CHAPTER 9

CONCLUSION

The objective of this thesis entitled "Development of Components with Effective Interoperability among Heterogeneous Platforms" is to find frequent itemsets with lesser number of scanning and without tree construction. Further in this thesis, a tool called Platform Interoperable solution (PIS), which provides the interoperability between the heterogeneous components EJB and .NET, is developed and implemented.

Many numbers of algorithms and techniques had been developed for finding frequent itemsets. The familiar algorithms are Apriori and FP-Growth algorithms. Chapter 2 described the familiar algorithms namely Apriori and FP-Growth for finding frequent itemsets. The Apriori and FP-Growth algorithms have been described with suitable illustration in this chapter. In Apriori algorithm, multiple number of scanning of transaction database is needed for finding frequent itemsets. FP-Growth algorithm can be used for finding the frequent itemsets with lesser number scanning compared to Apriori algorithm. FP-Growth algorithm finds the frequent itemsets by constructing tree. For every change in database new tree has to be constructed. Although FP-growth is fast, a major drawback is that in each recursive call a new FP-tree has to be built.

To overcome the above mentioned drawbacks, a new simple algorithm is introduced in Chapter 3. This algorithm reduces number of scanning of databases required compared to Apriori algorithm. And there is
no need for tree construction in order to find frequent itemsets as in FP-Growth algorithm. A comparison study is performed between this new algorithm and the familiar two existing algorithms, revealed that the proposed algorithm is simple, easy to implement, easy to understand and involves less computational work. With this new algorithm maximal frequent itemsets can be found effectively. In addition to these, a distributed component EJB has been created for this new, simple proposed algorithm. This component can be used by anyone, for having decision-making process to their business, by finding frequent itemsets.

Chapter 4 gave a brief outline of component based development system, integration of software components, and interoperability standards. This chapter also described the existing tools for providing interoperability between J2EE and .NET, which involve more manual process.

Chapter 5 outlined briefly, various existing interoperability technologies such as Web Services, Janeva and J-integra. Because of the protocol used in, Janeva and J-integra are better than WebServices. Janeva and J-integra are commercial products. Thus Janeva and J-integra are vendor dependent. Furthermore the classpath settings are very difficult in J-integra tool. Therefore an automated design tool called Platform Interoperable Solution Tool (PIS) is introduced in Chapter 6.

In Chapter 6, an automated tool called Platform Interoperable Solution Tool (PIS) for providing interoperability between .NET and J2EE components is introduced. This tool involves less manual process compared to the existing tools. This tool is developed using java only. So no extra software is needed to use this tool for interoperability. In PIS, IIOP .NET is used for configuration process. IIOP .NET is a .NET remoting channel based on IIOP protocol. In PIS, all the configuration steps will be stored in XML.
files. To configure a component into another form, PIS is used only once. Thereafter, straightaway the configured component form can be accessed.

For the integration of the components developed by different vendors say .NET and EJB, this interoperability tool PIS is very useful. It is not possible to integrate the components developed in different languages and in different platforms without having interoperability between them.

Chapter 8 briefly outlines a Banking Application which demonstrates creation of heterogeneous components (EJB and .NET) and their integration using the interoperability tool PIS.

Future development may be towards extending this tool for providing the interoperability among many other platforms.