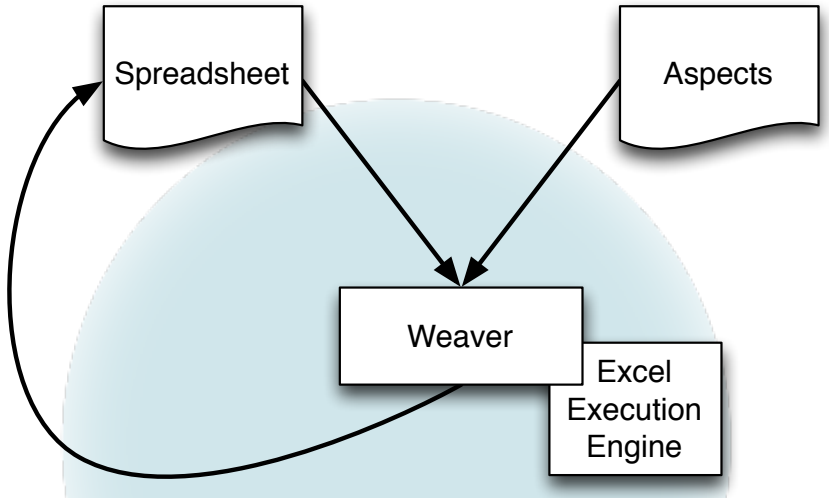
Example

Cell Transformation

```
aspect SmellyAverage
smellyAverage : select sheet{*}.cell{*}
around smellyAverage {
     AVERAGE({r})
   } when {
    cell.value = "SUM({r})/COUNT({r})"
   }
end
```

Weaver

- The order of the application is important.
- Aspects are only applied once.
- The advice of the aspect is evaluated like a normal cell.



Conclusion

- AOP for spreadsheets as a way to modularize them.
  - The concepts of AOP need to be adapted to spreadsheets.
  - Supported by a new textual language and a dynamic weaver.
- New challenges due to the 2-dimensional nature of spreadsheets.
  - Positioning of new cells.
  - Layout deformations.
- A prototype is being implemented.
- A visual language is being designed as an alternative.

Open Questions

- Are end users able to understand spreadsheets + AOP?
- Does AOP bring benefits to spreadsheet development?
- Is the current textual representation adequate?

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Towards the Design and Implementation of Aspect-Oriented Programming for Spreadsheets

Pedro Maia, Jorge Mendes, Jácome Cunha,
Henrique Rebêlo, João Saraiva

HASLab / INESC TEC, Portugal
Universidade do Minho, Portugal
Universidade Nova de Lisboa, Portugal
Universidade Federal de Pernambuco, Brazil

SEMS 15 / ICSE 2015

May 18, 2015

INESCTEC
TECHNOLOGY & SCIENCE
ASSOCIATE LABORATORY
PORTUGAL

Universidade do Minho