1. Introduction

In Computer Science and for Software Engineers, Software Architecture recovery is of supreme importance. This chapter discusses its organization in what is left in this thesis and it briefly presents the motivation for our work.

1.1 Motivation

The impact of large Software systems exists on our everyday life during the years of nineties and twenties. For approximating 15 to 20 years the software systems are very critical, large and complex. In most of the cases the maintenance of large systems are very difficult and also integrating and the components in most of the cases, where architectural design deviates from the original design. In this way architecture recovery [52] is an important component in the maintenance support like re-engineering and re-structuring. The use of domain knowledge and system documents gives major importance to pattern matching approaches in the reverse engineering community. As such we need to provide a user tool with cooperative environment for architecture recovery using reverse engineering tools. In addition to this, graphs are also used to represent the software systems. So from the engineering tools that are existing the reverse engineering committee [2, 17, 26, 29, 52, 54] adopted graph values.

More information is provided to the reverse engineer by the reverse engineering tools. In this mechanism filtering process is included to filter out information, which is useful for architectural recovery. The software views are still incomplete after the filtering process because of missing domain specific
information. As such the reverse engineer has to manually reengineer important software views.

1.2. Organization of the Thesis

Based on pattern matching in reverse engineering, Software architecture makes a Summary in chapter II. In this study and brief discussion of general techniques based on graphs and graph mining techniques [40] with Bipartite graphs is used and brief discussion of general techniques for minimizing context is also included.

The procedure of using an algorithm for minimizing the complexity involved in software architecture recovery by using bipartite graph is discussed in Chapter III

How SPIN algorithm is used to find the most associated data sets from the source graph presented in Chapter IV.

Chapter V includes improvements in matching process by using Error tolerant graph mining algorithm for Software Architecture Recovery.

Software Architecture Recovery through Fuzzy Clustering techniques is discussed in Chapter VI includes Recovery of software Architecture using Partitioning approach by Fiedler Vector and Clustering.

Chapter VII includes Recovery of software Architecture using partitioning approach by Fiedler Vector and Clustering.

Chapter VIII includes all the conclusions together and reiterates the Thesis statement for further studies in this area; a few recommendations are also given.
1.3. Contribution of the Thesis

Software architecture recovery thesis provides a milieu by using the techniques of Graph mining and clustering [3,10,28] to approximate graph pattern matching and architectural descriptive languages. In particular, the focus of this thesis is

“INCREASING THE EFFICIENCY OF SOFTWARE ARCHITECTURE RECOVERY BY GRAPH MINING TECHNIQUES”