

## REVIEW OF SYNTACTIC ANALYSIS SOFTWARE AUTOMATIC GENERATION

YUYAN XING AND YAO LIU\*

Institute of Scientific and Technical Information of China  
No. 15, Fuxing Road, Haidian District, Beijing 100038, P. R. China

\*Corresponding author: liuy@istic.ac.cn

Received March 2018; accepted June 2018

**ABSTRACT.** *With the popularization of computers, using a large number of software tools to carry out research has become a trend, involving numbers of techniques related to syntactic analysis. Syntactic analysis is one of the key technologies in the field of natural language processing. At present, there are lots of syntactic analysis software, and its function is becoming more and more. The artificial selection not only wastes a lot of time, but the processing efficiency is not high. So, it is difficult to find the software tool suitable for dealing with the current resources. Therefore, based on the trend of using syntactic analysis software and software automatically, this paper launches the research of automatic generation of syntactic analysis software. By combining the automatic code generation technology with the syntactic analysis software, the traditional manual processing is changed to automatic processing, which not only saves time but also effectively improves the knowledge utilization.*

**Keywords:** Syntactic analysis software, Natural language processing, Code automatic generation

**1. Introduction.** With more and more attention to the automatic generation of software, the relevant research has achieved fruitful results, and in many aspects we can see the application of automatic generation technology. According to the different methods and content, the software automatic generation can be divided into in three parts, respectively: CASE (Computer-Aided Software Engineering) [1], domain automatic generation technology [2] and automatic code generation technology. In the first three parts of this paper, CASE, domain automatic generation technology, code automatic generation technology are introduced, then compare the main technology and summarize the related tools. Finally, the paper introduces the automatic generation of syntactic analysis software and summarizes the related progress.

**2. CASE Automatic Generation Technology.** CASE is the concept of computer aided software engineering in the late 1960s, which requires the use of engineering principles and techniques to develop and maintain software, which is proposed in the early 1980s by the software engineering industry and generally accepts this terminology as a synonym for software development automation support. Therefore, the CASE can be simply understood as: CASE = software engineering + automation tools [3].

In the research project of code automatic generation based on case tool, Liu and Li [4] have proposed object-oriented CASE tools to automatically generate the most typical technical solutions. OOCASE code automatic generation generates executable code from the code generator based on user specifications. Specifically, based on the central information base, according to various transformation rules and knowledge, the reusable components and templates in the library can be used to convert the specifications of the

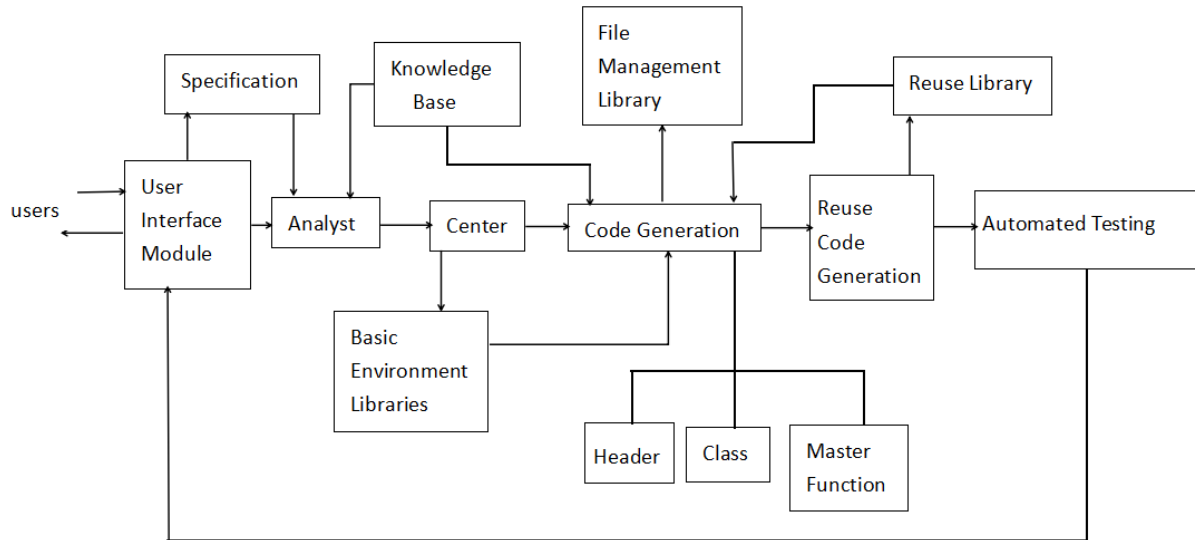


FIGURE 1. Code automatic generation system

system described in the central repository into executable code. In the process of conversion, the techniques are used as follows: lexical analysis, grammar monitoring, central information base operation, including inserting, querying, reusable parts extraction and so on. The system structure is shown in Figure 1.

**3. Domain Automatic Generation Technology.** The technology of domain automatic generation mainly includes two kinds: one is based on the domain language automatic generation technology, and this technology mainly includes the domain language, the application generator and the domain specialized library; the other is automatic generation technology based on domain modeling, which includes two parts: one part transforms the domain model into PIM; the other part converts the PIM to code automatically [2].

**3.1. Automatic generation technology based on domain language.** Zhou [5] has put forward the technology of automatic generation of application software based on domain-specific language, and carried out the research on GIS application system. The technical design framework of software automatic generation based on domain language consists of three levels: first of all, design a domain-specific language for the application domain, and then implement a domain-specific library of the application domain, on the basis of which, finally realize a set of tools, language and library integrated visual application generator [7]. The framework of the automatic generation technology is shown in Figure 2.

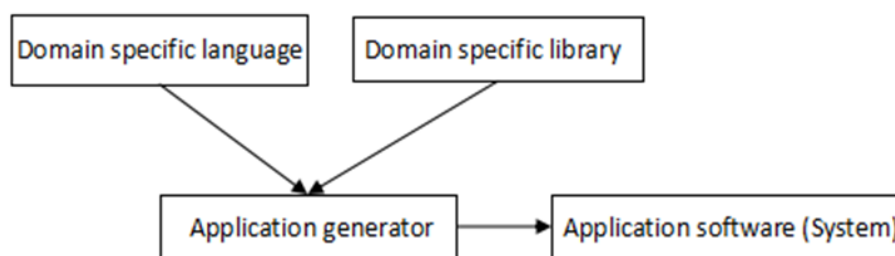


FIGURE 2. The framework of automatic generation technology

**3.2. Automatic generation technology based on domain modeling.** Gitzel and Schader [6] have proposed an automatic generation scheme of Web applications based on xml. With the help of the meta model, this scheme provides a prototype of an automatic code generation system, which is based on the three-layer meta model class; Zhou and Zheng [7] have proposed the method of code generation based on domain modeling. Model integrated computing is a domain-oriented development method [8], which supports domain model components by using domain-element model as modeling language, and developers can interpret domain models through model interpreters [9], thereby achieving rapid customization of domain applications. The code-generated schema is shown in Figure 3. The main functions are divided into two parts: one is to convert the domain model to PIM; the other is to automatically convert the generated PIM to code.

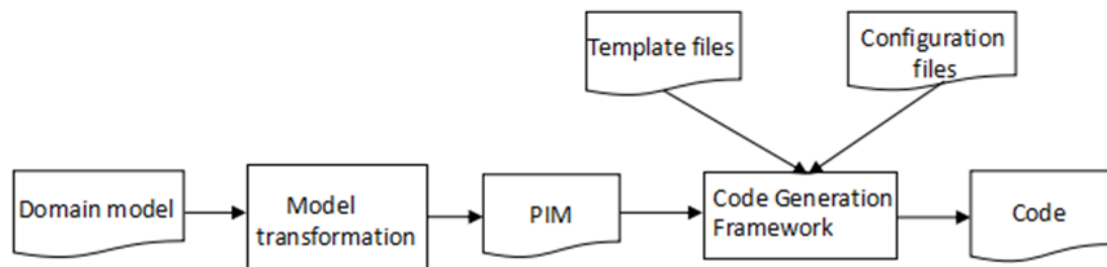


FIGURE 3. The architecture of code generation

**4. Automatic Code Generation Technology.** In recent years, in the research and implementation of automatic code generation technology, some researchers at home and abroad have carried out relevant research in different fields, and put forward corresponding code generation scheme. These schemes contain 5 methods for automatic code generation, namely, template-based code generation (including code generation based on XSLT, code generation based on template engine), code generation technology based on template, code generation technology based on DOM, code generation technology based on contract and code generation technology based on API.

**4.1. Code generation technology based on template.** Georgescu [10] has proposed a method of automatic code generation using template technology. The scheme uses XML to write the corresponding template, with some metadata files and parsing templates to realize the automatic generation of code. Kong et al. [11] have proposed code generation technique based on velocity, which combines the business requirement of rapid generation of information management system, and proposes a platform framework based on DOM4J analytic framework and velocity template technology. A complete template-based code generation process typically consists of three parts: input model, template, and generator [12]. The main process of template-based code generation technology is shown in Figure 4.

There are two main types of template-based code generation: code generation based on XSLT (eXtensible Stylesheet Language Transformation) and code generation based on template engine.

(1) Code generation technology based on XSLT

Xu and Zhang [13] have proposed code automatic generation technology scheme based on XML/XSLT, the scheme according to the selected template, does automatic validation, system code integration and generates system final target code. XSLT is the language of translating an XML (eXtensible Markup Language) document into another XML document or a plain text file [14]. It uses the XSLT transformation language and the XPath (eXtensible Markup Language Path Language) language as the template language, and uses XML for modeling and XML file to record metadata, combining the rules defined

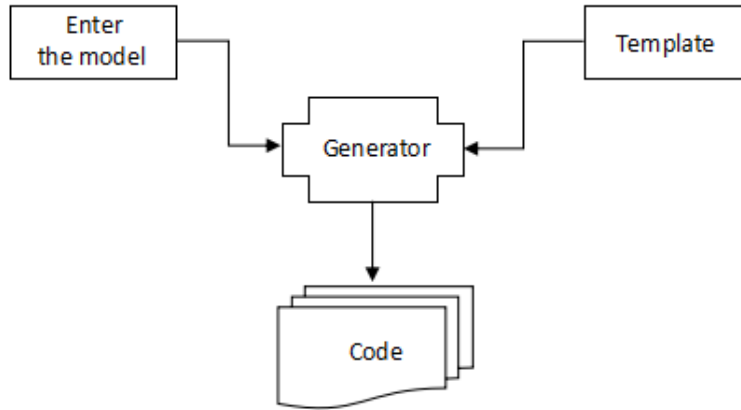


FIGURE 4. Process of code generation technology based on template

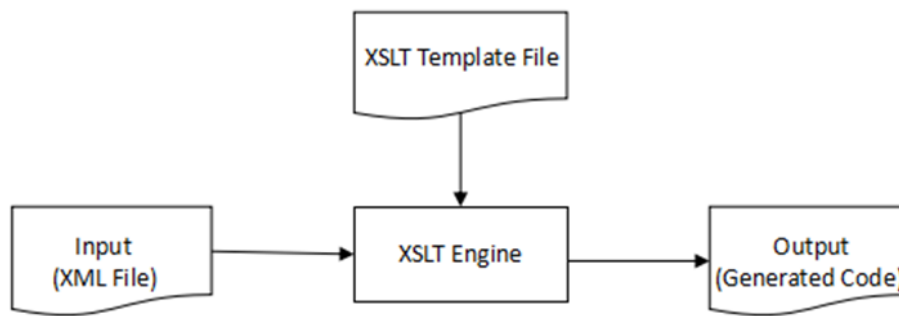


FIGURE 5. The principle of code generation

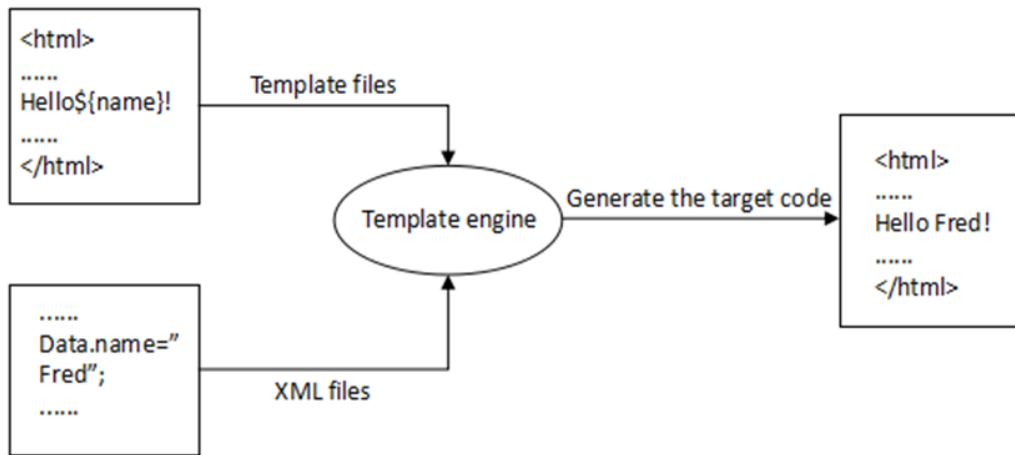


FIGURE 6. The theory of code generation technology based on template engine

by the XSLT transformation language to automatically generate the required code files through the XSLT engine [15]. It works as shown in Figure 5.

(2) Code generation technology based on templet engine

The code generation technology based on template engine parsing is one of the most widely used technologies. The principle is to generate the target code by replacing the variable part of the template file with the input XML file through the template engine [16]. The code generation principle based on the template engine is shown in Figure 6. It contains three parts: an XML file, a template file, a template engine, and the name value in the XML file you enter as Fred, and the template engine replaces the variable name value in the template file with the output Fred.

**4.2. Code generation technology based on model.** Ling [17] and Feng [18] have proposed a method of automatic code generation based on MDA, which defines model data and template files to achieve system conversion. MDA separates the model of software system into the platform independent model PIM and the specific platform model PSM, and unifies them through the transformation rules. The PIM model is responsible for transforming requirements into design in between requirements analysis and design, while the PSM model is a question of design and coding, which is designed to transform the design into code. This shows that MDA runs through the whole software development, and it plays a very important role in software development [19]. Figure 7 shows the basic principle of MDA.

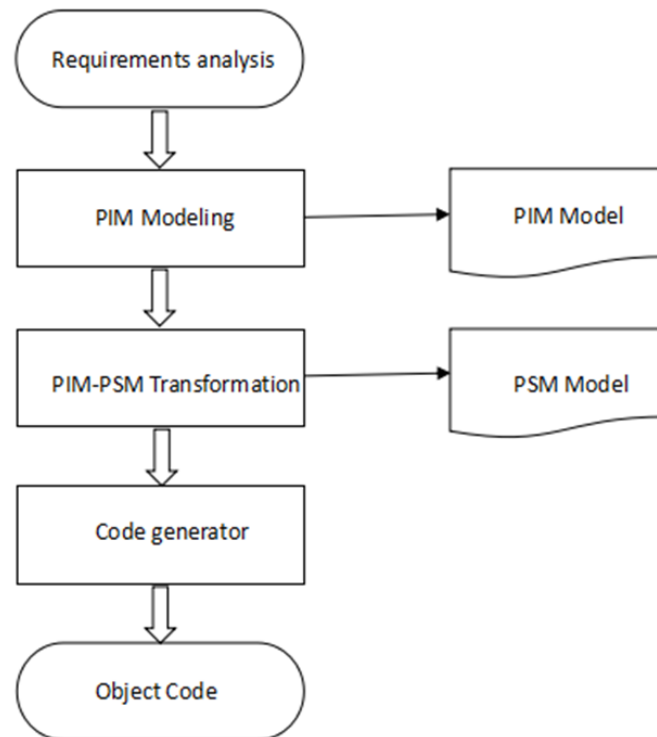


FIGURE 7. The basic principle of MDA

**4.3. Code generation technology based on DOM.** DOM is an application interface (API) for application development and programming of XML or HTML documents. As a cross-platform, language-independent interface specification published by W3C, DOM provides standard program interfaces in different environments and applications that can be implemented in any language [20]. Standard program generators that use XML input require an XML parser to convert an XML file into a DOM object. Some analysis and even conversion can take place on a DOM object, which is optional, and eventually generates a program that uses XML and DOM to build the program builder as shown in Figure 8 [21].

**4.4. Code generation technology based on API.** In addition to template-based code generation techniques, the most frequently used is code generation technology based on API. API-based code generation provides a set of APIs for generating target platform code, using these APIs to programmatically generate the code for the target platform at run time. Code generation technology based on API is based on the abstract syntax of the target language, or it can become a meta model [12]. Therefore, the method is related to the target language, which constructs the target code using the abstract syntax of the target language. Figure 9 shows the fundamentals of the API based code generation technology.

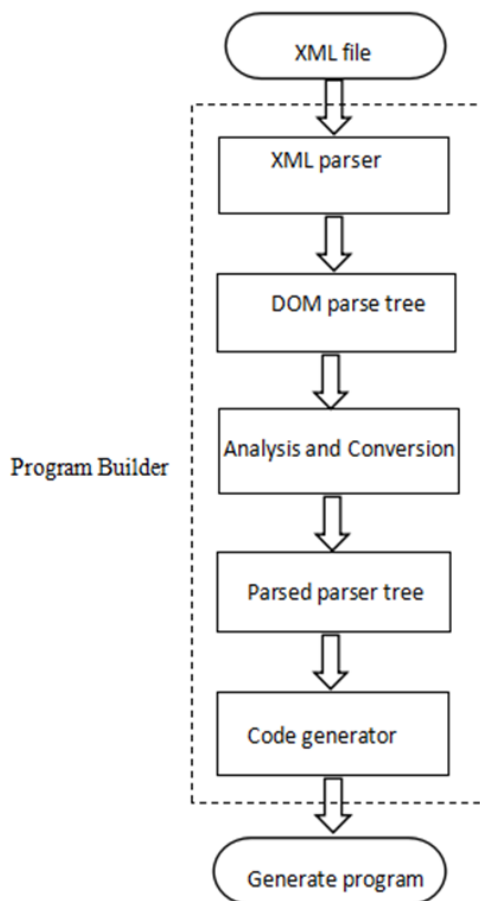


FIGURE 8. Using XML and DOM to build the program builder

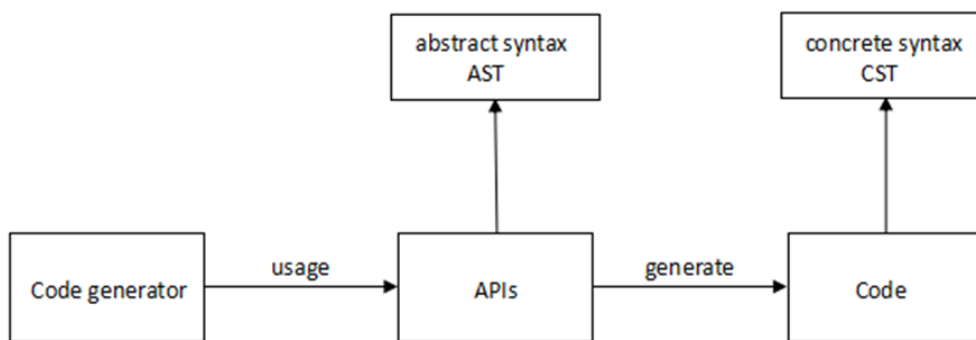


FIGURE 9. The theory of code generation technology based on API

**4.5. Code generation technology based on contract.** Design by Contract (DbC) is a universally accepted systematic approach to improving software reliability. The core idea of contract design is that a software system consists of a group of interacting components, and there is a precise definition of mutual responsibility-contract. Contracts generally include preconditions, post conditions, and invariant, which usually appear in the form of Boolean expressions. The content of the contract expresses the design intention of the program designer, and by checking the contract, it can ensure the actual behavior of the system conforms to the design purpose, which improves the reliability of the system [22].

**5. Comparison of Automatic Generation Technology.** According to the above introduction, we can find each method has its own advantages and disadvantages. Automatic generation based on templates is the most advantageous, the application domain is

the most extensive, in which the automatic code generation technology based on XSLT is easy to implement, but the shortcoming is obvious, and the template language is complex and needs to be further improved; the automatic code generation technology based on template engine is easy to reuse, we can define our own template according to the requirement, when you modify the code, just modify the template; OOCASE code generation technology is a combination of object-oriented and case tools to speed up the development of software, but so far there is no uniform standard; the automatic generation technology is based on domain language and domain modeling, although improving the development efficiency, but not high stability. At present, few people use the technology; automatic generation technology based on DOM, API, contract, the use of the domain is relatively narrow, the generated code is not readable, and it needs further research. These technologies not only save time but also effectively improve the knowledge utilization. Therefore, it is very necessary to carry out this research.

**6. Related Exploration of Syntactic Analysis Software.** Our research group has a complete idea on the automatic generation of syntactic analysis software: from some corpus, to build a model as a priori knowledge, then the corpus as data, iterative correction model, get an optimized model, and finally use the optimized model to generate software. The most important point of this idea is to use the code as a text, code as a corpus, in natural language processing methods.

According to the above ideas, the exploration steps of automatic syntactic analysis are as follows (Take fudannlp as an example).

The first step, investigate a large number of syntactic analysis software (consider only the Java programming language), such as Stanford Parser, Fudannlp, Hanlp, which meets our needs now. In the corresponding official website or gitub download software source.

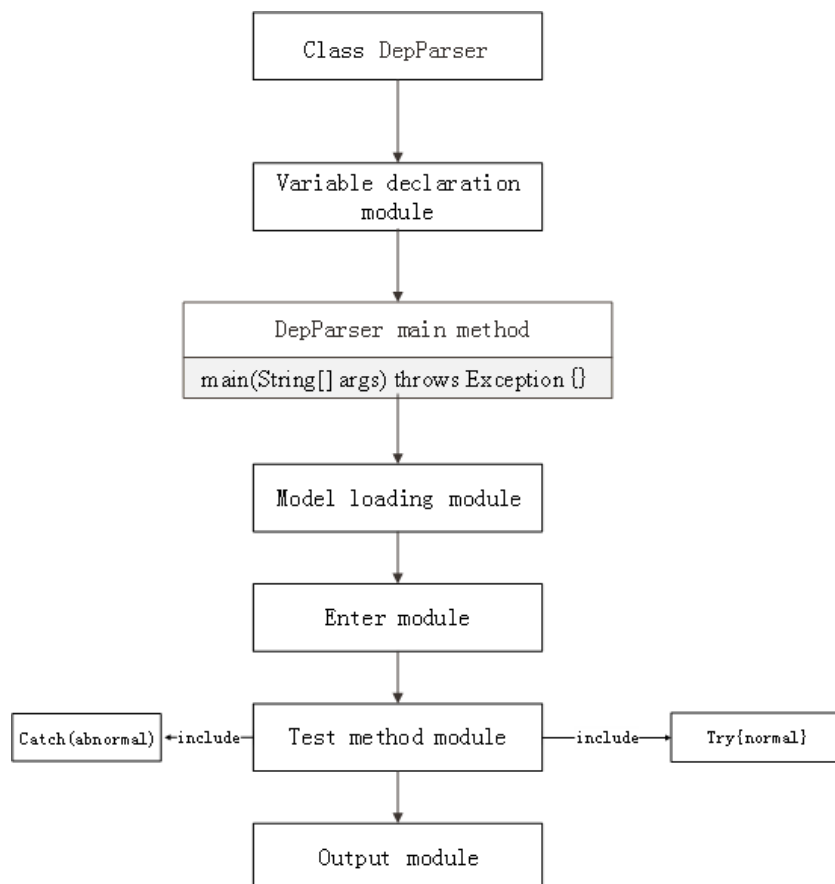


FIGURE 10. Logical structure

The second step, each software will have a corresponding demo, first in Eclipse in the demo run successfully, from the results of the operation of the general summary of the corresponding software steps, and then take the demo as the main line, to find the demo involved in the specific classes and methods, extracted from the source, to form a complete project. The logic structure of software has been extracted in this process. The logical structure is shown in Figure 10.

The third step, the project uploaded to the corpus processing platform, upload has summed up 63 keyword dictionary, with natural language processing method, the code as a corpus tag, as shown in Figure 11.

The fourth step, check the integrity of the label, if the effect is not good, but also to create a new dictionary of the software itself, repeated optimization, to get an optimization model. In the code generation technology, the generation method based on template engine is proposed, and the Freemarker tool is used to generate the code with strong practicability, and its generation process is more flexible, as shown in Figure 12.

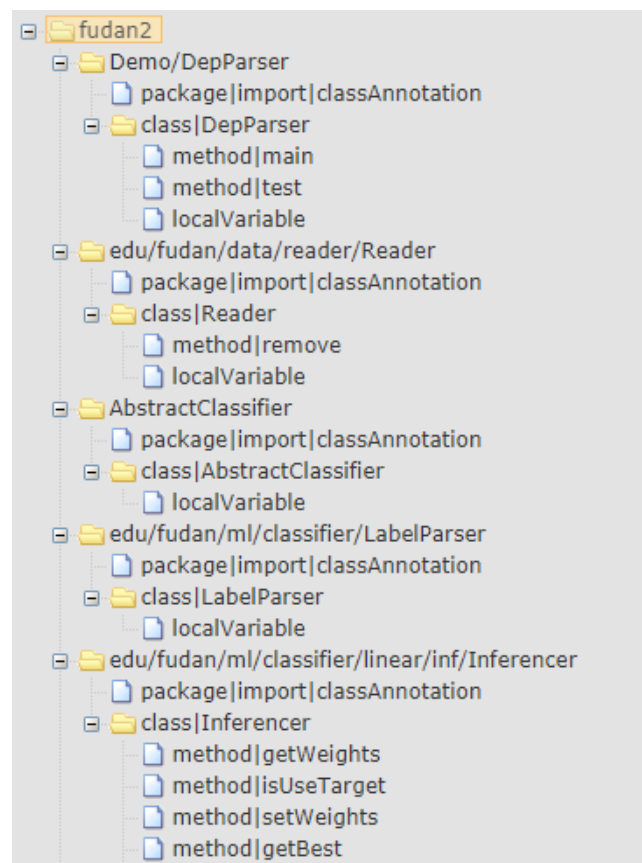


FIGURE 11. Preliminary analysis of the project

FIGURE 12. Label diagram



**7. Conclusions.** Syntactic analysis software is an indispensable tool for intelligence researchers and researchers in the field of natural language processing, and the acceleration of the intelligent age makes the status of tool automation more and more important, the demand of researchers for software automation becomes more and more obvious, and the related research and application development will be developed. Our topic group already has an idea in this direction, next we will synthesize each kind of technology, unceasingly carries on the research, and adapts the society unceasing development of the demand.

#### REFERENCES

- [1] X. Zhao, *Software Development Automation Technology*, National Defense Industry Press, 1997.
- [2] Y. Zhou, *Research on Automatic Generation of GIS Application System*, Master Thesis, Beijing University of Technology, Beijing, 2002.
- [3] Y. Tong, W. Chen, Y. Zhao and Q. Shen, CASE tool and its analysis of application, *Journal of Liaoning Institute of Technology*, no.1, pp.26-28, 2003.
- [4] M. Liu and W. Li, Automatic generation of code in object-oriented CASE tool, *Computer Science and Modernization*, vol.61, no.3, pp.22-25, 1999.
- [5] Y. Zhou, Automatic generation of application software based on domain specific language, *Computer Engineering and Applications*, no.10, pp.124-127, 2003.
- [6] R. Gitzel and M. Schader, *Generation of XML-Based Web Applications Using Metamodels*, Acta Press, Anaheim, 2003.
- [7] L. Zhou and Q. Zheng, Automatic generation of code based on domain modeling, *Bulletin of Science and Technology*, vol.27, no.2, pp.200-202, 2011.
- [8] A. Ledeczi, A. Bakay, M. Maroti et al., Composing domain specific design environments, *Computer*, vol.34, no.11, pp.44-51, 2001.
- [9] G. Edwards, N. Medvidovic et al., Modle interpreter frameworks: A foundation for the analysis of domain specific software architectures, *Journal of Universal Computer Science*, vol.14, no.8, pp.1182-1206, 2008.
- [10] C. Georgescu, Code generation templates using XML and XSL, *C/C++ Users Journal*, vol.20, no.1, pp.6-19, 2002.
- [11] D. Kong, F. Luo, W. Lin et al., The research on automatic generation technology of code based on velocity, *Computer Application and Software*, vol.31, no.10, pp.20-23, 2014.
- [12] Y. Dong, *Research and Application of Code Generation Technology in the Development of Management Information System*, Master Thesis, Jilin University, Jilin, 2012.
- [13] A. Xu and J. Zhang, Research on automatic generation of XML/XSLT code, *Journal of Hangzhou Institute of Electronic Technology*, vol.24, no.4, 2004.
- [14] Y. Zhang, Z. Chen, J. Lu and X. Zhang, Analysis and comparison of mainstream code generation methods, *Computer and Modernization*, no.11, pp.103-108, 2015.
- [15] L. Yu, Q. Yuan, J. Huo et al., Model-driven architecture based on XML and XSLT, *Computer Engineering*, vol.6, no.6, pp.49-51, 2010.
- [16] T. Guan, *Research on Template Based Automatic Code Generation Technology*, University of Electronic Science and Technology, Chengdu, 2007.
- [17] D. Ling, *Research and Implementation of Automatic Generation of Code Based on MDA*, Master Thesis, East China Normal University, Shanghai, 2006.
- [18] Y. Feng, *Research and Application of Code Generation Technology Based on MDA*, Master Thesis, Xi'an University of Electronic Science and Technology, Xi'an, 2011.
- [19] R. M'hamed and M. Samir, MDA-based ATL transformation to generate MVC 2 web models, *Internationnal Journal of Computer Science & International Technology*, vol.3, no.4, pp.57-70, 2011.
- [20] B. He, The document object model (DOM) and its application, *Modern Commercial Industry*, vol.11, pp.291-293, 2009.
- [21] J. C. Cleaveland, *Using XML and JAVA to Create Program Generator*, Science Press, Beijing, 2002.
- [22] C. Zhang, Research and implementation of AOP code generators for supporting contracts, *Micro-computer Applications*, vol.23, no.1, pp.1-4, 2007.