Software Configuration Management tools are widely available for monitoring the changes made by developers in their artifacts in order to prevent the conflicts, and integration or merging problem at the latter stages of software development. Generally, the parallel development concept leads to conflicts which are of two types. 1. Direct conflicts and 2. Indirect Conflicts. Former includes the parallel code changes of same workspace and the latter specifies the parallel code changes of different workspace i.e. change in one workspace affects the other. These conflicts are handled by the following approaches: Pessimistic, in which a lock is used for preventing parallel changes in the code. Optimistic, in which multiple developers are given access to same code simultaneously. Hence leading to conflicts but which are identified and resolved using merging tools. The direct conflicts are avoided and resolved in both approaches by usage of locks and merging after changes are made respectively. But indirect conflicts are not addressed in any of these. This paper explains the idea of sharing the modified information to all the developers which helps in the detection of conflicts in class signatures. Detailed information of the changes made by each developer is broadcasted to all workspaces involved in the project. The process involves sharing the details of indirect conflicts across workspaces, summarizing and analyzing those changes and finally notifications are sent to the developers.

Keywords: - Configuration management; Conflict; Workspace; Awareness;