

# Aspect-Oriented Modeling: Bridging the Gap Between Implementation and Design

Tzilla Elrad  
Illinois Institute of Technology  
[Elrad@iit.edu](mailto:Elrad@iit.edu)

Omar Aldawud  
Lucent Technologies  
[oaldawud@lucent.com](mailto:oaldawud@lucent.com)

Atef Bader  
Lucent Technologies  
[abader@lucent.com](mailto:abader@lucent.com)

## Abstract.

*Separation of Concerns is one of the software engineering design principles that is getting more attention from practitioners and researchers in order to promote design reuse and code reuse. Separation of Concerns tends to separate requirements such as synchronization and scheduling from the core functionality of the software system. These requirements are often referred to as crosscutting concerns. The implementation of such requirements is scattered throughout the software system, which results in the code-tangling problem. Code tangling makes the software system hard to reuse, maintain, and comprehend. Aspect Oriented Programming provides the user with the ability to modularize, and weave the representation of crosscutting concerns in order to maximize code reusability and solves the code-tangling*

*problem. Weaving is the process of combining crosscutting concerns with the core functionality of the software system. Using the Unified Modeling Language (UML) to model and inter-weave these crosscutting concerns clearly is a craft that is hard to master due to the lack of a formal modeling technique based on separation of concerns. In this paper we present a formal design methodology to model the system's concerns based on aspect-orientation. Through the use of advanced features of statecharts and class diagrams, we show how to model and interweave the different sub-models that represent different concerns. In Addition our initial study shows that our approach demonstrates the benefits for aspect-oriented modeling, like maintainability, reusability and trace-ability<sup>1</sup>.*