Chapter 5

CONCLUSION AND SCOPE FOR FUTURE WORK

This research work pertaining to software estimation involved review of literature, software industry experience survey, implementation of models/methods involving AI techniques and technologies, which are proposed by researchers, experimentation with modification and combination of methods/models and analysis of results.

Software development project life cycle with irrespective to process model is broadly divided in three stages such as Initial, Intermediate and Final. This research work has focus on first two stages only. During initial stage, Request for proposal is available. Requirements are stated in textual form. With less clarity of problem as well as solution domain, Human expert estimates effort by applying experience as well as intuition. During Intermediate stage, requirements are clear, analysis and design models are developed hence more knowledge about the project is acquired. This stage is suitable to apply formal models.

With reference to software estimation, It is realized that Expert Judgment Method has been superior but the accuracy is dependent on experience of human expert estimator. With change in project, process, personnel, product accuracy is not guaranteed. Algorithmic or Empirical model such as COCOMO II is easy to use. The level of experience required for estimator need not to be application area specific. It is widely accepted model but accuracy of this model is dependent upon availability of calibration data.

It has been revealed that estimator cannot rely on only one method or model. Accuracy of estimates derived by any method cannot be determined unless project is over. Hence, in absence of actual effort, use of more than one effort estimation method is a recommended practice. In this dissertation, at each stage combination of more than one method is proposed.
Results obtained from experiments strongly support to conclude that AI techniques and technologies applied in combination with formal model improves accuracy of estimates.

COCOMO II proposes sub-models, suitable for initial and intermediate stage. The final conclusion of experiments are presented here with reference to stages.

5.1 EARLY STAGE ESTIMATION: IMPLICATIONS

Software Project Document Matching by LSA, Ontology representation by semantic graph and use of Wordnet complements manual identification of similar projects and reduces effort and time required to find similar project from knowledgebase.

AI model using knowledge processing by means of ontology representation and semantic similarity finding proved that similar project identification is simplified for novice estimator. When estimator has less experience and less information regarding project, similar project data is useful for estimation at earlier stages and helps to reduce deviation of estimates from actual.

It is also concluded that, when more than two methods for similar document identification are used, the intersection of output i.e. similar project RFP documents found are really similar, which were seconded by manual verification.

Though in experimental setup results are encouraging, In practical application the usability is constrained by following limitations:

- LSA captures semantics but cannot handle word sense disambiguation. Similarity identification of documents need more meaningful representation

- Ontology is represented in the form of Semantic Graph. The relationship is established between Concepts and Categories as well as Categories and Software Project Documents. Human intervention is also required to establish relationships. This is required when new document is to be added to the
knowledgebase. To minimize, human intervention comprehensive ontology beyond domain is also required, which is rarely available or difficult to build.

- **Wordnet**: Lexical database provides help to identify cognitive synonyms (synsets). These are interlinked by means of semantic and lexical relations. When semantic graphs is to be compared. The complex algorithm is required to identify similarity. Complexity lies with hierarchical level matching.

In spite of limitations and constraint imposed, these technologies proved usefulness. Further evaluation and enhancements are needed to mimic human expertise in identifying similar projects.

### 5.2 INTERMEDIATE STAGE ESTIMATION : IMPLICATIONS

Estimation model involving Neuro-Fuzzy Inference System integrated with COCOMO II, reduces error and improves accuracy compared to only COCOMO II. Further the accuracy is increased by providing cost driver values on continuous scale by Slider. This improvement is notable by result. Mean Absolute Error for the set of 10 projects for ANFIS with Slider is 29.829 where as ANFIS without slider is 59.565 and only COCOMO is II 99.123 (Table 4.2,4.3 and Figure 4.6).

Neuro-Fuzzy Model is centered around COCOMO II. Limitation of COCOMO II are inherent to data available from past project used for training the neural network. Data referred for ANFIS training is from COCOMO literature. The project experience in previous decades are suspected about usefulness for contemporary projects. This data proved usefulness to support test the proposed model. Enterprise-wide knowledgebase need to be maintained and proposed model can be used for estimation. Unfortunately industry data was unavailable to test the proposed model. Hence experiments are carried with data made available by research laboratories.
The Usecase Point and Class Point Sizing is yet to be matured and evaluated thoroughly. Since it is based on abstract representation, accurate estimation can never be claimed just by using these methods, but these can be complementary to others.

Instead of making claim for universal applicability, with fully matured enterprise-wide knowledgebase and application of these AI models along with established empirical model and human expert, will prove its usability by improving accuracy in effort estimates at earlier and intermediate stages of software development project life cycle.

5.3 CONSOLIDATED ESTIMATION MODEL FOR INITIAL & INTERMEDIATE STAGES

The sizing has been important issue while using any algorithmic model such as COCOMO II. It accepts Size in terms of KSLOC (thousands of Source Line of Code). The Function Points emerged as dominant size metric, which is irrespective of programming language. Application Composition model of COCOMO II use Object Points and other model make use of KSLOC. COCOMO II literature has also provided table for conversion of Function Point to Source Line of Code in different programming languages. COCOMO II also elaborated adjusting KSLOC Size when COTS are used.

The proposed model is complementary to expert estimator, but not a replacement. At each stage of software development process, COCOMO II Model may be used as reference to estimate. Additional sub-model is proposed depending upon stage. At each stage provision is made for input from Expert Estimator.

Before commencement of project Request for Proposal document is matched and similar projects are identified from historical knowledge-base consisting of semantic representation of past project documents along with post-mortem metrics. Expert Estimator is allowed to select project from matched project list generated according to similarity index and intuition by experience.
The assumption is made that during intermediate stage, usecase model and class model is available. Trained Neural Network is also available as Knowledge-base component. Neuro-Fuzzy model integrated with COCOMO II with modified interface is used and output range is available as reference estimate. Usecase Point & Class Point based estimates can be generated and averaging along with expert input is used to generate estimate.

During further stages, effort multipliers and scale factor values are modified as more project details are available. Expert estimator has to play important role while selecting input. Parallel models will restrict expert estimator to drift drastically away from actual.

In both of these software estimation method human expert plays important role. AI approaches mimic human problem solving techniques involving knowledge processing. AI Model for Software Estimation improves accuracy and complements to the Human Expert Judgment Method as well as Empirical / Algorithmic Model (COCOMO II).

5.4 LIMITATIONS & FUTURE SCOPE

The availability of knowledgebase content in the proper representational structure is major bottleneck. Enterprise-wide knowledgebase can be developed and made available by integrating with CASE tool. Semantic representation is another limitation. Research is going on these issues, specifically automating the semantic representation from text documents. The completeness and usefulness of semantic representation is highly dependent on content and format of text documents. There is need of improvement in CASE tools to establish linkage between models and additional knowledge acquired about the project, process, problem and people during development. Overall integrated knowledgebase will be useful for estimation process. Use of Wordnet is to be evaluated for semantic graph based ontology development and similarity identification. There is scope to modify algorithm based on Wordnet to identify partial graph structure matching with other partial semantic graphs representing documents. The Artificial Neural Network can be applied for document clustering, further which can be used for similarity identification and software estimation.
Use case model is developed at the beginning of intermediate stage. The gist of requirement is modeled from user’s perspective. The use case can be an important component of project knowledgebase. Graphical representation is represented in XML format with enterprise-wide customized XML Schema. This can be transformed into a semantic graph-like structure and a Wordnet-based algorithm can be applied to find similar use case representation and identify similar projects.

5.5 FINAL WORD

AI models for software estimation are applied with present technologies to improve accuracy of estimates even though vague, insufficient, and unclear requirement information related to software is available during initial and intermediate stages of software project development with respect to any process model. It is also experienced that only one method or model or technique for software estimation cannot be relied at any stage, hence more than one approaches need to be followed. This must be convincing to human estimator, otherwise human experience must be respected and accordingly parameters and inputs used for applied model should be verified and adjusted. The proof for suitability can be obtained only when these proposed AI Models will be integrated into an estimating tool and used for a enterprise with enriched experience reflected into input parameters over the time.