CHAPTER 7

CONCLUSIONS AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

This thesis has been aimed at development of automatic software document generation using design patterns.

A knowledge based approach, frames, extended with perspective descriptors are used for content determination, i.e. identifying inputs required for document generation. The inputs have been identified from the requirements and analysis phase of software development life cycle. From the available software requirement specification techniques such as dataflow diagrams, entity relationship diagrams and state transition diagrams, the inputs required for requirement document generation such as process, subprocess, data store, data, entity, relation, cordiality, initial state, final state, action, event, trigger, post condition, pre condition etc. are identified and they are represented using frames with perspective descriptors. These perspective descriptors enhance effective reusability. Since it enhances reusability these perspective descriptors can be also used for story generation, where focus shifts will be incorporated in a wide manner. Apart for nature language generation perspective descriptors can be applicable for various applications (that require a large amount of reusability) such as morphological analyzers, if it is required to parse at different levels depending upon the requirement. In machine translation also perspective descriptors can be used if the deals with multiple domains. Apart
from using perspective descriptors, causal links and conceptual links are required in order to get the result in a better understandable way.

Using causal links and conceptual links we can retrieve necessary document based on the need and expensive of the user. The identified inputs are organized during the planning phase of the natural language generation. Abstract plan is obtained from the basic plan representation and these abstract plan provide necessary operators required in order to prepare software document based on the software standards. Since software standards gradually increments step by step, a hierarchical planning is required. Patterns, reusable components are used to organize the contents based on the standards. The output of the patterns will be the phrasal conditions. Patterns are used to obtain the content in an organized and uniform manner. For example design patterns can be used for various applications such as education, in which if teleducation has to be developed for different types of users different types of levels Generally patterns (can be widely used for manufacturing purposes).

Apart from design purposes, patterns are applicable for various activities such as Analysis process, debugging etc. During the realization phase, mapping the syntactic and semantic part is caused out. A generative grammar has been designed and developed for surface realization and linguistic realization. Based on the grammar software document has been generated.

Linguistic content can be delivered to a reader or hearer in various many ways. For example, a weather report might be presented to an audience via Email, Newspaper article etc. While generating continuous documents the semantic of paragraphs have to be considered (Zadrozny, W. and Jensen 1991).
7.2 FUTURE ENHANCEMENTS

Syntactic generation is one of the most elaborated and investigated fields in the area of natural language generation. In particular, due to the growing research in the Computational Linguistics area, syntactic generation has now achieved a methodological status comparable to that of natural language parsing. However, there are still strong limitations, which weakens their general applicability for arbitrary application systems. Probably the most basic problems are:

- Lexical and grammatical coverage
- Re-usability
- Limited functional flexibility

None of the syntactic generators process grammars whose size and status would go beyond that of a laboratory one. The newly proposed approaches in Computational Linguistics are in principle capable of processing declaratively specified grammars, and hence are potentially open to grammars which can be incrementally extended. However, as long as the grammars do not achieve a critical mass, the usability of the approaches for very large grammars is a speculation. The same is true for the status of the lexicons. Currently, generators only use small lexicons. Consequently most of the systems trivialize the problem of lexical choice as being a simple look-up method. However, if very large lexicons were to be used then the lexical choice problem would require more sophisticated strategies.