A Unified Approach towards Software Requirement Clarification Using Mind Map

Chaitanyakumar Patel[1] Harita Chocha[2]
[1,2] Department Of Computer Science And Engineering,
[1,2] Institute Of Technology, Nirma University, Ahmedabad,Gujarat, 382481

Abstract---- Software development process is challenging and ambitious process. Software requirement specification which gives the developer a direction in designing and development of software and to clarify these requirements mind mapping technique is used to elucidate the software requirements. This paper aims to use the mind mapping method along with extensive elicitation methods to find out the software requirements specification.

key world: - UML – Unified Modeling Language, mind map, NLP – Natural Language Processing

I. Introduction

Software, an untouchable and intangible part of a computer systems. Software is developed and not manufactured like its counterpart hardware. Software development is a very difficult process. Since a software not like a hardware can be touched or seen in its intermediate development stages, makes software development a challenging process. Finding out the software requirement specification is a very tedious, time consuming and expensive work. Improper identification of requirements can lead to extra expenses, missing some key features, delayed release or even scraping of the software. Software requirement identification cannot be just collected from the user. Requirements elicitation includes interviews, questionnaires, user observation, workspace, use cases, prototyping and brain storming etc. Requirement elicitation is an integral part of requirement engineering process before requirements can be analyzed, modelled and implemented. Example of requirement elicitation process is a meeting between the software engineers, stack holders and the customer to discuss about the domain and expectation from the process.

This paper represents a basic idea to simplify this tedious process of extracting key components of the software requirements and using mind mapping technique to generate a unified model of the software requirements and specification.

With the aim to focus on user expectation, designing a model that helps in identifying various requirements clearly. A step further is taken in the generation of various UML diagrams. In the proposed work the clarification process is divided into four phases. In the initial phase the NPL processor processes a text document containing the information collected from requirement elicitation. Next a pre-defined mind map is generated or designed by the software engineer. In this phase the extracted verbs and nouns matches with the mind map then it is converted into domain specific mind map. In the third phase mind map will be extracted. The final phase converts the extracted mind map into a UML diagram.

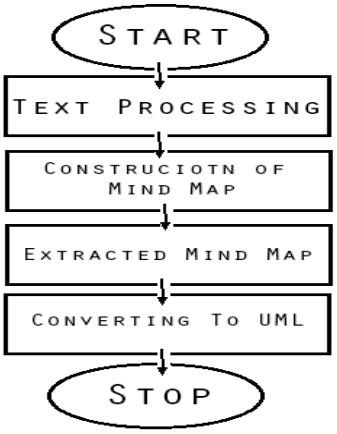


Fig 1: Flowchart of the proposed process

II. Detailed Description

The proceeding section describes the detailed process behind the proposed idea. As mentioned earlier it has four phases. But it also has a 'phase 0' which lays down the foundation of the proposed model.

The 4 major phases are:

- Text Processor Mind Map Construction
- Extractor
- UML-mind mapper

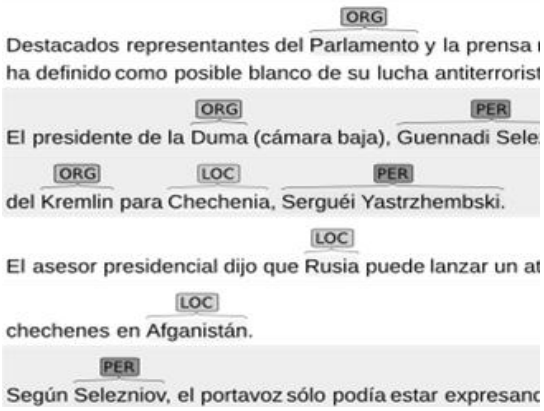
A. Phase

The phase 0 includes the rigorous process of specifying the domain of the software. It includes the elicitation process as in [1] which includes include interviews, questionnaires, user observation, workshops, brain storming, use cases, role playing and prototyping. Before requirements can be analyzed, modeled, or specified they must be gathered through an elicitation process. Requirements elicitation is a part of the requirements engineering process, usually followed by analysis and specification of the requirements.

B. Phase

After the exact definition of the software is derived from the elicitation process next goes the phase takes the document produced from the elicitation process containing the key nouns and verbs. The document is analyzed and tokenized using NPL as given in reference [2].

Expanding on the topic. The NLP process the text documents extracts nouns and verbs. This are a useful in making modal classes and removing useless classes. This helps in creation of a clear classes.



*ORG - organization, LOC – location, PER - person

Fig. 2: Entity mention detection

Example: CoNLL 2002 Shared Task: Language-Independent Named Entity Recognition as in reference [3]

C. Phase

After the detection of key words i.e. the nouns and verbs from the text document. The next phase is construction of a mind map. There are many software, freeware and paid, available that helps in the process of mind map construction [4].

Mind Map: Mind map can be understood from the figure 3. It can be described as a visual outlining of information. As can be seen from the figure 3, it is created around a single word or text. The associated ideas, tasks and sub categories are surrounding the word. Construction of mind map purely depends on the creativity and imagination of the software engineer. The software engineer creates a mind map using his deciphered model from the requirement document.

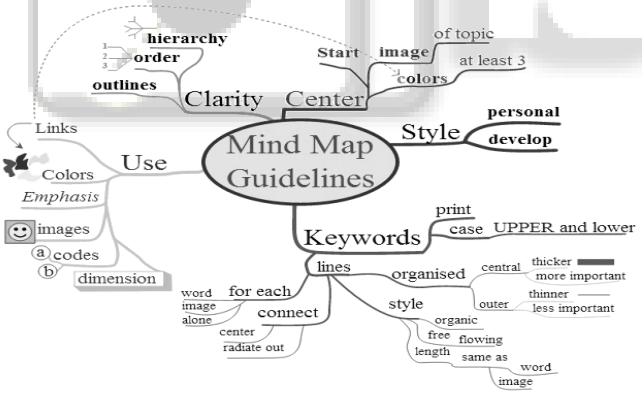


Fig. 3: Mind Map Guidelines

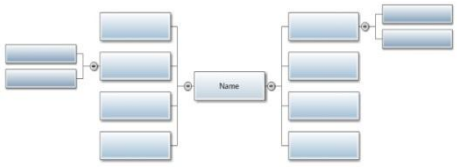


Fig. 4: A screenshot of the software used in reference [4] to create a mind map

D. Phase

From the constructed mind map and the extracted keywords now we need to make a domain specific mind map. This helps in removing of unassociated tasks, ideas from the mind map along with the addition and modification of the extra tasks. The keywords help in generation of rough mind map, these are then compared with the constructed mind map. After the process we get a domain specific mind map.

Illustrated in figure 5 and figure 6 below. Figure 5 shows a mind map designed by an engineer. After comparing from the tokenized document from NPL we find out advertising missing so this module need to be added as shown in figure 6.

Fig. 5: A supermarket mind map designed by engineer.

Fig. 6: Mind Map after comparing with the domain elicitation in Phase I

E. Phase

The extracted mind map is then converted into UML diagram. This phase can be done either using a software [6] that converts mind map to UML or manually using the UML designing software. The key advantage in this process is that we can get a better and clarified UML diagram as we have a domain specific mind map. The entire process focuses on the domain of the software and constructing a UML diagram to get a better and clear picture of the software.

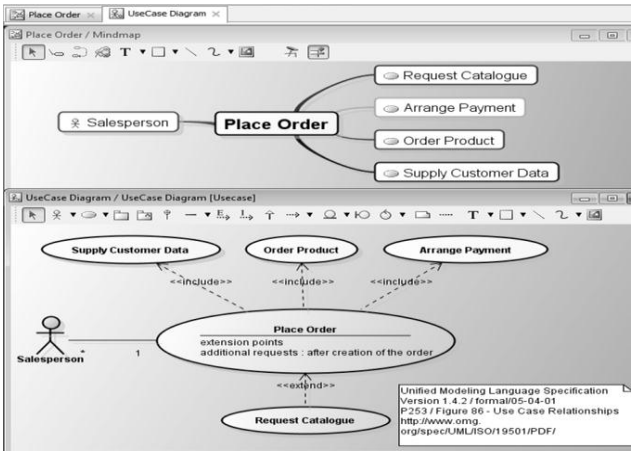


Fig. 7: Screenshot of software [6] that converts mind map in use case diagram

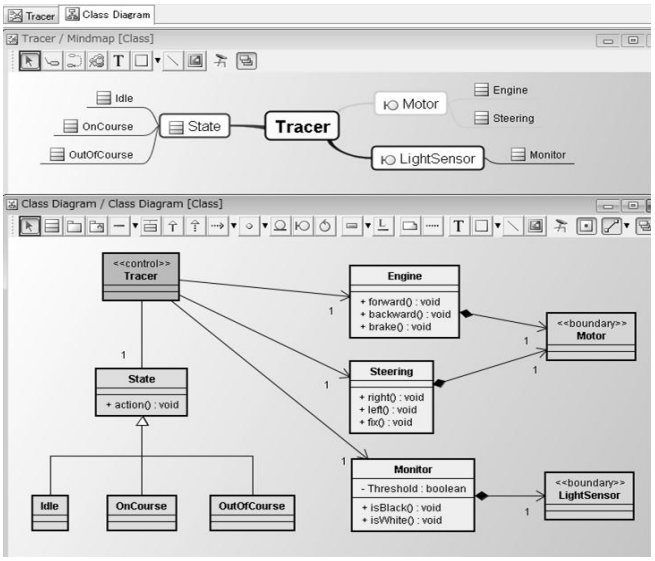


Fig. 8: Screenshot of software [6] that converts mind map in class diagram

II. Conclusion

There are various advantages and disadvantages of the proposed model.

Some of the advantages are:

- It helps in domain identification and putting forward a clear picture of the proposed software model.
- Helps in the accurate creation of various UML diagrams which are not possible with the conventional method.
- An automated software can be generated which can help reduce the pen and paper work of the developer.

Some disadvantages are:

- It requires a lot of time in creation of an accurate mind map.
- As it is based on a human designed software it is prone to error.

Acknowledgment

We would like to thank Dr. Madhuri Bhavsar for her valuable guidance and motivation. It would not have been possible to complete this paper without her proper direction.

References

[1] http://www.cs.iastate.edu/~cs362/handouts/handout02.pdf
[2] http://web.cs.hacettepe.edu.tr/~ilyas/Courses/BIL711/lec05-pos.PPT
[3] http://brat.nlplab.org/examples.html
[4] http://www.smartdraw.com/
[5] http://en.wikipedia.org/wiki/File:MindMapGuidlines.svg
[6] http://astah.net/features/convert-mindmap-to-uml
[7] http://www.ijert.org/view.php?id=3424&title=an-integrated-approach-towards-software-requirement-elucidation-using-mindmap

