Third Turkish Aspect-Oriented Software Development Workshop

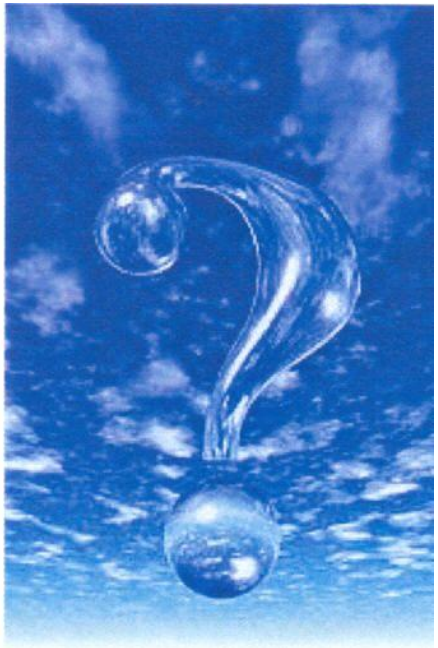
Bilkent University, Ankara, Turkey
December 26, 2008

Bedir Tekinerdoğan

Short Introduction to AOSD

Short Survey...

- Who has ever heard of Aspect-Oriented Software Development ?
- Who has ever used it ?
- Who is planning to use it?



Separation of Concerns

Concern

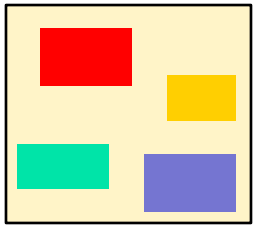
- properties or area of interest

Separation of Concerns Principle

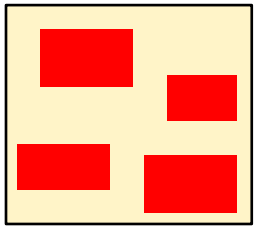
- Identification of the various distinct concerns
- Assignment of single concerns to single modules
- … to support quality factors

Cohesion and Coupling

- **Cohesion**
  - parts of a module are grouped because they all contribute to a single concern of the module

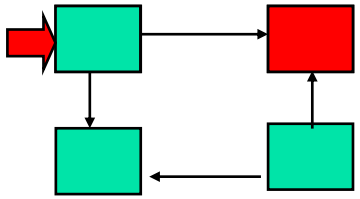


Low Cohesive Module

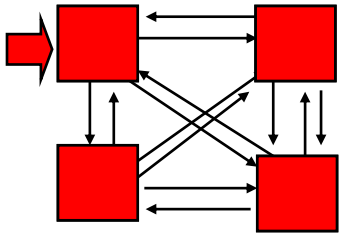


High Cohesive Module

- **Coupling**
  - the degree to which each program module relies on each one of the other modules



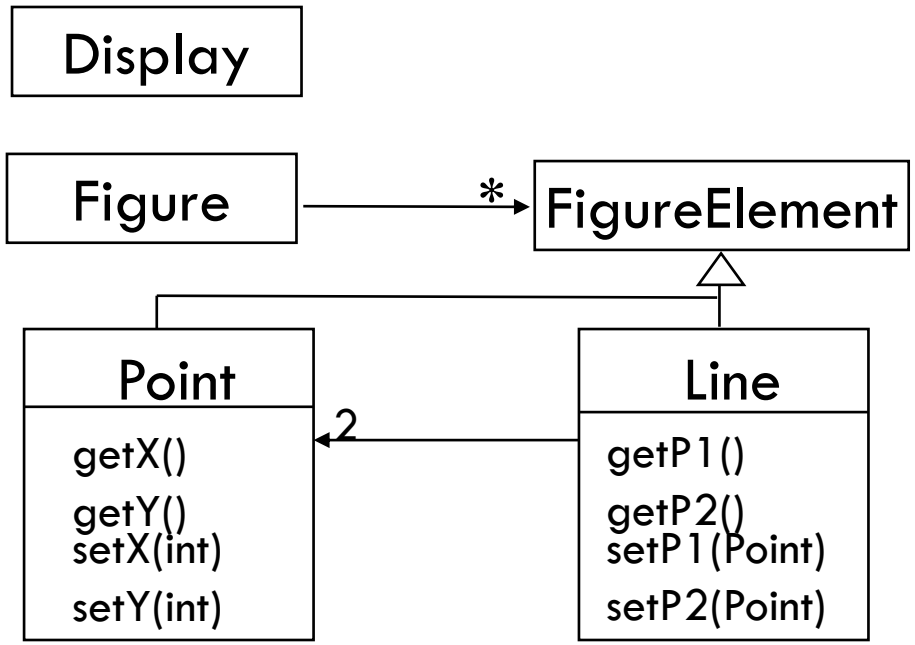
Loosely Coupling



High Coupling

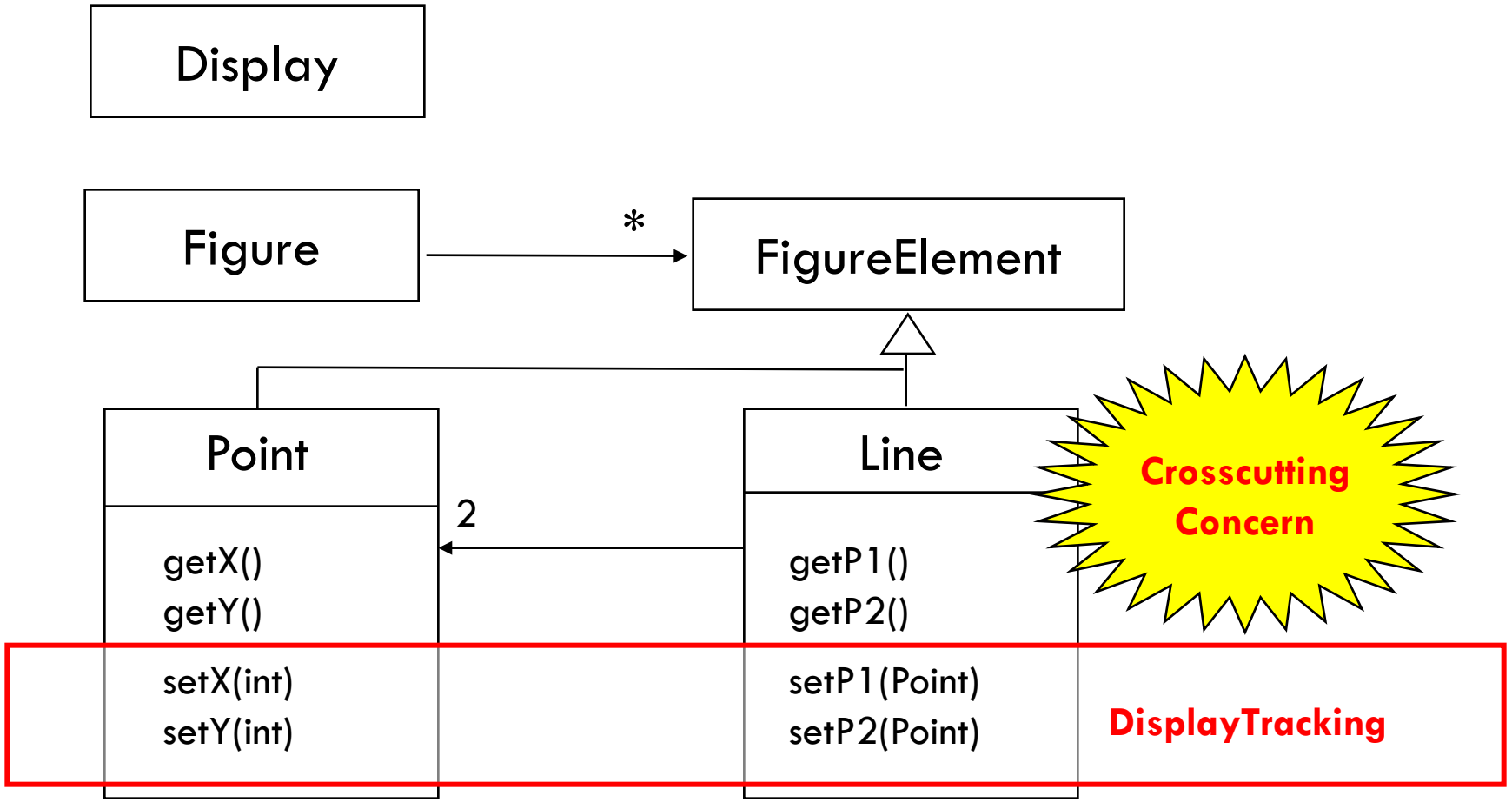
Figure Editor Example

A *figure* consists of several *figure elements*. A figure element is either a *point* or a *line*. Figures are drawn on *Display*. A point includes X and Y coordinates. A line is defined as two points.

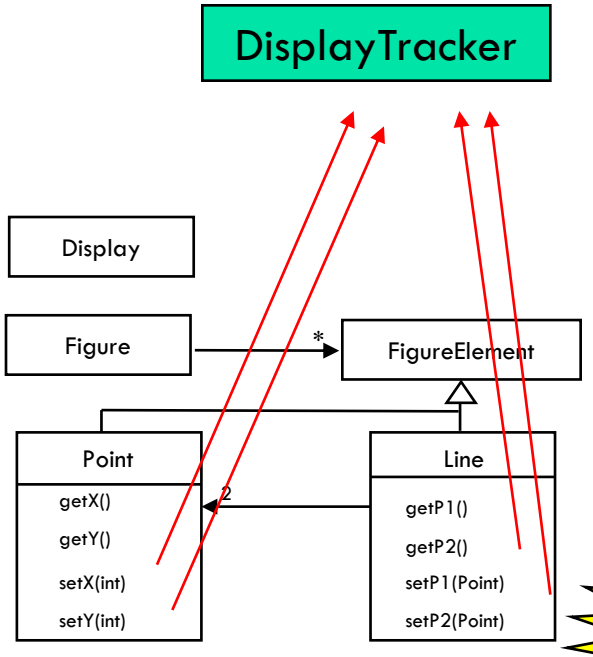


Update Display if elements move → Add Concern tracking

Crosscutting Concern - example



Example - Tracing




```
class DisplayTracker {

 static void updatePoint(Point p)
 {
            this.display(p);
            ....
 }
 static void updateLine(Line l)
 {
            this.display(l);
            ....
```

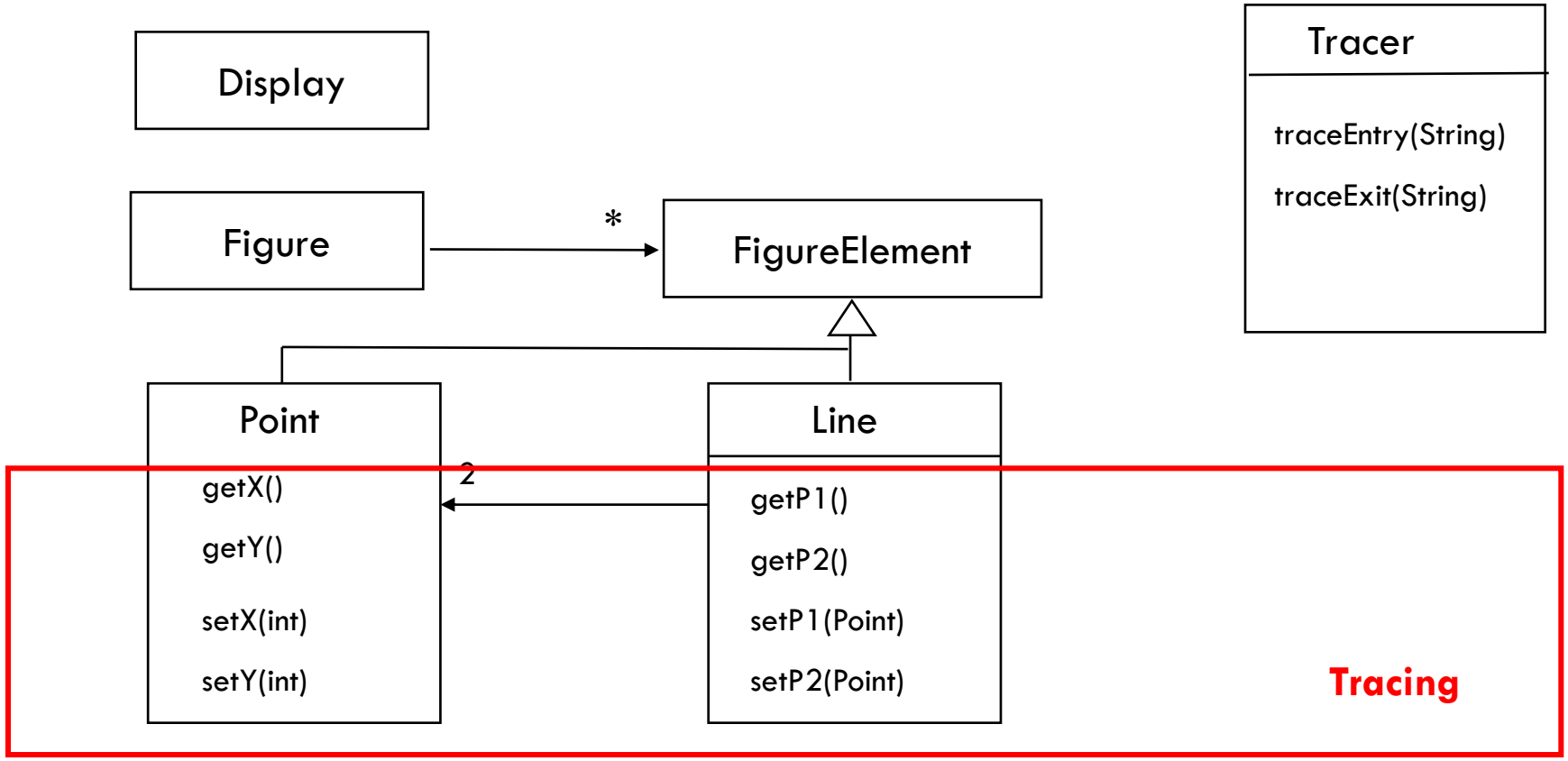
Scattered Concern

```
class Point {
  void set(int x) {
  DisplayTracker.updatePoint(this);
    _x = x;

}
}
```

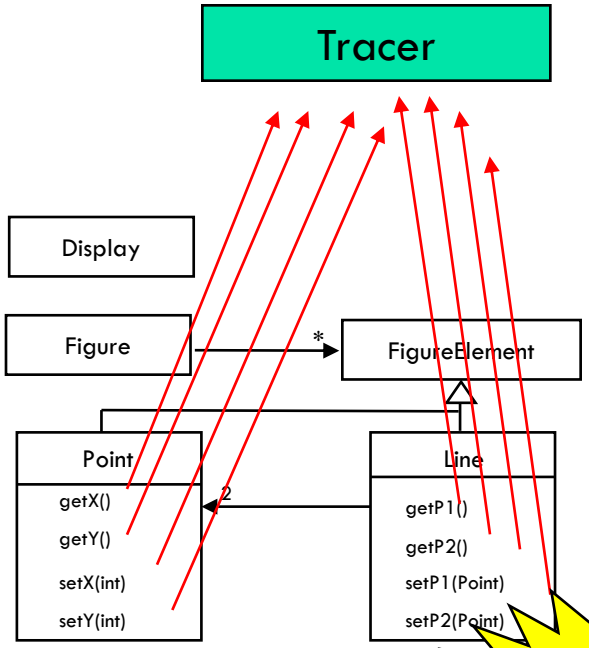
```
 Line {
 void setP1(Point p1 {
  DisplayTracker.updateLine(this);
   _p1 = p1;

}
}
```

Tangling Code

Example - Tracing



Example: Tracing




```
class Tracer {

  static void traceEntry(String str)
  {
            System.out.println(str);
  }
  static void traceExit(String str)
  {
            System.out.println(str);
  }
}
```

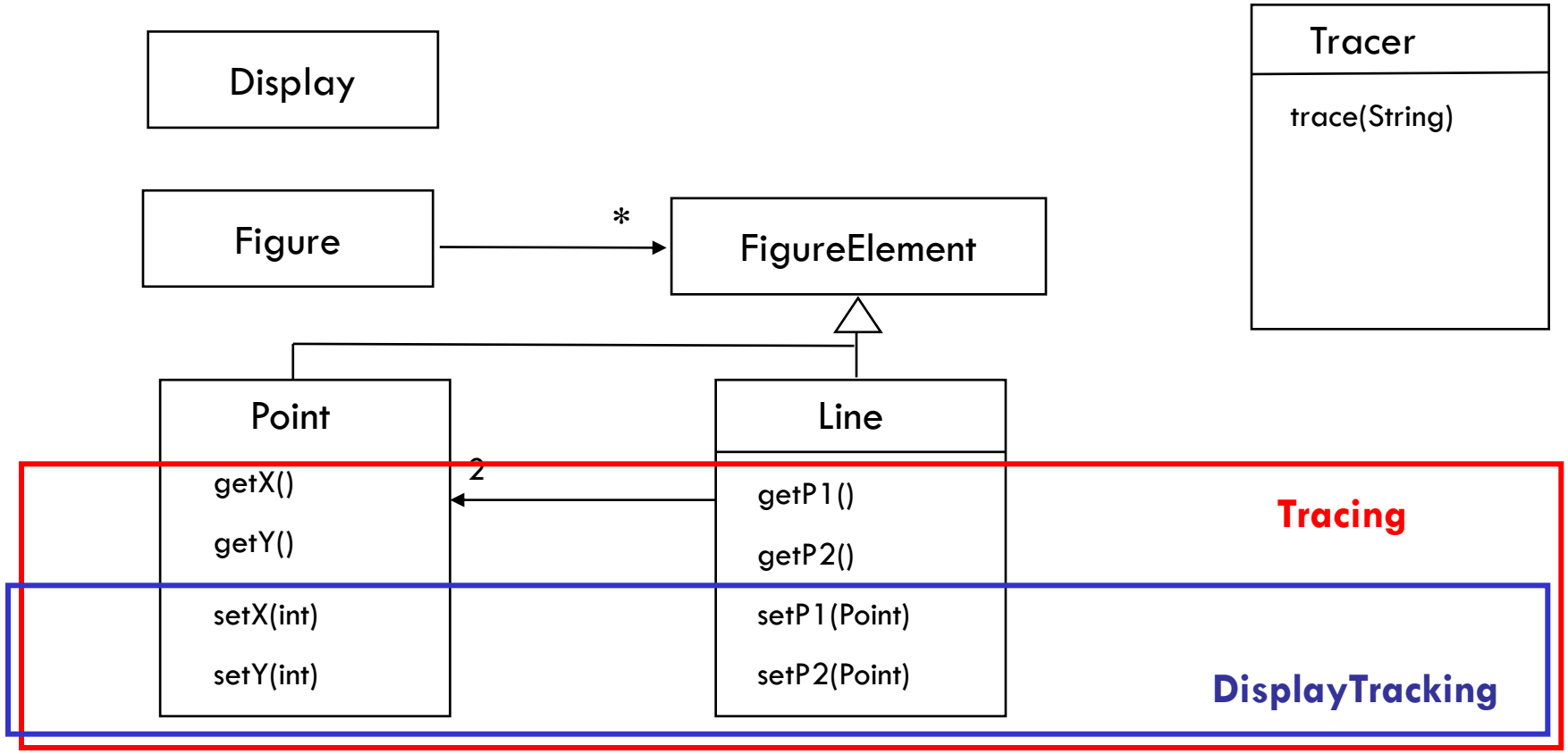
Tangling Code

Scattered Concern

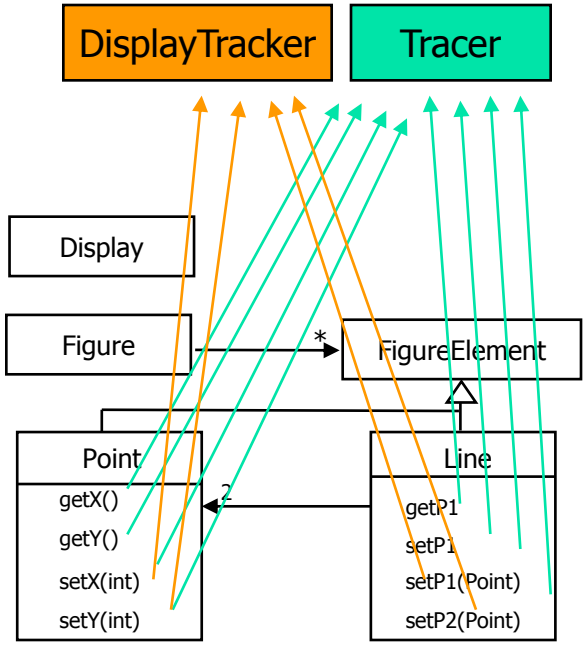
```
class Point {
  void set(int x) {
   Tracer.traceEntry("Point.set");
     _x = x;
       Tracer.traceExit("Point.set");
}
}
```

```
class Line {
  void setP1(Point p1) {
   Tracer.traceEntry("Line.set");
    _p1 = p1;
       Tracer.traceExit("Line.set");
}
}
```

Example – Tracing and Display Tracking



Example: Tracing and Display Tracking

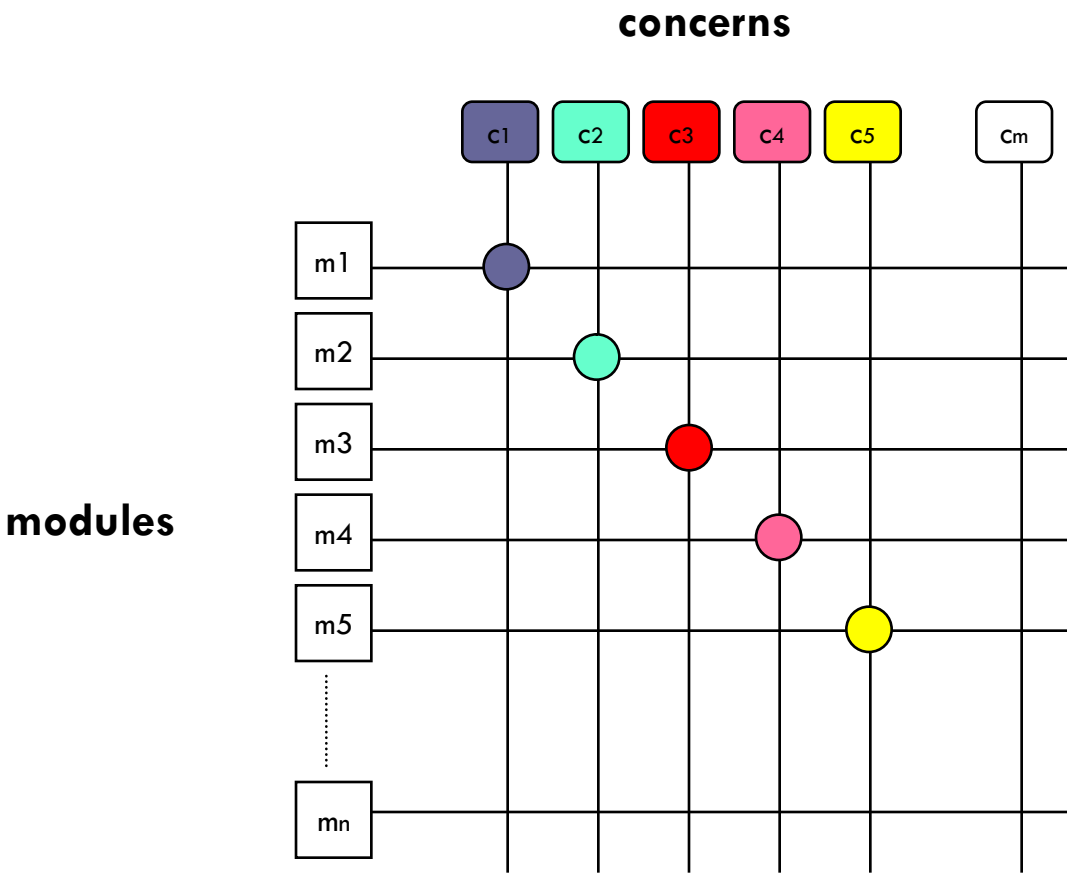



```
class Point {
  void setX(int x) {
     Tracer.traceEntry("Entry Point.set");
      _x = x;
     Tracer.traceExit("Exit Point.set");
     DisplayTracker.updatePoint(this);
}
}
```

High coupling

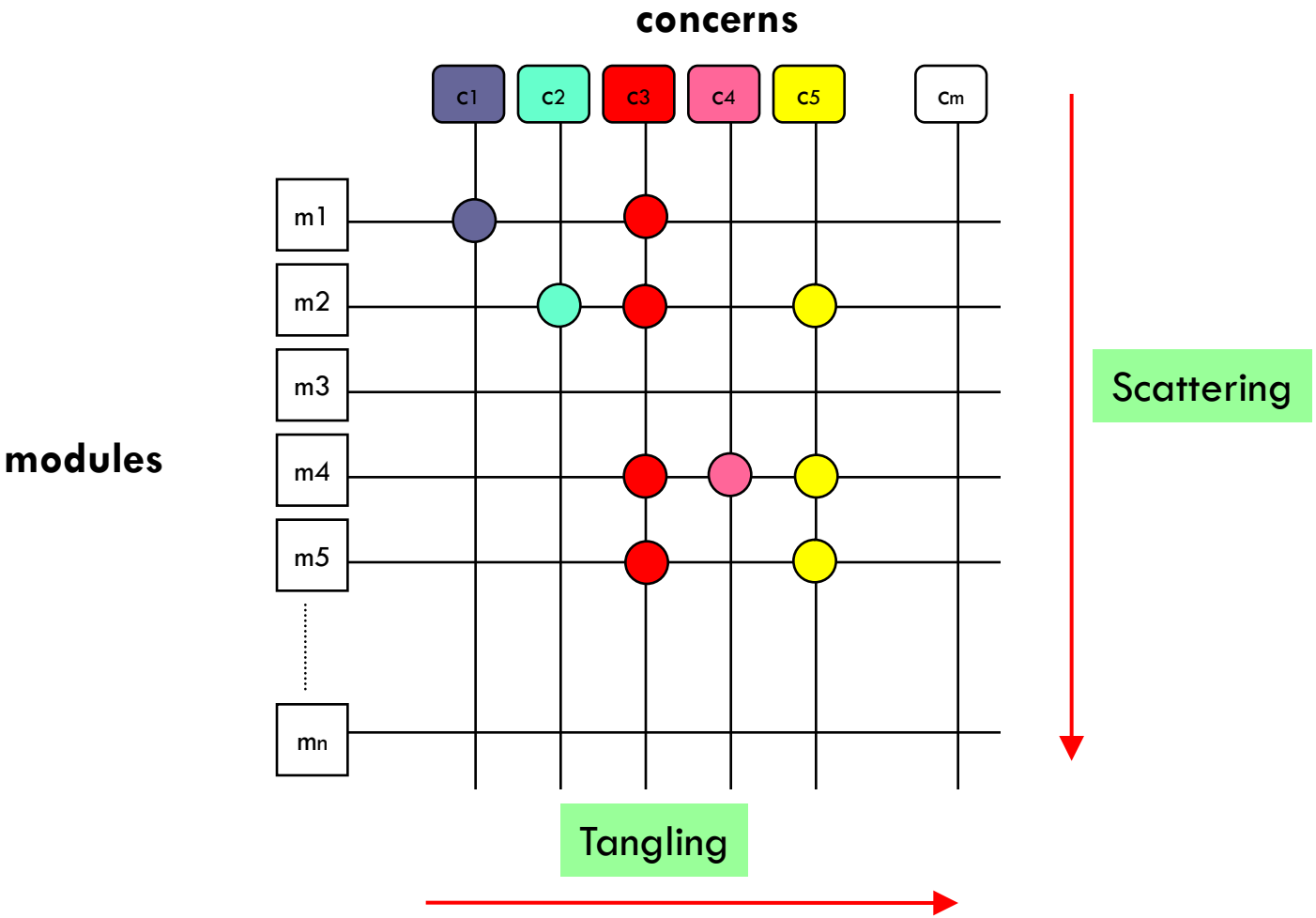
Low Cohesion

Mapping Concerns to Modules…



Separation of Concerns
Managing complexity
Increased Maintainability

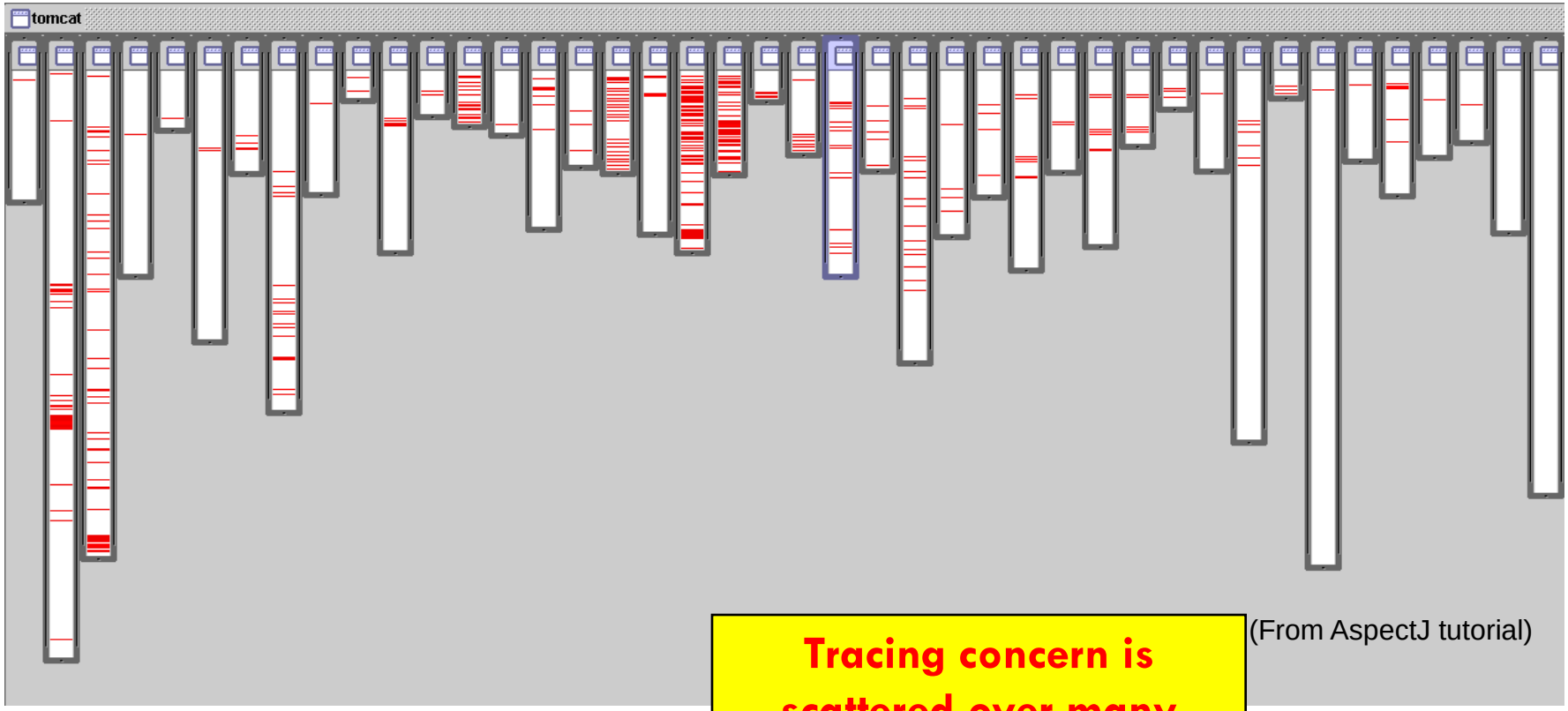
Crosscutting Concerns



Crosscutting, Scattering and Tangling

- Crosscutting
  - concern that *inherently* relates to multiple components.
  - results in scattered concern and tangled code
- Scattering
  - Single concern affects multiple modules
- Tangling
  - multiple concerns are interleaved in a single module

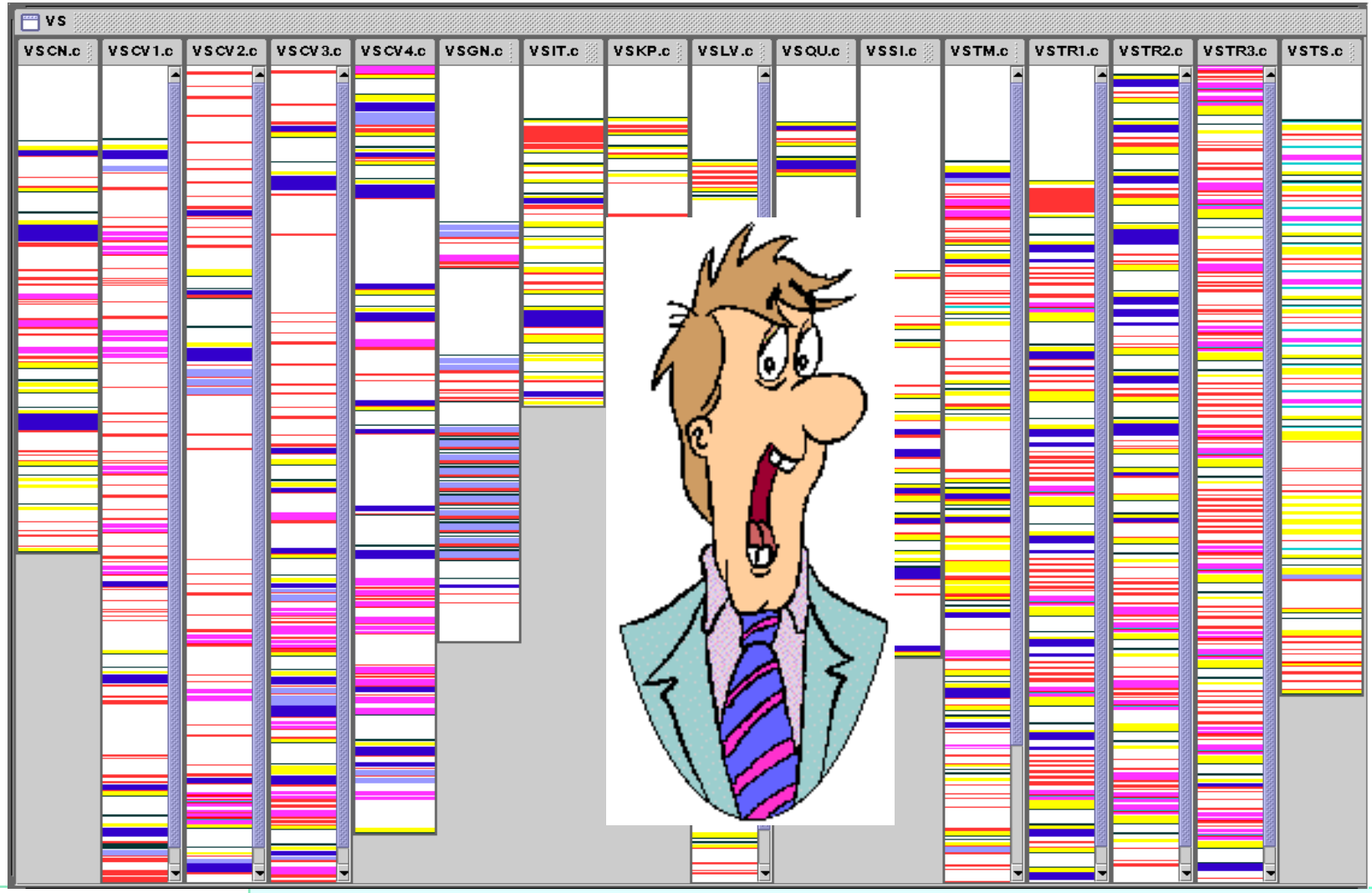
Aspects Visualized - Tracing



(From AspectJ tutorial)

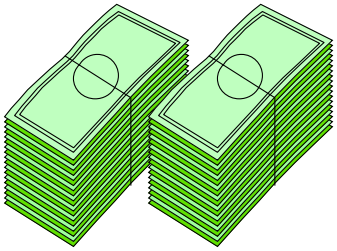
Tracing concern is scattered over many modules.
In every module tangled code.

Many more aspects… ☹



The Cost of Crosscutting Concerns

- Reduced understandability
  - Redundant code in many places
  - non-explicit structure
- Decreased adaptability
  - have to find all the code involved
  - and be sure to change it consistently
  - and be sure not to break it by accident
  - New concerns cannot be easily added
- Decreased reusability
  - component code is tangled with specific tangling code
- Decreased maintainability
  - 'ripple effect'



Example of crosscutting concerns

- Synchronization
- Real-time constraints
- error-checking
- object interaction constraints
- memory management
- persistency
- security
- caching
- logging
- monitoring
- testing
- domain specific optimization
- ...

What to do...?

- improve requirements analysis
- more and better design
- apply coding guidelines
- improve project management
- process improvement (CMMI)
- improve testing
- more documentation
- .....

Aspect-Oriented Software Development

- provides better separation of concerns by explicitly considering crosscutting concerns (as well)
- Does this by providing explicit abstractions for representing crosscutting concerns, i.e. aspects
- and composing these into programs, i.e. aspect weaving or aspect composing.
- As such AOSD improves modularity
- and supports quality factors such as
  - maintainability
  - adaptability
  - reusability
  - understandability
  - ...

Example - AspectJ

```
class Line {
  private Point _p1, _p2;

  Point getP1() { return _p1; }
  Point getP2() { return _p2; }

  void setP1(Point p1) {
    _p1 = p1;
  }
  void setP2(Point p2) {
    _p2 = p2;
  }
}

class Point {
  private int _x = 0, _y = 0;

  int getX() { return _x; }
  int getY() { return _y; }

  void setX(int x) {
    _x = x;
  }
  void setY(int y) {
    _y = y;
  }
}
```

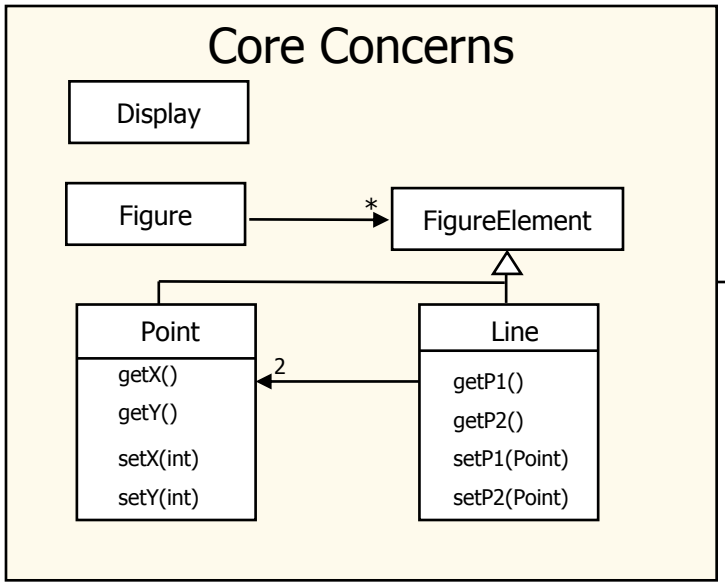
```
aspect Tracing {

  pointcut traced():
    call(* Line.*) ||
    call(* Point.*);

  before(): traced() {
    println("Entering:" +
            thisjopinpoint);

  after(): traced() {
    println("Exit:" +
            thisjopinpoint);


  void println(String str)
  {<write to appropriate stream>}

  }
}
```

aspect

pointcut

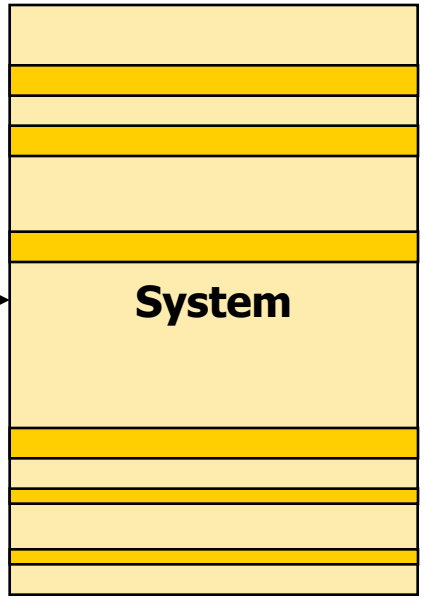
advice

AOP Solution

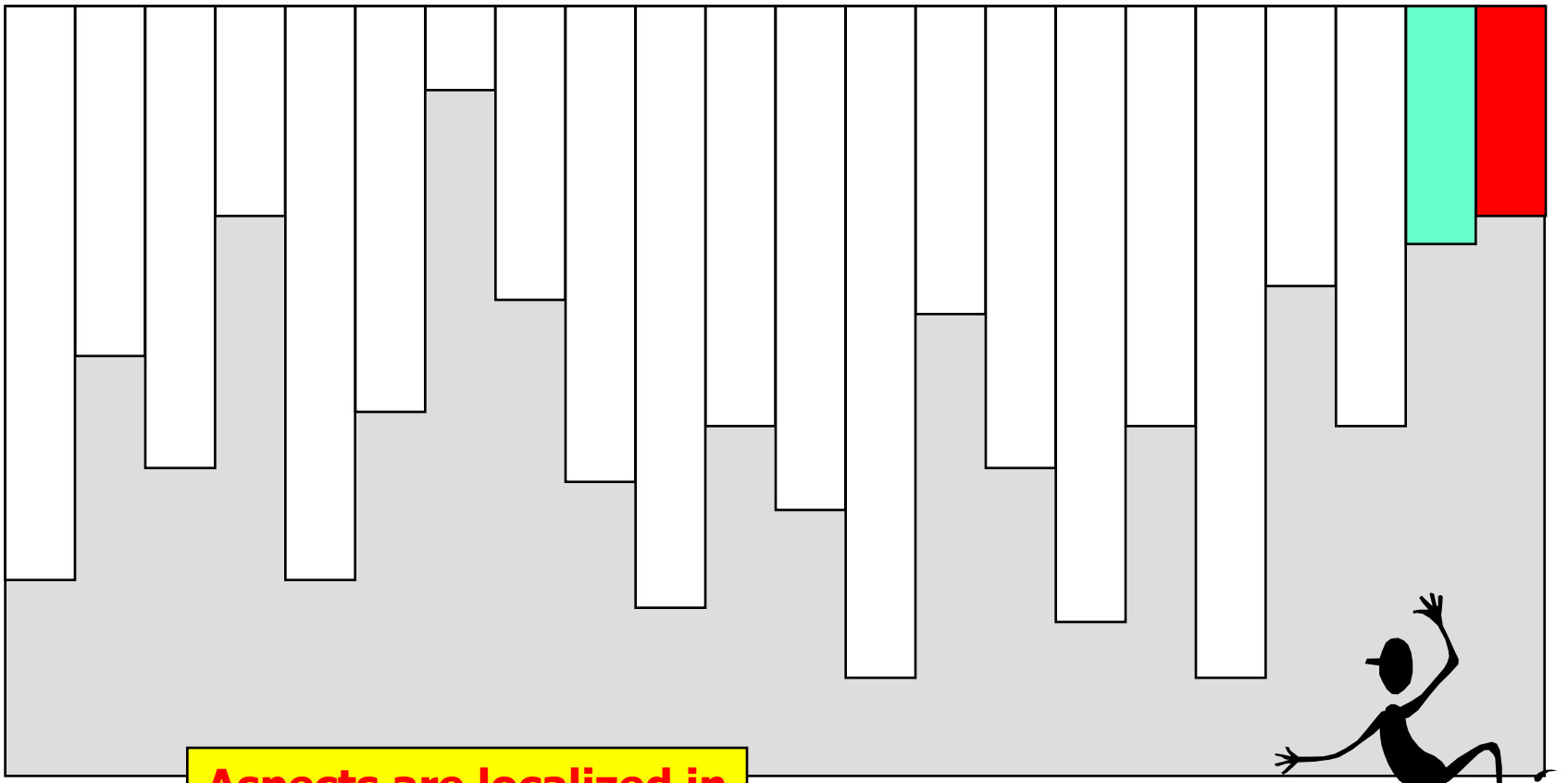



```
aspect Tracing {

  pointcut traced():
    call(* Line.*) ||
    call(* Point.*);

  before(): traced() {
    println("Entering:" + thisjopinpoint);

   after(): traced() {
    println("Exit:" + thisjopinpoint);

  void println(String str)
  {<write to appropriate stream>}

  }
}
```

Aspect-Oriented Weaver



Aspects Visualized



Aspects are localized in separate modules. No crosscutting.

Tracing Aspect

Display Tracking Aspect

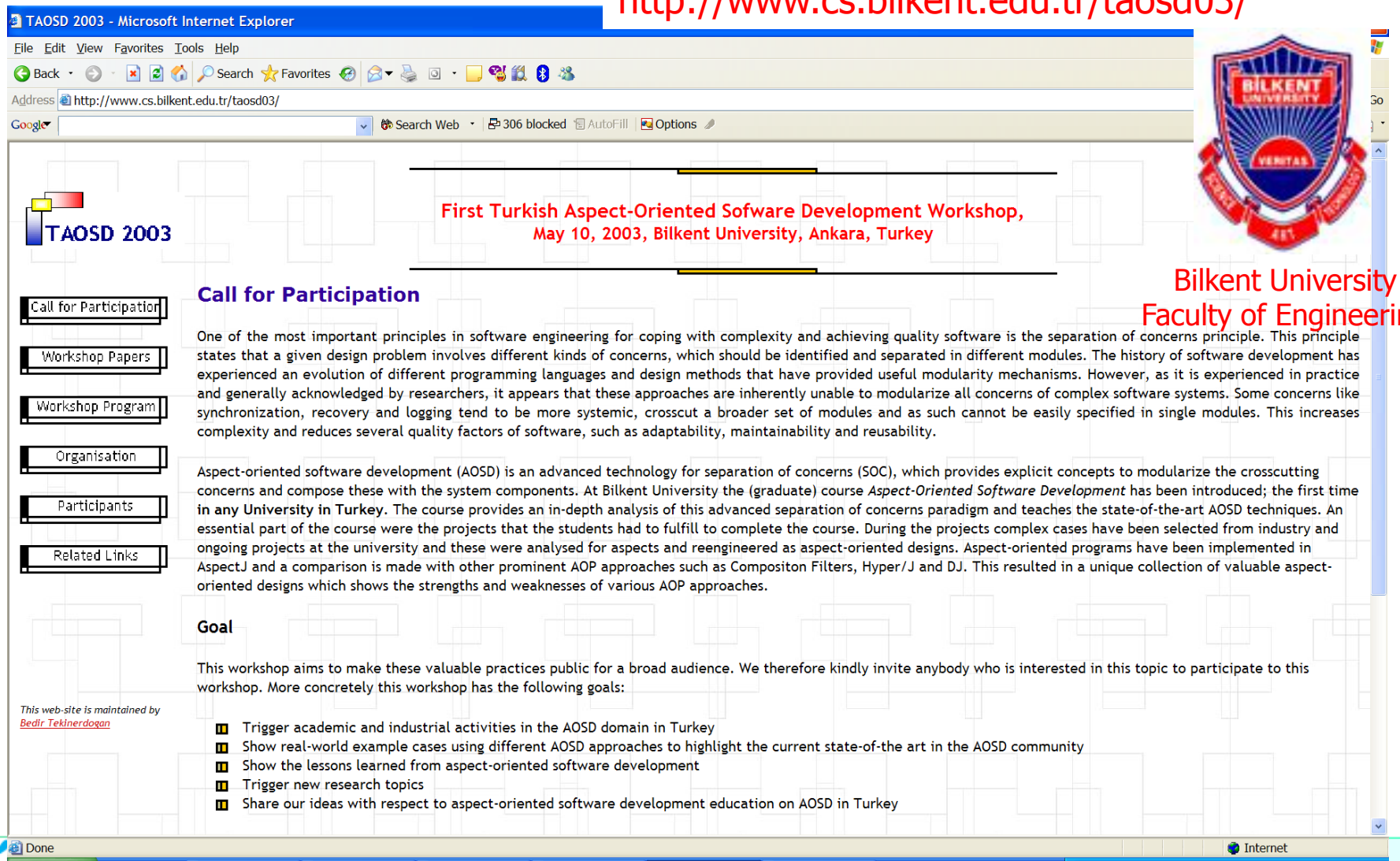
Birazda Türkçe... ☺

English	Türkçe
concern	ilgi
cohesion	uyuşma
coupling	bağlaşım
scattering	saçılma
crosscutting	enine kesen
aspect	aspekt ☺

Workshop Organization and Goals

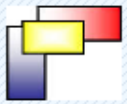
First TAOSD Workshop - 2003

http://www.cs.bilkent.edu.tr/taosd03/



Bilkent University
Faculty of Engineering

Second TAOSD Workshop - 2007



TAOSD 2007

II. Turkish Aspect-Oriented Software Development Workshop

Saturday, 29-09-2007

Bilkent University Turkey

- Goals
- Call For Papers
- Important Dates
- Submissions
- Organisation
- Papers
- Program
- Registration
- Links
- Contact

Türkçe

Goals

TAOSD is entirely devoted to Aspect-Oriented Software Development. Its purpose is to bring together software engineering practitioners and researchers from industry and academia in Turkey to exchange experiences, results and ideas related to AOSD concepts. The particular goals of this workshop are the following:

Improve consciousness on AOSD

We can observe a widespread use of AOSD in the world. It is being teached in universities, companies apply AOSD techniques, conferences and workshops are organized regulary, and large projects on AOSD are funded. The primary goal of this workshop is to foster the consciousness about this key concept in software engineering and provide an objective evaluation.

Stimulate research and education on AOSD

We hope to stimulate the research and education in Turkey with respect to aspect-oriented software development. Researchers will find a forum and a channel to present and share their ideas. Educators will find the important topics in aspect-oriented software development and include these in their courses.

Reflect and foster state-of-the practice of AOSD

With this workshop we hope to reflect the state-of-the-practice with respect to AOSD in Turkey. The workshop will provide an opportunity to represent the latest developments in industrial software projects and highlight the identified problems and the solutions. Software companies will thus have an opportunity to evaluate the benefits and risks for AOSD for their business.

Support MSc and PhD students in providing directions guidelines for their research

The workshop will provide an opportunity for PhD students who are doing research on AOSD to communicate and discuss their ideas in the context of the workshop.

Third TAOSD Workshop - 2008

http://www.cs.bilkent.edu.tr/Bilsen/TAOSD-2008/

TAOSD Ankara 2008

- Call for Participation
- Organization
- Participants
- Workshop Papers
- Workshop Program
- Registration
- Links
- Contact

Bilkent University

BilSen
Bilkent Software Engineering Research and Education

Third Turkish Aspect-Oriented Software Development Workshop

Friday 26 December 2008, Bilkent University, Ankara, Turkey
Güzel Sanatlar Fakültesi, FFB05

Call for Participation

One of the most important principles in software engineering for coping with complexity and achieving quality software is the separation of concerns principle. This principle states that a given design problem involves different kinds of concerns, which should be identified and separated in different modules. The history of software development has experienced an evolution of different programming languages and design methods that have provided useful modularity mechanisms. However, as it is experienced in practice and generally acknowledged by researchers, it appears that these approaches are inherently unable to modularize all concerns of complex software systems. Some concerns like synchronization, recovery and logging tend to be more systemic, crosscut a broader set of modules and as such cannot be easily specified in single modules. This increases complexity and reduces several quality factors of software, such as adaptability, maintainability and reusability.

Aspect-oriented software development (AOSD) is an advanced technology for separation of concerns (SOC), which provides explicit concepts to modularize the crosscutting concerns and compose these with the system components.

In Turkey, the First Aspect-Oriented Software Development Workshop has been organized in June 2003. The Second Aspect-Oriented Software Development Workshop has been organized in September 2007. With this workshop we aim to further stimulate the research and education on AOSD in Turkey. The workshop is organized within the AOSD (graduate) course that is given at Bilkent University. The course provides an in-depth analysis of AOSD and teaches the state-of-the-art AOSD techniques. An essential part of the course are the projects that the students had to fulfill to complete the course. During the projects complex cases have been selected from industry and ongoing projects at the university and these were analysed for aspects and reengineered as aspect-oriented designs. Aspect-oriented programs have been implemented in AspectJ and a comparison is made with other AOP approaches. This resulted in a unique collection of valuable aspect-oriented designs which shows the strengths and weaknesses of various AOP approaches. We think that the results of these projects are of interest to a broader public and as such would like to share our experiences in this workshop.

In particular, the goals of this workshop are the following:

- ***Share knowledge and improve consciousness on AOSD***

  We can observe a widespread use of AOSD in the world. It is being teached in universities, companies apply AOSD techniques, conferences and workshops are organized regulary, and large projects on AOSD are funded. The primary goal of this

Goal

- Improve consciousness on AOSD
- Stimulate research and education on AOSD
- Reflect and foster state-of-the practice of AOSD
- Support MSc and PhD students in providing directions guidelines for their research

AOSD Course at Bilkent University

http://www.cs.bilkent.edu.tr/~bedir/CS586-AOSD/

CS 586 - Aspect-Oriented Software Development

- Description
- Study Material
- Schedule
- Grading
- Project
- Course Slides
- Related Links

Instructor:
Dr. Bedir Tekinerdoğan
bedir@cs.bilkent.edu.tr

Bilkent University

CS 586 Aspect-Oriented Software Development NEW

One of the most important principles in software engineering for coping with complexity and achieving quality software is the separation of concerns principle. This principle states that a given design problem involves different kinds of concerns, which should be identified and separated in different modules. The history of software development has experienced an evolution of different programming languages and design methods that have provided useful modularity mechanisms. However, as it is experienced in practice and generally acknowledged by researchers, it appears that these approaches are inherently unable to modularize all concerns of complex software systems. Some concerns like synchronization, recovery and logging tend to be more systemic, crosscut a broader set of modules and as such cannot be easily specified in single modules. This increases complexity and reduces several quality factors of software, such as adaptability, maintainability and reusability.

Aspect-oriented software development (AOSD) is an advanced technology for separation of concerns (SOC), which provides explicit concepts to modularize the crosscutting concerns and compose these with the system components. This course will provide an in-depth analysis of this advanced separation of concerns paradigm and teach the state-of-the-art AOSD techniques.

The important topics in this course are the following:

- separation of concerns;
- object-oriented design patterns
- software evolution problems;
- examples of crosscutting aspects;
- composition anomalies.
- aspect-oriented programming
- aspect-oriented design;
- aspect-oriented modeling;
- aspects at the requirements and architecture design level;
- reflection and delegation techniques;

Prerequisites

It is -highly- recommended that you have followed a course on object-oriented software development, know the fundamental object-oriented concepts, have sufficient knowledge on UML, and have sufficient programming skills in Java.

AOSD Project

Tasks

- Select a case (existing or developed from scratch)
- Object-oriented design of the case; using object-oriented design patterns
- Develop change scenarios
- Problem Statement: explain the shortcomings of the object-oriented design and identify crosscutting concerns, aspects
- Aspect-Oriented Programming in AspectJ
- and comparison with another AOP language

Deliverables

- Aspect-oriented program(s) in Java.
- PowerPoint presentation
- Workshop paper

Bilsen - Software Engineering Group

http://www.cs.bilkent.edu.tr/Bilsen

BilSen
Bilkent Software Engineering Research and Education

Home
Mission and Vision
Research
Education
Members
Publications
Projects
Events
News
Links
Bilsen Talks
Tools
Contact

Bilkent University

Mission Statement

We are dedicated to continued growth in the field of software engineering and aim to provide excellence in research, teaching and service relative to the university in particular and the community in general.

We are committed to responsible and innovative way of working and collaboration with academic and industrial partners, both in Turkey and abroad.

We aim to achieve our goals through publishing high quality papers, actively participating in the software engineering community, writing industrially relevant project proposals, and constant collaboration with our colleagues and students who understand the importance and value of software engineering.

Vision

We believe that software engineering is one of the most important disciplines in computer science with the largest impact on the community.

We envision that this influence on the society will remain for the future and will even get stronger for the coming decades.

In particular, for Turkey with a highly developing potential, we think that software engineering should be put on the agenda for a broader interest and participation.

Based on our in-depth experiences in different domains of software engineering we think that we can substantially contribute to the developments in state-of-the-art and state-of-the-practice of software engineering.

Organisation

Unique in two perspectives

- Presenters
  - Mostly students
- Audience
  - Students
  - Visitors with probably no explicit/mature knowledge on AOSD

Synergy – Win x Win x Win x ….

- *Synergy*: Final outcome is greater than the sum of its parts…

Workshop goal = Win^n

- AOSD course students
  - Experience in workshop setting
  - Experience in writing papers
  - Experience in writing AOP program
- Workshop Participants
  - Knowledge of AOSD domain
  - Acquiring new ideas for development and research
- Bilkent University…
- Bilsen ☺
- Software Engineering *Research and Education* in Turkey
- …?

New AOSD Experts in Turkey... ☺

#		Last name	Name
1		Aksoy	Cem
2		Aksu	Hidayet
3		Aksu	Muharrem Uğur
4		Algan	Eren
5		Altunbay	Doğan
6		Arifoğlu	Damla
7		Bağlıoğlu	Özgür
8		Belet	Faruk
9		Belviranlı	Mehmet Esat
10		Berktaş	Doğan Kaya
11		Cebe	Mümin
12		Ceylan	Duygu
13		Çığır	Celal
14		Dilek	Alptuğ
15		Gündoğdu	Ramazan Bertan
16		Gürcüoğlu	Gizem
17		Kurtcephe	Murat
18		Kutlu	Mücahid
19		Oğuz	Oğuzcan
20		Onarlıoğlu	Kaan
21		Özçelik	Mehmet
22		Özdemir	Bahadır
23		Sevil	Sare Gül
24		Tekdoğan	Rıdvan
25		Türel	Bahri
26		Yazıcı	Volkan
27		Zitouni	Hilal

Workshop Program - Morning

Time	Topic	Presenter(s)
8:45-9:15	Introduction to Aspect-Oriented Software Development and Goal of Workshop	Bedir Tekinerdoğan
	Presentation Session: Visualization and Gaming	
9:15-9:45	Analyzing Aspects in Gaming Applications	Cem Aksoy, Bahri Türel, Mücahid Kutlu
9:45-10:15	Aspect-Oriented Refactoring of a Graph Visualization Tool: CHISIO	Mehmet Esat Belviranli Celal Cigir, Alptug Dilek
10:15-10:30	*Break*	
	Presentation Session: P2P and Distributed Systems	
10:30-11:00	Aspect-Oriented Multi-Client Chat Applications	Duygu Ceylan, Gizem Gürcüoğlu, Sare Sevil
11:00-11:30	Aspect-Oriented Development of a P2P Secure File Synchronization Application	R. Bertan Gündoğdu, Kaan Onarlıoğlu, Volkan Yazıcıoğlu
11:30-12:00	Aspect-Oriented Development of P2P File Sharing System	Eren Algan, Doğan Kaya Berktaş, Ridvan Tekdoğan
12:00-13:00	*Lunch*	

Workshop Program - Afternoon

13:00-14:00	**Invited Talk:** **"Business Rule Seggregation with Aspects"**	**Semih Çetin, Cybersoft**
	Presentation Session: **Domain Applications**	
14:00-14:30	Aspect-Oriented Development of an E-Commerce Application	Doğan Altunbay, Damla Arifoğlu, Hilal Zitouni
14:30-15:00	Analysis and Implementation of Database Concerns using Aspects	Murat Kurtcephe, Oğuzcan Oğuz, Mehmet Özçelik
15:00-15:15	*Break*	
	Presentation Session: **Simulations**	
15:15-15:45	Aspect-Oriented development of Visual MIPS Datapath Simulator	Hidayet Aksu, Özgür Bağlıoğlu Mümin Cebe
15:45-16:15	Aspect-Oriented Development of a Discrete Event Simulation System	M. Uğur Aksu Faruk Belet, Bahadır Bozdemir,
16:15-17:00	**Question Session** **Summary and plans for the future**	Bedir Tekinerdoğan (moderator)

Participants

- Around 50-60 participants
- Both from industry and academy
  - Ankara Üniversitesi
  - Aselsan
  - Bilkent Üniversitesi
  - Cybersoft
  - Hacettepe Universitesi
  - Havelsan
  - ODTÜ
  - Sosyal Güvenlik Kurumu
  - Tübitak
  - University of Twente, Hollanda
  - Uppsala University, Sweden
  - ….

Third Turkish Aspect-Oriented Software Development Workshop

Bilkent University, Ankara, Turkey
December 26, 2008