

Speech to UML: An Intelligent Modeling Tool for Software Engineering

B.Venkata Durga Kumar¹

School Of Computer Technology; Sunway University College, Malaysia

Abstract

Software engineering tools are the key role in software development. The main focus, as the title implies, is on the use of Artificial Intelligence in software modeling tool. The key concern here is to create a user friendly and intelligent tool, which can be used to assist software engineers in the analysis and design stages and it, can even be used in educational institutions for various learning purposes. The key feature of this tool will be automatic generation of classified requirements, use case diagram, and object interaction diagram (OID) from a text based case or voice which will be first converted to text. Besides that there is no similar existing software which contains fully functional, which has a feature that directly converts speech to text and then to requirements and modeling diagrams.

This paper discusses an existing modeling tool issues and implementing a new modeling tool using speech reorganization tool, and an artificial Intelligence concept, and aims to design and implement a software-modeling tool to extract software requirements and develop different UML representations.

Keywords: *Modeling tools, UML representations, Intelligent Systems, Speech recognition, Artificial intelligence.*

1. Introduction

Artificial intelligence (AI) as a science was founded by three generations of researchers. The goal of artificial intelligence as a science is to make machines do things that would require intelligence if done by humans. AI researchers attempted to simulate the complex thinking process by inventing general methods for solving broad classes of problems [7].

The strong artificial intelligence view is that the only good AI is that produced by logic and reasoning has had very limited successes outside the closed world of the laboratory. Successful AI-based systems that have withstood the harsh reality of a rough, competitive marketplace have usually relied on biologically-based systems. AI was divided into broad paradigms based on models that were inspired by symbolic logic, with the other main branch inspired by biological processes [4].

¹Corresponding Author: B.Venkata .Durga Kumar
Email: burra@sunway.edu.my

Unified Modeling Language (UML) models are constructed graphically, by drawing diagrams. However it is not easy to manipulate diagrams using the CASE-tools of today, typically functionality and information hidden behind complex menu or dialog hierarchies, diminishing the usability of the tools. Speech recognition can be used to promote usability of an existing tool as a complementary user interface that allows simultaneous use of other interface. The accuracy of speech input is affected by several factors [20].

According to Peacocke (1990), speech consists of isolated words is easier to recognize than continuous speech. Many studies suggest that instead of using speech recognition alone, a faster and more natural way to work is to use gesture or a pointing device simultaneously with speech. According to Samuel Lathinen (2005), UML is factually not one single language, but a set of language.

For example, the UML class diagram notation forms one language, and the sequence diagram notation forms another one. This means that instead of one big grammar we have a set of smaller grammars that can be activated as the need arises, and thus get even better accuracy for speech recognition. In order to use UML the user should have knowledge of the vocabulary and its restrictions. Because UML models are presented, constructed and manipulated as diagrams. The user has to know the notation of the elements as well. Consequently, the vocabulary is known beforehand by the system as well as the users. This makes the interpretation task even more accurate, and also eases the user's learning curves. Likewise, due to the common basic vocabulary, it is possible to construct a spoken language that can be used similarly in all of the UML tools, and even use multiple tools simultaneously through a coherent user interface [12].

UML is supported by several CASE-tools where the models are presented, constructed, and manipulated as diagrams. According to our experience, many of the potential UML CASE-tool users feel that the tools do not offer enough support for software engineering, reducing the tools to the level of ordinary drawing tools or pen and paper. At the same time they think that the CASE-tools are complex and hard to use. Consequently, many do not use the tools [13].

2. Issues:

2.1 In the current tool, developer chooses to make any changes to the requirements list; the changes to the diagrams must also be done manually. It takes longer than developer might think even by using complicated design tools, as they have to go through the requirements and match them with the diagrams.

2.2 The major concern would be when the system analysts conduct interviews with the clients, the recordings of the interview are usually replayed and the requirements are extracted. This process is error-prone because the analyst might miss some important content mentioned if the recorded voice clip is not clear enough.

2.3 The existing software modeling tools are not fully intelligent systems, these require user to analyze the case study and manually produce the requirements and develop and draw the UML designs manually. Most of the software design tools available in the market are very complicated to understand and use, these commercial software's are far too much expensive for the small and medium scale companies.

3. What is Speech Recognition?

Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. The recognized words can be the final results, as for applications such as commands & control, data entry, and document preparation. It can also serve as the input to further linguistic processing in order to achieve speech understanding. Speech recognition systems can be characterized by many parameters. An isolated-word speech recognition system requires that the speaker pause briefly between words, whereas a continuous speech recognition system does not. Spontaneous, or extemporaneously generated, speech contains dissiliences, and is much more difficult to recognize than speech read from script. Some systems require speaker enrollment and a user must provide samples of his or her speech before using them, whereas other systems are said to be speaker-independent, in that no enrollment is necessary. Some of the other parameters depend on the specific task. Recognition is generally more difficult when vocabularies are large or have many similar-sounding words. When speech is produced in a sequence of words, language models or artificial grammars are used to restrict the combination of words [18].

4. Speech Recognition

Speech recognition has been a popular topic with technologists for decades. Although speech recognition has existed in research laboratories and as a part of extremely sophisticated computer system, only now has the technology become sufficiently advanced to be practical in typical household and business computer systems. The speech recognition software that is part of these typical systems can be divided into two categories: continuous and discrete [19].

Continuous recognition systems are the technology promised by science fiction several people interacting with a computer simultaneously and using normal language and speech patterns. Such systems go far beyond simply recognizing speech. They are able to actually understand the meaning of spoken language. These systems are currently error prone, and extremely expensive to develop. The phonology, grammar, and semantics of English require a lot of computational power to interpret the complexity and reach to lthe subtleties of spoken language. The more pragmatic technology commercially available today is discrete recognition [15].

The system can recognize a predetermined lexicon of commands which represent predictable tasks the system can perform. Discrete recognition software relies strictly on phonological input, not the grammatical or syntactic interpretation used by continuous recognition systems. Unlike continuous recognition, the speaker may only use specified words and phrases and must pause between users speak one of the predetermined commands [19].

5. Speech synthesis

The term speech synthesis refers to the technologies that enable computers or other electronic system to output simulated human speech. They provide acoustic information that is phonologically acceptable yet has meaning to human listeners. Speech synthesis has an even longer history than speech recognition but is still an evolving technology used for reading computer screens and providing verbal instruction, feedback, or assistance [17].

6. Concatenated synthesis.

Concatenated synthesis uses computer assembly of recorded voice sounds to create meaningful speech output. Because concatenated synthesis uses recorded human voice sounds, it tends to sound more natural than formant synthesis, which uses machine-generated speech. The basic process for developing concatenated synthesizers is to have a human reader read units of speech and store the recorded units speech. These units are then assembled on demand according to given business rules. This is cost-prohibitive for many applications because of the necessary storage space, computational power required for assembly, and myriad of speech units required [10].

7. Speech-to-Text

The Speech-to-Text task (STT) is the core EARS research task. The fundamental program goal is a substantial improvement in STT system performance, measured in terms of overall word error rate. In addition to requiring thousands of hours of audio data in support of this goal, sites also need corresponding transcripts in order to develop language models and provide for system training. Benchmark data is also needed to allow sites and program sponsors to measure performance on a stable test set [16].

8. Description

This tool can be used to assist software engineers in the analysis and design stages and it can even be used in educational institutions for UML design learning purposes.

9. Functions:

- Convert speech (interview) to text
- Extract the requirement list from the text based case study and classifies the requirements as functional or Non-Functional
- Generate use case and OID model based on the requirements derived.
- Editing and deleting the requirements in the list generated from the case.
- Export diagrams as jpg graphics format.
- Save the requirements list in a formatted word document or as an html page.
- Direct diagram printing capabilities.
- Allows deleting and modifying of the requirements generated.

10. Methodology

Methodology to be used for this work analysis and design is Object oriented design, and the tool will be created using Object oriented language and modeling. Figure 1 show the architecture to generate speech to UML diagrams.

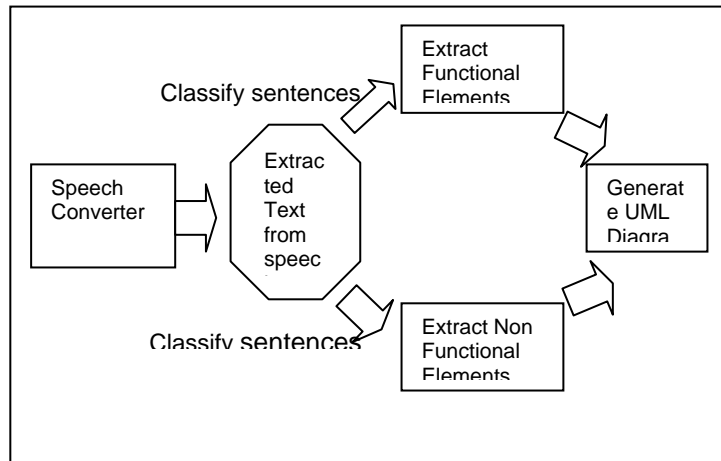


Figure 1: Architecture

11. Research Component

Artificial Intelligence techniques are used to extract the requirements list and to generate the modeling diagrams, using existing AI algorithms to convert speech to text and Integrating AI techniques with java classes. Make use of SQL to communicate with SQL Sever to retrieving and modifying the data and components.

12. Modeling

Systems analysts and designers produce models of systems. A business analyst will start by producing a model of how an organization works: a systems analyst will produce a more abstract model of the objects in that business and how they interact with one another: a designer will produce a model of how a new computerized system will work with the organization. The best way to understand what we mean by a model is to look at an example: In the Unified process (the method of developing systems that is promoted by the developers of UML (Unified modeling Language). In the UML, the rules are laid down in the OMG (Object Modeling Group) Unified Modeling Language. UML consists mainly of a graphical language to represent the concepts that we require in the development of an object-oriented information system. UML diagrams are made up of four elements: icons, two dimensional symbols, paths, and strings [15].

13. Research Problems

- Software Engineering tools are the key role in software development.
- In the current tool, developer chooses to make any changes to the requirements list; the changes to the diagrams must also be done manually.
- The current system takes longer than developer might think even by using complicated design tools, as they have to go through the requirements and match them with the diagrams.

14. Conclusion

In industry the UML has become the standard notation for modeling object-oriented systems during analysis and design in particular, these techniques include the use of the UML, which is a widely accepted standard notation for describing object oriented software. To generate modeling using UML with out facing any problems, I come up with this idea that is speech to UML, Using speech

recognition-based language tool software demonstrably improves the mean analysis abilities of a heterogeneous group of developers. However, results from the tool indicate that use of the software model was beneficial for the industry. The key feature of this tool will be automatic generation of classified requirements, use case diagram, and object interaction diagram (OID) from a text based case or voice which will be first converted to text and then to modeling. This tool helps the industry to save the time during analysis and design stage, and also it reduces the cost of the project.

References

- [1]. Alex J. Champanand, (2003). AI Depot, Extracted on 2-2-2008, from: <http://ai-depot.com/Knowledge.html>
- [2]. IBM(b), (1990). IBM International Technical Support Centers, Object-Oriented Analysis of the ITSO Common Scenario-Document GG24-3566-00. IBM International Technical Support Center, Raleigh, North Carolina.
- [3]. IBM(c), (1991). IBM International Technical Support Centers, Developing a CUA Workplace Application-Document GG24-3580-00. IBM International Technical Support Center, Raleigh, North Carolina, 1990.
- [4]. Joseph C. Giarratang, Gary D Riley,(2005). Expert Systems, Principles and Programming, NY, Thomson course Technology, pp55-59
- [5]. Kondo, A. M. Ahmet M.,(2004). Digital speech: coding for low bit rate communication systems, Chichester : J. Wiley, pp72-132.
- [6]. Luger, George F. (2005), Artificial intelligence: structures and strategies, Harlow: Addison-Wesley
- [7]. Michael Negnevitsky,(2002). Artificial Intelligence, A Guide to Intelligent systems, NY: Addison Wesley, pp 2-7
- [8]. Negnevitsky, Michael, (2002), Artificial Intelligence: A Guide to Intelligent systems, Harlow, EN: Addison-Wesley.
- [9]. Quatieri, T. F. Thomas F., (2002). Discrete-time Speech Signal Processing : principles and practice, Upper Saddle River, NJ : Prentice Hall PTR, pp34-81.
- [10]. Rabiner, Lawrence R,(1993). Fundamentals of Speech Recognition, Upper Saddle River, NJ : Pearson Education, pp27-51.
- [11]. Rumbaugh, J. M. Blaha, W. Premerlani, F. Eddy, and W. Lorenzen, (1991), Object Oriented Modeling and Design, Prentice Hall, Englewood Cliffs, pp53-222.
- [12]. Samuel Lahtinen, Jari Peltonen, (2004). Adding speech recognition support to UML tools, Journal of Visual Languages & Computing, 16(2005) 85-118.
- [13]. Samuel Lahtinen and Jari Peltonen,(2003). Enhancing Usability of UML CASE-Tools with Speech Recognition.

- [14]. Shlaer.S and S.J. Mellor, (1998), Object-Oriented Systems Analysis: Modeling the World In Data, Yourdon Press: Prentice Hall, Englewood Cliffs, New Jersey, pp 53-167.
- [15]. Simon Bennett, Steve McRobb and Ray Farmer,(2004). Object Oriented Systems Analysis and Design Using UML, NY: Mc Graw Hill,pp96-100.
- [16]. Stephanie Strassel, David Miller, Kevin Walker, Christopher Cieri, (2002). Shared Resources for Robust Speech-to-Text Technology, Linguistic Data Consortium, University of Pennsylvania, ccieri@ldc.upenn.edu.
- [17]. Tatham, Mark, (2005), Developments in Speech Synthesis, Chichester: J. Wiley,pp41-151.
- [18]. Victor Zue, Ron Cole, & Wayne Ward, (2008).Speech Recognition, from: <http://cslu.cse.ogi.edu/HLTsurvey/ch1node4.html>
- [19]. Weinschenk, Susan, (2000). Designing Effective Speech Interfaces, New York : John Wiley & Sons,pp23-71.
- [20]. R. D. Peacocke and D. H. Graf. (1990).An introduction to speech and speaker recognition. From: ieeexplore.ieee.org/iel5/4483171/4483172/04483217.pdf.