Chapter 8

CONCLUSIONS

8.1 Introduction

The need for real time expert systems is being felt increasingly and their design is receiving much attention during recent past. The significance attached to real time expert systems can be gauged by the vast and expanding literature on this subject.

In the preceding chapters, we considered an architecture and its implementation for developing real time expert system. The purpose of this chapter is to summarize the significant aspects of the previous chapters and suggest further work in this area.

8.2 A Summary of the Work

The need to design any system arises out of certain requirements not being met by similar systems already available. Real time expert systems have some specific requirements like continuous reasoning, temporal data and knowledge representation, interruptability and reactive response behaviour. These requirements could not be met by traditional expert system architectures. So, a new architecture has to be designed for real time expert system.

We started out with identifying the specific requirements of real time expert systems. This was followed by the identification of necessary features to be incorporated in the real time expert system architecture. The key requirements identified are

- Temporal Representation of Data
• Representing and Reasoning about Temporal Relationships

• Modelling Delayed Feedbacks

• Continuous Reasoning

• Simultaneous Handling of Multiple Events

• Interruptible Reasoning

• Embedded Operation

• Reactive Response

• Focusing Attention

• Handling Asynchronous Events.

Recently, many real time expert systems are being reported in literature. New knowledge representation techniques and inference strategies to address specific application requirements have been reported. A study of such techniques, tools and systems from the perspective of the above requirements was carried out. The results of this study are contained in Chapter 2.

Based on the specific requirements identified, a formal model to represent various aspects of a real time expert system has been defined. This model is an Extended Petri Net (EPN). The EPN model has the ability to abstract various aspects of real time expert systems like representation of temporal data and knowledge, knowledge base verification and reasoning. On the basis of the EPN model, an object oriented real time asynchronous production system architecture called REX has been developed. The key features of this architecture are

• an unified object oriented data and knowledge representation model.

• asynchronous multiple rule firing reasoning process.

• integration of external inputs into the reasoning process.
The proposed object oriented data and knowledge representation model is a distinguishing feature of REX. The data model has been designed to facilitate representation and storage of temporal data. Since, real time data have life spans, a 'timeout' facet has been provided to represent data life span. Unlike other expert system architectures, in which rules are defined above objects and are treated differently \[9\], rules in REX are treated as objects. A rule schema and taxonomy have been defined in REX. Three different rule types viz. Autonomous rules, Clock synchronised rules and Spanning rules have been defined. Autonomous rules deal with current data values. Clock synchronised rules represent knowledge about temporal relationships and deal with data to be obtained in future. Spanning rules pertain to trends in past (historical) data. These different rule types can effectively capture knowledge in real time domains.

The reasoning process in real time expert systems should meet the reactive response requirements and suit the semantics of the data and knowledge representation scheme. The reasoning algorithms in most production systems fail to meet real time response requirements. This is because of their combinatorial match phase. In order to achieve a reactive response behaviour it is necessary to reduce the amount of time spent in the match phase. This can be achieved by designing a suitable match algorithm. The design of a match algorithm should take into account the rule semantics. Most match algorithms model the match process as a sequence of Select and Join operations on the working memory. In a similar vein, REX models premise evaluations as query evaluations with Select and Join operations on the working memory. The evaluation of a Spanning premise is different from that of other premises. The evaluation of a Spanning premise can be modelled as an aggregation query on a set of object instances. With these models of evaluations, a new match algorithm to suit REX rule semantics was designed. It has been shown that the number of evaluations made per attribute update is a linear function of the number of premises and events.

Real time expert systems have to deal with simultaneously occurring multiple events in the external world. If the concept of single rule firing in an inference cycle is adopted, then it will not be possible to handle multiple events occurring simultaneously. So, the concept of firing multiple rules concurrently in a single inference
cycle has been adopted in REX. If a multiple rule firing model is adopted, it is necessary to ensure the integrity of the working memory. This process in production systems is termed as interference analysis. Interference analysis techniques reported in literature are compute intense. Some techniques are prone to deadlocks. We have proposed a deadlock free interference analysis algorithm and proved its correctness. The algorithm is $O(n)$ complex, where $n$ is the number of matched rule instantiations. This algorithm is not specific to REX alone, but can be used to production systems like OPS5.

The current prototype is being used to implement an aerospace application.

8.3 Scope for further work

The size of the knowledge base in real time expert systems increases drastically with the increase in application complexity. In such cases, it is necessary to adopted a distributed problem solving paradigm. Two specific issues have to be addressed if such a paradigm is adopted. They are

- knowledge base partitioning and
- communication paradigm

These two factors are interwoven and can have a significant effect on the real time performance of the system. Though work on partitioning for real time expert systems is reported [7], it is necessary to study the effect of partitioning strategies on the communication. Similarly, the effect of communication paradigm on rule base partitioning have to be studied. Further work in defining the cooperative system paradigm and its implementation is under progress.

Another possible direction is in predicting the upper bound on the match time. Recently, a method that predicts a run time upper bound on the match time for RETE is reported [89]. The worst case upper bound predicted by this technique is five time more than the actual run time. Further work on developing such prediction
techniques for the REX system can be undertaken. It is also necessary to develop techniques whose predictions are more accurate.

Interference analysis is performed using serializability as the correctness criterion. It is observed that serializability is a strict criterion in real time systems. Correctness criterion based on application semantics are being proposed for real time databases [41]. The applicability of such criteria to interference analysis in real time expert systems can be studied.