Chapter 7

Conclusions and Scope for Future work

7.1 Conclusions

As part of this research work, a novel approach of enhancing the self-management capabilities in terms of self-healing and self-configuration of a software product has been proposed. Enhancing the self-management capabilities of a software systems, reduces the IT costs for companies, simplify management of their IT resources, realize a fast return on their IT investment and provide high levels of availability, performance, security and asset utilization.

In the first phase traditional approach of enhancing the self-management capabilities in terms of self-healing of a software product (IIS-server) that supports prototype implementation of MAPE-K (Monitor, Analyze, Plan, Execute-Knowledge) autonomic Computing reference model has been implemented. The structure of symptom database is simplified by using a variable length record-field structure and these records are stored in hast-table that has constant access time. It is estimated that for a table having ‘n’ symptom records, the symptom access time in terms of average path length would be $O(1) + |k|$, where ‘k’ is the length of the chain. On the other hand when the XML structure was used for the same, estimated symptom access-time in terms of average path length would be $L_{avg} = \sum_{i=1}^{K}(pli * pl)$. It is observed that for
n=100 symptom records, an improvement in symptom access-time was found to be 30% as compared to the traditional approach. Also, it is quite evident that the self-healing time and action execution time is improved in real time whenever the symptoms are matched by the ASM software.

Research findings:

1. This work presents a simpler method of implementing IBM’s MAPE-K framework of Autonomic system that supports that supports simpler structure and easier mechanism to express the symptom rules and associated actions in human readable form.

2. It improves symptom reference time by 33% and facilitate better problem determination task to administrator (ASM) with significant improvement in healing in action taking time.

3. It also empowers administrator with the ability to create and maintain primitive policies and user defined actions for any application in a convenient way.

In the second phase, the complexity of problem determination in the process of autonomic computing is handled by Know-how model to support heterogeneity of log-files of various software products. The Know-how is supplied by the product vendor in the form of Regular expression and language grammar which are generic in nature. The ASM would then edit the Know-How structure and writes symptom policies in the action part of the production rule using C-language constructs. The ASM can also attach required actions routines for each symptom policy as a corrective major whenever the symptom is matched. This avoids writing of multiple log adapters for each application. It is observed that for 30KB of log-file size of IIS and Apache. 35 lines and 300 lines of Adapter code has been eliminated respectively. This novel approach saves the time and cost of code writing and helps IT professionals to focus more on the service that they provide to the customer.

Research findings:

1. The heterogeneity of log-files of various software products are supported by novel approach called as Know-How in the process of problem determination task in Autonomic computing system.

2. The proposed ASM system eliminates the need for writing the Adapters for every log-file and intern saves the time of Autonomic System Manager.
3. Symptoms and associated actions can be added conveniently in the rule section of the Know-How –YACC-specification.

4. Specialized LR parsing technique called canonical sequences of reductions would be used to map log entries to the appropriate segments of symptom catalog and there by 30% improvement in healing time and action taking time could be achieved.

5. This Novel approach helps IT professionals to focus more on the service that they provide to the customers and business partners rather than maintenance.

In the third phase of the research work to support the above two methods, a platform is provided to convert the symptom data to a Policy Base (symptom catalog). It incorporates processing engine that gets connected to Know-How structure for editing symptoms, actions and makes it easy to build Autonomic System Manager (ASM) database. Administrator can add /delete /modify complex policies and nested policies using the language editor that strengthens the symptom catalog. The Language editor designed for ASM presents a user friendly way of defining boundaries for the working of the applications. Also, ASM can derive the new policies from existing policies. The nested policy feature helps to generate new policies based on existing policies. The newly generated policy remains unchanged even when the policies on which it depends are deleted. The systematically written policies, strengthens the symptom database that enhances the self-healing capability of a software products.

**Research findings:**

1. The work strengthens the policy building capabilities to the symptom catalog with following editing features for symptoms for a Self-healing Autonomic System.
   a. It provides a well defined language and user friendly editor for building databases, such as Appbase, Logbase, policybase and Actionbase for autonomic system. This strengthens the policy building capabilities of symptom catalog.
   b. Reduces programming effort required for writing Adapters for each software product and supports heterogeneity of log-file formats.
   c. ASM language editor empowers administrator with the ability to create and maintain complex policies for any application along with manual action execution, in a convenient way so that the ASM can automate the application easily and can find solution for recurring system problems.

In the last phase, Self configuration feature of autonomic computing has been attempted. As an extension of application, a Wireless Sensor Network (WSN) is chosen to show the effective utilization of power by proposing a Self-configuration feature for memory management of
sensor node. In this architecture, the memory module is divided into several blocks known as memory banks that are activated by the memory controller unit and power-switching module. Depending on the amount of traffic flow the memory banks are activated. Simulations have been performed for varying interarrival and service times using the M/M/1 queuing model. The graphs and simulations show a small difference in the energy utilisation at the initial stages when there are fewer packets in the queue and the service time is low. As the number of packets getting queued up and service time increase, the energy utilization at the sensor node is reduced by an average of 70% in modified memory architecture as compared with normal memory architecture. Also by considering the packet drop as performance parameter it is observed during the simulations that a variation in packet arrival is observed, i.e., interarrival time, where more packets get queued up to be serviced and the modified memory architecture show less packet drop in performance as against the normal architecture. This increases the node lifetime and network lifetime.

Research findings:

1. Modified memory architecture for sensor node performs significantly better than the normal memory architecture. Considering the idle energy consumption as performance parameter, the energy consumption is reduced by an average of 70% when the packets are queued up.

2. Considering the packet drop as performance parameter, it is observed there is a less packet drop in case of modified memory Architecture as compared to Normal, hence there is 70% increase in node and network lifetime has been achieved.

3. Quality of service is enhanced.

7.2 Scope for future work

The process of providing the self-management capabilities into an IT infrastructure is an evolutionary process. It must be implemented by every organization through the revision and adaptation of autonomic computing technologies and skills. The IT industry in particular, software systems will further develop self-management capabilities to help improve staff productivity, increase IT business resiliency and reduce operating costs. Following are the future scope for extending the work in creating and enhancing the self-managing autonomic computing capabilities into the software products for improved operational efficiency, supporting business needs and workforce productivity.
1. To explore the possibility of evolving new techniques to develop a new open standards for autonomic computing systems to deploy the autonomic managers across the IT infrastructures.

2. Development of new mechanism to expand the depth and breadth of intelligent behavior of the agents in ABLE tool kit in ABLE toolkit.

3. To explore the new mechanism for Support global communication between Managed resources and Autonomic Manager in IBM’s Autonomic Computing Toolkit.

4. To evolve a new framework that supports the Web Services Distributed Management (WSDM), for managing heterogeneous resources.

5. The Monitor and Execute functions of Autonomic manager are automated at the basic and managed level of autonomic maturity. There is a need to automate the additional parts (Analyse and Plan functions) of autonomic manager to higher levels of autonomic maturity that includes predictive, adaptive and autonomic.

6. To evolve a technique that has an ability to manage the StarMx framework and its properties dynamically.