Chapter 4
Aspect-Oriented Recommender Systems

Punam Bedi  
University of Delhi, India

Sumit Kr Agarwal  
University of Delhi, India

ABSTRACT

Recommender systems are widely used intelligent applications which assist users in a decision-making process to choose one item amongst a potentially overwhelming set of alternative products or services. Recommender systems use the opinions of members of a community to help individuals in that community by identifying information most likely to be interesting to them or relevant to their needs. Recommender systems have various core design crosscutting issues such as: user preference learning, security, mobility, visualization, interaction etc that are required to be handled properly in order to implement an efficient, good quality and maintainable recommender system. Implementation of these crosscutting design issues of the recommender systems using conventional agent-oriented approach creates the problem of code scattering and code tangling. An Aspect-Oriented Recommender System is a multi agent system that handles core design issues of the recommender system in a better modular way by using the concepts of aspect oriented programming, which in turn improves the system reusability, maintainability, and removes the scattering and tangling problems from the recommender system.

1 INTRODUCTION

People are often overwhelmed with the number of options available to them. The sheer number of available options often makes a wise choice impossible without some intelligent computational assistance. Recommender systems or the concept of automatic recommendation generation is one of the possible solutions to this information overload problem. Recommender Systems attempt to reduce information overload and retain customers by selecting a subset of items from a universal set.
Aspect-Oriented Recommender Systems

based on user preferences. A recommender system recommends items to users by predicting items relevant to the user, based on user profile which contains various kinds of information including items, user information and interactions between users and items (Adomavicius & Tuzhilin, 2005). Some of the core design issues in Recommender systems are as follows:

- **User Preference Learning:** Recommender systems acquire information about user preferences in an explicit (e.g., letting users express their opinion about items) or implicit (e.g., observing some behavioral features) way which is required to generate a list of recommended items (Bedi & Agarwal, 2011b).

- **Security:** Recommender systems are highly vulnerable to profiles injection attacks. Therefore, security mechanisms are needed for protecting the recommender systems against these attacks (Bedi & Agarwal, 2011c).

- **Mobility:** Recommender systems use the demographic information of the user i.e., the knowledge of the user’s location at a particular time for generating the more relevant recommendations for him (Bedi & Agarwal, 2011a).

- **Visualization:** Recommender systems use effective visualization interfaces in order to provide more effective and persuasive recommendations (Vashisth et al., 2011).

- **Interaction:** Recommender systems use various interactive interfaces in order to gather information about user preferences with minimum user interaction with the system (Zhang et al., 2008).

These core design issues of the recommender systems are crosscutting concerns. Crosscutting concerns usually refer to non functional properties of system such as learning, security, transaction management, synchronization, mobility, interaction, visualization and error handling etc that are used across the scope of a piece of the system. Implementation of these crosscutting concerns using conventional agent-oriented approach creates the problem of code scattering and code tangling. Scattering in agent-oriented models is the manifestation of design elements that belong to one specific concern, over several modeling units referred to other multi-agent system concerns. Tangling in agent-oriented models is the mix of multiple concerns together in the same modeling elements. Aspect-Oriented Recommender Systems (AORS) is the solution of these crosscutting design issues of the recommender system. Aspect-oriented recommender systems are multi agent systems that handle the core design issues of the recommender system in a better modular way by using the concept of aspect oriented programming to remove the scattering and tangling problems from the recommender system.

This chapter describes the brief introduction of an Aspect-oriented programming (AOP) and its constructs along with the various aspects of recommender system. Beginning with the description of various constructs of AOP and crosscutting issues of recommender system, this chapter introduces the inherent problems and their consequences of not having explicit support for the modularization of crosscutting concerns in the recommender systems. This chapter also provides a solution of these inherent problems to support crosscutting issues in design of multi-agent recommender system along with its advantages.

The chapter is organized as follows. Section 2 reviews the related work in this area. Concepts used in the proposed approach are discussed in section 3. Section 4 describes the proposed approach to deal with crosscutting concerns in the recommender system. Experimental details and results are shown in section 5 and finally conclusion and future work are discussed in section 6.