

# Reflection on Turkish Aspect-Oriented Software Development Workshop Series

Bedir Tekinerdogan  
Bilkent University, Turkey  
[bedir@cs.bilkent.edu.tr](mailto:bedir@cs.bilkent.edu.tr)  
DOI: 10.1145/2108144.2108165  
<http://doi.acm.org/10.1145/2108144.2108165>

## Abstract

In the last decade aspect-oriented software development (AOSD) has gained a broad interest in both academic institutions and industry. Likewise, several international conferences and workshops have been organized around the topic of AOSD. This paper summarizes the results of the national Turkish Aspect-Oriented Software Development Workshop series that have been organized in the last decade with the goal to stimulate the research and education on AOSD in Turkey. The fifth workshop was organized in December 2011. Aspects identified and demonstrated during the workshops have been collected in the so-called Aspect Browser. We report both on the experiences from the workshop series and the resulting aspect browser.

**Keywords:** Aspect-Oriented Software Development, Workshop organization

## 1. Introduction

One of the most important principles in software engineering for coping with complexity and achieving quality software is the separation of concerns principle [3][13]. 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 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 paradigm for separation of concerns (SOC), which provides explicit abstractions to modularize the crosscutting concerns and compose these with the system components [6][3].

AOSD has basically emerged from research in object-oriented programming. The problems encountered and the related solutions of aspect-oriented software development were originally discussed in various workshops in conferences such as ECOOP [5], OOPSLA [11] and ICSE [7]. The topics AOSD is concerned with are traditionally taught as part of object-oriented software development and software engineering courses.

After an increasing maturation and a consensus on the core

AOSD concepts we can now speak of a separate AOSD community [1]. A separate conference called Aspect-Oriented Software Development is organized yearly, international projects such as the network of excellence on AOSD has been set up, and special issues of journals have started to publish papers on AOSD. The logical consequences of the separation and emergence of AOSD as an independent activity affect software engineering education as well. The need for separate courses on aspect-oriented software development is growing and an increasing number of courses on aspect-oriented software development are actually starting to be taught or are planned to be introduced into the curriculum.

In Turkey, we have started the workshop series on AOSD with the First Aspect-Oriented Software Development Workshop which has been organized for the first time in June 2003. Later on it was organized in September 2007, December 2008, December 2009 and December 2011. The goal of these workshops was to stimulate the research and education on AOSD in Turkey. In these workshops different papers were presented that included the discussion of various kinds of aspects. We have collected these aspects in the so-called Aspect Browser that includes all the aspects that have been discussed during the workshops. We report both on the experiences from the workshop and the resulting aspect browser.

The remainder of the paper is organized as follows. In Section 2, we describe the workshop organization. Section 3 describes the concrete workshop goals. Section 4 discusses the workshop topics. Section 5 presents the aspect browser and finally section 6 concludes the paper.

## 2. Workshop Organization

The workshop started to be organized in 2003 at Bilkent University, Ankara, Turkey. In fact, the workshop organization was initiated as a result of the graduate course CS586 Aspect-Oriented Software Development that has been introduced for the first time in 2003 in Turkey [16]. The course provides an in-depth analysis of this advanced separation of concerns paradigm and teaches the state-of-the-art AOSD techniques. An essential part of the course was the project assignment 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 analyzed for aspects and reengineered as aspect-oriented designs. Aspect-oriented programs have been implemented in AspectJ [8] and a comparison is made with other prominent AOP approaches such as Composition Filters [2], Hyper/J [15], DJ [10], JBossAOP and SpringAOP. This resulted in a unique collection of valuable aspect-oriented designs which

shows the strengths and weaknesses of AOSD. Because of this unique experience and the interesting results that came out of these projects we decided to start organizing this workshop to make these valuable practices public for a broad audience.

The workshop appeared to be a big success regarding the interest that it received from both the academic institutions and the Turkish software industry. Table 1 shows the number of participants to the five workshops that were organized since 2003.

**Table 1.** The organized TAOSD workshops and the number of participants

Workshop	# of participants
First TAOSD (2003)	78
Second TAOSD (2007)	50
Third TAOSD (2008)	56
Fourth TAOSD (2009)	35
Fifth TAOSD (2011)	25

### 3. Workshop Goals

TAOSD workshops have been entirely devoted to Aspect-Oriented Software Development. The purpose of the workshops was 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 were the following:

- *Improve consciousness on AOSD*

We can observe a widespread use of AOSD in the world. It is being taught in universities, companies apply AOSD techniques, conferences and workshops are organized regularly, and large projects on AOSD are funded. The primary goal of the workshops was to foster the consciousness about this key concept in software engineering in Turkey.

- *Stimulate research and education on AOSD*

We hoped to stimulate the research and education in Turkey with respect to aspect-oriented software development. Researchers would find a forum and a channel to present and share their ideas. Educators would 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 series we hoped also to reflect the state-of-the-practice with respect to AOSD in Turkey. The workshop aimed to provide an opportunity to represent the latest developments in industrial software projects and highlight the identified problems and the solutions. Software companies would 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 would provide an opportunity for PhD students who are doing research on AOSD to communicate and discuss their ideas in the context of the workshop.

Of course it is very hard to measure to what extent these goals have been achieved. Nevertheless, given the large number of

participants and the active involvement of the participants during the workshop we believe that these workshops have been very useful to support these goals. Besides of the interest in the topic we could also observe a specialization and broader experience and knowledge among the workshop participants in the later workshops.

### 4. Workshop Topics

In fact we deliberately did not constrain the topics for the workshop. The workshop papers were primarily derived from the reports as defined in the AOSD project in the Bilkent AOSD course. The project's aim was to clarify the concepts that had been taught during the classes using teamwork. For this, the students formed groups of three or four team members, working together on a case that they had selected themselves. Most of these cases were existing cases or research topics of the group in which the students had to finalize their MSc assignments. As such the projects had also an indirect positive effect since the students directly applied the concepts in the real research setting. Also, in principle workshop papers could also be sent from external authors. However, since the AOSD topic was not broadly known most of the papers were derived from the course project reports.

Another important aim of the AOSD project was to learn how to identify crosscutting concerns (problems) and specify, implement and evaluate aspects (solutions). For this, the students had to formulate the problem of crosscutting concerns in the given case, plan the solution, design and translate the solution, and finally implement the solution. To exercise the AOP problems, the selected cases had to include at least four aspects, including both *development aspects*, *enforcement aspects*, or *production aspects*. *Development aspects* represent the aspects that are used to facilitate the development of aspects, such as tracing, debugging and testing aspects. *Enforcement aspects* are aspects that aim to ensure contractual specifications. *Production aspects* represent aspects that are inherently intended to be included in production. Considering multiple aspects at a time is important to highlight the dependencies between aspects, and the additional complexities they impose on the eventual design. In general, the development aspects of the groups did not differ much because there seems to be a fixed set of these aspects. The selected production aspects on the other hand were all different because every group had a different case.

Using the selected cases the following subtasks had to be performed:

- Object-oriented design of the case: to include at least two object-oriented design patterns;
- Develop change scenarios which cannot be easily integrated into the case;
- Problem Statement: explain the shortcomings of the object-oriented model with respect to the crosscutting concerns and show this using the defined change scenarios.
- Aspect-Oriented Design of the case using a selected notation.
- Aspect-Oriented Programming using a preferred AOP language (usually AspectJ)
- Provide alternative aspect specifications using alternative

AOP approaches such as Composition Filters, HyperJ, DJ or the more recent framework based AOP approaches such as SpringAOP or JBossAop.

- Evaluate the aspect-oriented design and compare the chosen AOP approaches.

Table 2 shows the list of the TAOSD workshops with the key presentation sessions/topics. Each presentation session had usually around 2-4 presentations. Each presentation took around 30 minutes including presentation of the ideas, a real demo and discussion.

**Table 2.** The organized TAOSD workshops and the presentation topics

Workshop	Presentation Sessions/Topics
First TAOSD (2003)	<ul style="list-style-type: none"> <li>• Domain Specific Aspects,</li> <li>• Tools</li> <li>• Distributed Systems</li> <li>• Aspects in Simulation Systems</li> </ul>
Second TAOSD (2007)	<ul style="list-style-type: none"> <li>• Aspect Weaving,</li> <li>• AOP in Embedded Systems</li> <li>• Empirical Results of AOP</li> </ul>
Third TAOSD (2008)	<ul style="list-style-type: none"> <li>• Aspects in Database Systems</li> <li>• Software Refactoring using AOP</li> <li>• Aspects in Simulation Systems</li> <li>• Aspects in Distributed Systems</li> </ul>
Fourth TAOSD (2009)	<ul style="list-style-type: none"> <li>• Refactoring using AOP</li> <li>• Domain Specific Aspects</li> <li>• Aspects in Simulation Systems</li> <li>• Enhancing application frameworks using aspects</li> </ul>
Fifth TAOSD (2011)	<ul style="list-style-type: none"> <li>• Refactoring using AOP</li> <li>• Gaming Aspects</li> <li>• Mining Aspects</li> <li>• AOP-based Middleware</li> </ul>

## 5. Aspect Browser

The TAOSD workshops have been successfully organized since 2003. Many different aspects have been discussed in the papers that were submitted to the TAOSD workshops. We have analyzed these and collected these in the so-called *Aspect Browser* that is available online [14]. We think that the aspects can be useful for a broader audience and as such would like to share our experiences.

The following table includes an example set of the aspects that can be identified in the papers in the TAOSD workshops. Aspects are categorized as Production Aspect, Development Aspect or Enforcement Aspect [6].

Each aspect belongs to some application domain such as Gaming, Graph Visualization or Distributed Systems. The last column indicates the source and the page numbers where the aspect is explained. The Aspect Browser is an active document and is continuously updated.

**Table 3.** Aspect Browser that includes the aspects (Partial) presented at TAOSD workshops

Aspect	Domain	Type	Description of the Aspect	Source TAOSD
Access Control	General	Dev.	Access control enables an authority to control access to areas and resources in a given physical facility or computer-based information system.	2003 2009
Authentication	General	Prod.	In many applications clients need to authenticate themselves before accessing the resources. This aspect modularizes the authentication concern.	2003 2008 2009
Authorization	General	Prod.	Authorization, is a process that establishes whether an authenticated user has sufficient permissions to access certain resources.	2003 2008 2009
Checking pre and post conditions	General, Constraint-based systems	Enf.	In many applications the execution of operations is bound to the pre and post conditions check.	2008
Checking in/Checking out	Configuration Management, Versioning Management	Prod.	In configuration management and revision control systems one developer at a time has write access to the central repository copies of those files. Once one developer "checks out" a file, others can read that file, but no one else is allowed to change that file until that developer "checks in" the updated version (or cancels the checkout)	2003 2011
Collision Detection	Gaming	Prod	In gaming applications multiple objects can move. This aspect detects the collision of objects in the game.	2008
Connection Setting	Resource Management, Database Systems	Prod.	A Database connection is a facility that allows client software to communicate with database server software, whether on the same machine or not. A connection is required to send commands and receive answers.	2003 2009
Constraint Checking	General, Constraint-based systems	Prod.	When changing the behavior or state of an object usually some constraints need to be checked. This aspect captures the locations of these situations and defines the constraint	2003

			handling	
Creation of Objects	General	Production	Creation of objects can be scattered over an application.	2008
Communication Protocol	General	Production	a communications protocol is the set of standard rules for data representation, signaling, authentication and error detection required to send information over a communications channel.	2009

## 6. Conclusion

In this paper, we have reported on the Turkish Aspect-Oriented Software Development (TAOSD) workshop series that were organized in the last decade. The primary goal of the workshop series was to foster the research and education in the field of aspect-oriented software development in the local setting of Turkey. In addition to the workshop organization we have also discussed the Aspect Browser that includes the aspects discussed during the TAOSD workshops. These aspects can support practitioners and researchers to identify and implement aspects. In fact all these aspects are discussed in papers which also provide the specification of the aspects. From a workshop organization perspective we believe that our experiences can help other peer researchers who wish to organize also workshop in a national setting. The eventual goal of the national workshop is to support the involvement of practitioners and researchers in the international AOSD conferences and workshops.

## Acknowledgments

The organizers of the workshop would like to thank the participants and the workshops' program committee.

## References

- [1] AOSD community home page, <http://aosd.net>
- [2] M. Aksit, L. Bergmans and B. Tekinerdogan. "Aspect-Composition using Composition Filters", in: *Software Architectures and Component Technology: The State of the Art in Research and Practice*, M. Aksit (Ed.), Kluwer Academic Publishers, pp. 357 - 382, 2001.
- [3] R. Chitchyan, A. Rashid, P. Sawyer, A. Garcia, J. Bakker, M. Pinto Alarcon, B. Tekinerdogan, S. Clarke, and A. Jackson, *Survey of Aspect-Oriented Analysis and Design*, AOSD Europe network of excellence, no. AOSD-Europe-ULANC-9, pp. 1-259, 2005.
- [4] E.W.Dijkstra. "The structure of "THE"-multiprogramming system". *Communications of the ACM* 11, 5, 341-346, 1968.
- [5] European Conference on Object-Oriented Programming (ECOOP), <http://www.ecoop.org/>.
- [6] T. Elrad, R. Fillman and A. Bader. "Aspect-Oriented Programming". *Communication of the ACM*, Vol. 44, No. 10, October 2001.
- [7] International Conference on Software Engineering, <http://www.ifi.uzh.ch/icse2012/>, 2012.
- [8] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm,

and W.G. Griswold. "An Overview of AspectJ". In J. Lindskov Knudsen (ed.), *ECOOP 2001 Object-Oriented Programming 15th European Conference*, Budapest Hungary, pages 327-353. Volume 2072 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, June, 1997.

- [9] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier and J. Irwin. "Aspect-Oriented Programming". In *proceedings of ECOOP '97*, Springer-Verlag LNCS 1241. June 1997.
- [10] K.Lieberherr, D. Orleans and J. Ovlinger. "Aspect-Oriented Programming with Adaptive Methods", *Communications of the ACM*, Vol. 44, No. 10, pp. 39-43, October 2001..
- [11] *Object-Oriented Programming, Systems, Languages and Applications conference*, <http://www.oopsla.org/>.
- [12] H. Ossher and P. Tarr. "Multi-Dimensional Separation of Concerns using Hyperspaces". IBM Research Report 21452, April, 1999.
- [13] D. Parnas. "On the Criteria To Be Used in Decomposing Systems into Modules", *Communications of the ACM*, Vol. 15, No. 12, pp. 1053 - 1058, December 1972.
- [14] TAOSD workshops, <http://www.cs.bilkent.edu.tr/Bilsen/TAOSD-2011/>
- [15] P. Tarr, H. Ossher, W. Harrison and S.M. Sutton, Jr. "N Degrees of Separation: Multi-Dimensional Separation of Concerns". In *proceedings of ICSE 21*, May, 1999.
- [16] B. Tekinerdogan, CS586-Aspect-Oriented Software Development Course Home Page, Bilkent University, Ankara, Turkey <http://www.cs.bilkent.edu.tr/~bedir/CS586-AOSD/>, 2003.