International Journal of Knowledge
and Language Processing
Volume 9, Number 1, **2018**

www.ijklp.org
KLP International ⓒ2018 ISSN 2191-2734
pp.36—56

Research on key technologies of automatic generation of syntactic parser

Xing Yuyan
Institute of Scientific and Technical Information of China
No.15 Fuxing Road, Haidian District, 10038
Beijing, China
xingyy@istic.ac.cn

Received March 2018; revised March 2018

Abstract. *With the popularization of computers, using a large number of software tools to carry out research has become a trend, involving a large number 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 a lot of syntactic analysis software, its function is become more and more .The artificial selection not only waste a lot of time, 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, Automatic Generation

1. **Introduction.** With the more and more attention to the automatic generation of software, the relevant research has achieved fruitful results, in many aspects 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, 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 compares the main technology and summarizes 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 accepted this terminology, and 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, Liumanyun[4] and others proposed object-oriented CASE tools 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 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.

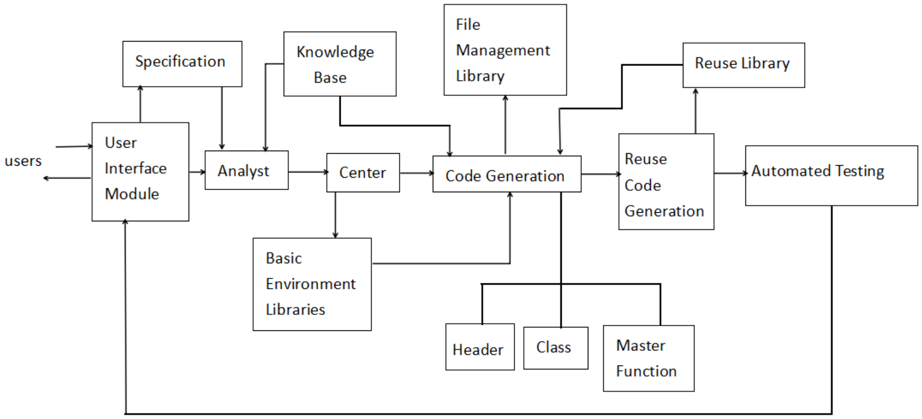


FIGURE 1. CODE AUTOMATIC GENERATION SYSTEM

(1) Human-machine interface module: This module belongs to the software analysis, design stage.

(2) Analysis Program: Responsible for checking the software specifications of the syntax, semantic correctness. The examination of grammatical correctness is similar to the grammar check of compiler, and can be analyzed by recursive subroutine.

(3) Center Environment Builder: Create basic Environment Library according to specification, i.e. class table, inheritance relation table, etc.

(4) Code Generation Program: Basic Environment Database tables provide a description of the entire software system, including static parts, such as class description, inheritance relationship description, dynamic part, such as class instance description, event occurrence description. Code generators convert them to programs, and you will find relevant

information at any time during the build process. The File management library stores the generated files and documents, and as code continues to occur, the file management library inserts the resulting code into the appropriate location. The Reuse library provides reusable artifacts that can be used for the system, and the Code generation program finds the extraction as needed. The Knowledge Base provides the basic rules for code generation and the knowledge that applies to the specialized domain. The code generation program looks for the rules to guide the generation work to ensure the correctness of the program. The code generation program can be divided into three parts, the header file generates a header file for generating language programs, including global variables, constants, and so on. Class to generate all of the custom classes used to build the system. The main function generation part generates the main () function, which completes the system through a class method call.

(5) Reuse Component generators: Extract the reusable part of the generated code, fill in the reuse library, including various management information, such as version, author, meaning, etc.

(6) Automatic test: Automatically generate test cases, complete the system test, if there is a mistake, re-enter human-computer interaction, the user changes the specification.

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, this technology mainly includes the domain language, the application generator and the domain specialized library three main elements. Domain specialized language (DSL) is one kind uses in the specific domain or the question computer programming language, it provides this domain suitable, fixed abstract concepts and symbols. Application generator is an application that can produce a series of similar applications in a certain field. The application generator itself encapsulates a reusable, relatively robust design and is a planned reuse. A domain-specific library is a library of functions used to provide reuse for applications in a particular domain, class library or dynamic link library, which provides component and product software reuse for application generators and secondary developers.

Another is automatic generation technology based on domain modeling, which includes two parts, one part transforms the domain model into PIM, and the other part converts the PIM to code automatically. The main idea is to set up a modeling language or rule for a particular field of application. Due to the certainty of domain model application domain, makes it possible to generate automatic code based on domain models[5].

3.1. **Automatic generation technology based on domain language.** Zhou Yanming[6] and others put forward the technology of automatic generation of application software based on domain-specific language, and carry 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 8. The framework of the automatic generation technology is shown in FIGURE 2.

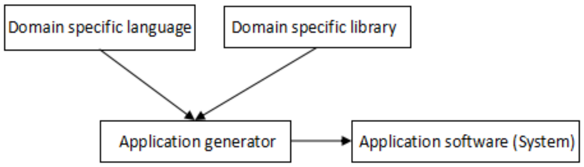


FIGURE 2. THE FRAMEWORK OF AUTOMATIC GENERATION TECHNOLOGY

If the design framework is to be implemented in its entirety, the above three parts must be implemented separately. Using the application generator, the development process of implementing a complete application software (after called Application System) is divided into two steps: The first step, the software designer uses the application generator, generates the application system framework; The second step, secondary developers to complement and improve the application system framework, resulting in application system6.

3.2. **Automatic generation technology based on domain modeling.** Ralf[7] and others 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 li[8] and others proposed the method of code generation based on domain modeling. Model integrated Computing is a domain-oriented development method[9], which supports domain model components by using domain-element model as modeling language, and developers can interpret domain models through model interpreters[10], thereby achieves rapid customization of domain applications. Domain model is a domain element model described by the domain element model, which is a set of domain experts and developers in the abstract of the modeling rules, target code that needs to be generated through the domain model also has certain syntax rules, so the automatic transformation from the domain model to the code is fully possible[11].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 , another is to automatically convert the generated PIM to code.

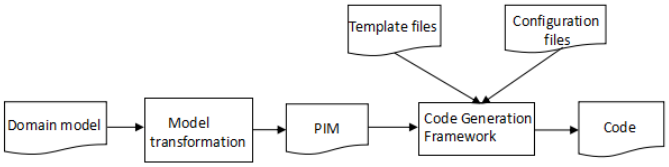


FIGURE 3. THE ARCHITECTURE OF CODE GENERATION

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[12] proposes 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 deyu[13] and others proposes 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. The input model is the raw material that generates the code, usually from the architectural information. Templates are defined by the relevant user, and the definition of a template takes into account information about the input model, so the template is usually related to the input model, and a class of input models can share a template. The main role of the builder is to parse the input model and the information in the template, combining the two to generate new code[14]. The main process of template-based code generation technology is shown in FIGURE 4.

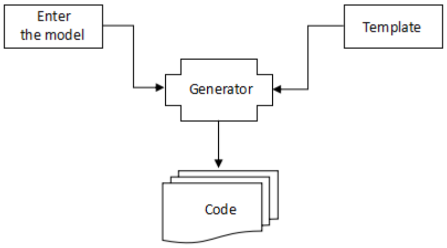


FIGURE 4. PROCESS OF CODE GENERATION TECHNOLOGY BASED ON TEMPLATE

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 aichun[15] and other people proposes code automatic Generation technology scheme based on XML/XSLT , the scheme according to the selected template, automatic validation, system code integration and generate 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[16]. It uses the XSLT transformation language and the XPath (Extensible Markup Language Path Language) language as the template language, uses XML for modeling and XML file record metadata, combining the rules defined by the XSLT transformation language to automatically generate the required code files through the XSLT engine[17]. It works as shown in FIGURE 5.

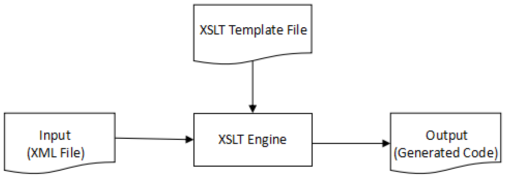


FIGURE 5. THE PRINCIPLE OF CODE GENERATION

(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[18]. The code generation principle based on the template engine, as shown in FIGURE 6. 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.

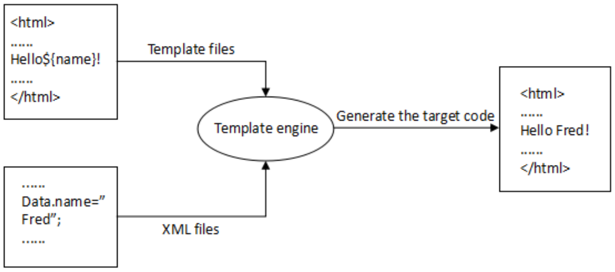


FIGURE 6. THE THEORY OF CODE GENERATION TECHNOLOGY BASED ON TEMPLATE ENGINE

4.2. **Code generation technology based on model.** Model-based automatic code generation technology is the software system to be abstracted as a model, through the model to the code mapping rules automatically generate code. Among them, the establishment model belongs to the software development design stage, the code generation belongs to the coding stage, thus completes the design and realizes the integration seamless integration [19].

Ling huade[20], Feng yingying[21] proposes 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, it plays a very important role in software development[22].FIGURE 7 is the basic principle of MDA.

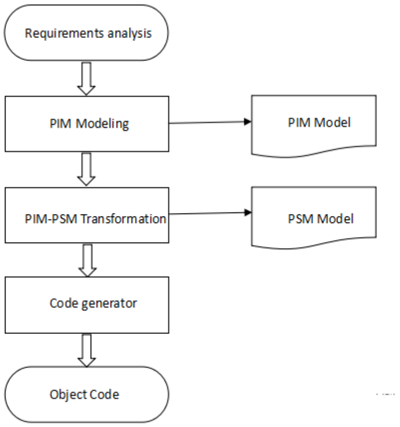


FIGURE 7. THE BASIC PRINCIPLE OF MDA

In addition, Frohner[23] and others proposes an automatic code generation scheme based on UML model. Based on the analysis of multi-layer structure system, the automatic generation of each layer code is realized, and the relationship between the layers is preserved, and the multiple paradigm method provided by this scheme enables the developer to choose the appropriate solution for its implementation. Wang xiaoyu[24] and others proposes a new code generation technology scheme based on UML and class diagrams, which is a method of combining class diagram and sequence diagram to generate C++ code including static structure and dynamic behavior information. Mathupayas[25] and other people put forward UML to Java Code conversion method. UML is a standard modeling language for analyzing and designing the system graphically, and UML modeling can clearly represent the structure and behavior information of the system. There are a lot of code studies about translating UML model diagrams into code, and document lists the rules for converting multiple sequential graphs into Java code. FIGURE 8 illustrates the process framework for UML model conversion to Java.

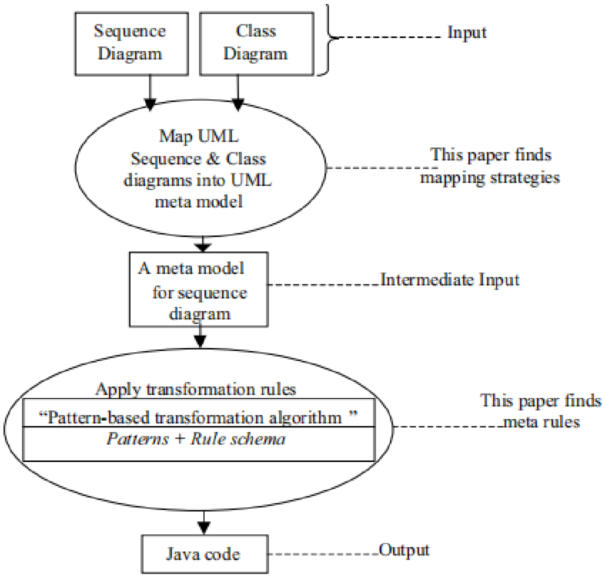

Sequence Diagram
Class Diagram
Input
Map UML Sequence & Class diagrams into UML meta model
This paper finds mapping strategies
A meta model for sequence diagram
Intermediate Input
Apply transformation rules
"Pattern-based transformation algorithm "
Patterns + Rule schema
This paper finds meta rules
Java code
Output

FIGURE 8. THE TRANSFORMATION METHOD OF UML TO JAVA CODE [25]

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. With DOM, developers can dynamically create XML or HTML documents, traverse structures, add, modify, delete content, and so on. Its object-oriented features, so that people in the processing of XML or HTML parsing related transactions save a lot of energy, which is a code to reuse the idea of a powerful programming tool[26].

The DOM parser maps an XML document to a tree-like node hierarchy in the form of a document object model, as shown in FIGURE 9[27].

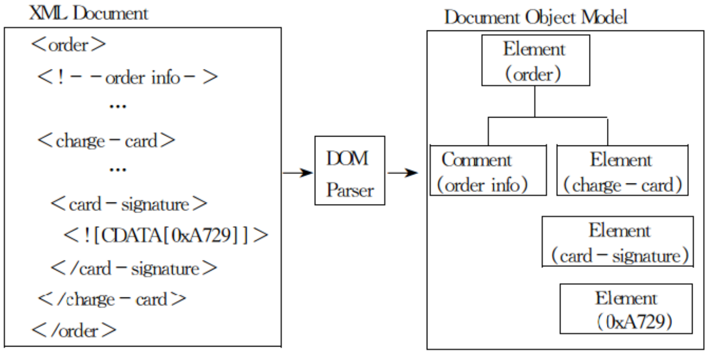


FIGURE 9. DOCUMENT OBJECT MODEL 27

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 10[28].

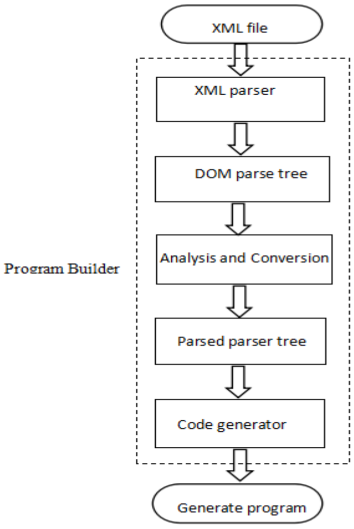


FIGURE 10. USES XML AND DOM TO BUILD THE PROGRAM BUILDER

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[29]. Therefore, the method is related to the target language, which constructs the target code using the abstract syntax of the target language. Code generation technology based on API is intuitive and easy to use, and can guarantee the syntax correctness of the generated code, because incorrect syntax cannot be compiled by the compiler at all. However, the biggest problem with this technique is that it requires programming to implement the generation of fixed code. When the generated code is large, it is easy to create confusion for programmers because of the lack of intuitive perception of the generated code. Also, since this approach is for abstract syntax, if an abstract syntax has multiple languages corresponding to multiple specific syntaxes, you can select the appropriate builder to generate code in the specified language[34]. FIGURE 11 shows the fundamentals of the API based code generation technology.

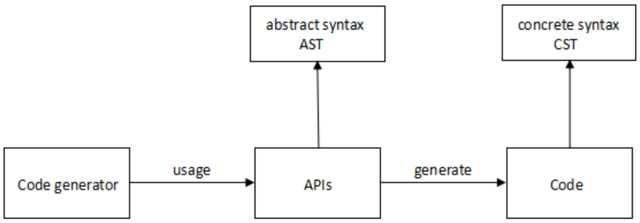


FIGURE 11. THE THEORY OF CODE GENERATION TECHNOLOGY BASED ON API

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. Both the precondition and the post condition are for the method: Object A must ensure that the precondition of method M is satisfied before calling a method M of object B, and that the method cannot be invoked, and after the method call, object B must guarantee that the method M's post condition is satisfied, or the method call will fail. Invariant for the entire class, it represents the class has no change of characteristics, so before and after the method is performed need to check the invariant. 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. Aspect-oriented programming is a new programming method, whose foothold lies in the separation of attention points. In the concept of AOP, a software system consists of several concerns, some of which are related to the business logic of the system, known as the "primary focus", while others are commonly found in various components of the system, called crosscutting concerns. Based on the basic objectives defined by its specifications, AOP first separates crosscutting concerns from primary concerns, and then maintains individual crosscutting concerns in a particular component called "aspect". After the main focus code has been completed (typically using object-oriented methods), each aspect will be woven into the specified location of the program, and the resulting program will provide complete functionality[30].

AspectJ is an extension of the Java language that implements the AOP language specification at the language level, and its primary language elements include[30]:

●Join point: A connection point is a well-defined location in the main code (standard Java code), which can be a method invocation, exception handling, and so on.

●Pointcut: pointcuts use the Pointcut keyword declaration, which is used to determine the matching time of a connection point and get the context of a connection point.

●Advice: Advice is used to determine the code to be executed at the specified pointcut and to specify how the code is executed, either before, after, or in place of the pointcut execution.

●Aspect:The aspect uses the Aspect declaration, which encapsulates all the code that implements crosscutting concerns.

In addition, ASPECTJ also supports attributes such as declaration between types, and provides a specialized compiler, the main work of the ASPECTJ compiler is to complete the aspect of weaving.

5. **Comparison of automatic generation technology.** The following is a comparison of the pros and cons of existing technologies, as shown in TABLE 1.

TABLE 1. THE COMPARISON TABLE OF AUTOMATIC GENERATION TECHNOLOGY

Automatic generation technology	Pros	Cons
OOCASE	Increased efficiency; reduced complexity and cost of software maintenance; high quality and high reliability applications; more responsive to customer needs	Compatibility is poor; different tool model format is different; there is no unified standard.
based on domain language	Developers can modify the system efficiently; reduce development time ; improve development efficiency.	lack of flexibility;
based on domain modeling	Developers can modify or recreate the system efficiently; improve development efficiency.	Small application scope ; lack of stability.

based on templet	Code style is uniform, easy to reuse; code is efficient, good templates can greatly improve the efficiency of the code; modify the code only need to modify the template.	Relies on a particular technology platform and programming language.
based on model	The model is simple and intuitive, it is easy to depict the software system; high degree of project integration.	Generate more code redundancy; low operational efficiency of code
based on DOM	Object-oriented features that enable people to save a lot of effort in dealing with XML or HTML parsing related transactions.	The generated program is difficult to read; used infrequently now.
based on API	Intuitive and easy to use; can guarantee the correctness of code syntax.	Programming is required to fix code generation, and when the scale becomes larger, make programmers are confused.
based on contract	The generated code and language are more consistent and can be developed two times.	Can only generate interfaces, can not generate running code; poor stability.

As we can see from table 1, 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, 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 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, need further research. When we select code generation technology, we need to evaluate the requirements and consider all the technologies, and choose the most appropriate code generation scheme.

6. **Related tools of auto generated.** In the process of automatic generation, a large number of tools are involved, including case tools, code generation tools, template engine tools. The basic function of the code generation tool is to read the metadata and generate the corresponding source code according to certain patterns and rules. The three elements of a typical code generation tool are design patterns, domain metadata, and domain rules, as shown in FIGURE 12. The design pattern is the template file that produces the code; Domain metadata is a topology that needs to be modeled in code, typically with the growth of specific data provided by the developer; The domain rule specifies the rule for the domain metadata structure and behavior, which is usually encapsulated in the builder program[31].

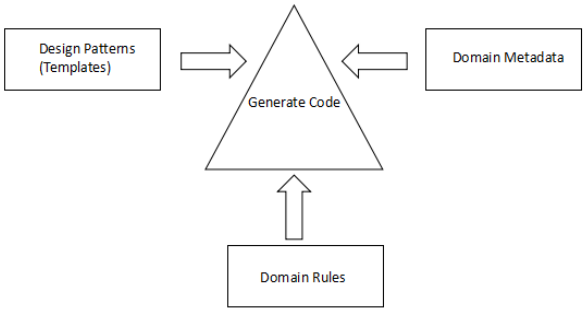


FIGURE 12. TYPICAL CODE GENERATION TOOLS[32]

6.1. **CASE tools.** At present the domestic and foreign market popular case tool mainly concentrates in the system design and the analysis, the system modelling, the database modelling, the frame code generation and so on, the common case tool is shown in TABLE 2.

TABLE 2. COMMON CASE TOOLS[33]

Case name	range of application
PlayCASE	System modeling,documents automatically generated
PowerDesigner	System modeling,database modeling and design,data dictionary generation,database generation,framework code generation
BPWin	System modeling
ERWin	Database modeling and design,data dictionary
ModleMart	Generation,database generation Team development and management tools combined with ERWin
Rational Rose	System Analysis,Design;framework code generation
SoDA	System document generation combined with Rose

Together	System analysis, design;framework code generation
With Class	System modeling,framework code generation
Stp/SE	System analysis,design and code framework generation,document generation
ObjectMaker	System analysis,design
Stp/T	Software test
SQA	Software Testing - Black Box Testing
VxWorks	General language embedded systems integrated development environment

The application scope of case tool mainly concentrates on simple programmer tool, analysis design workbench and integrated case aspect. In the case of integration, the IBM Rational Rose series in the United States, occupies a very large market share, and rational provides tools for architecture, design modeling, building, model-driven development, rapid application Development (RAD), component testing, and run-time analysis activities. These tools help maximize developer productivity when building business applications, software products and systems, and embedded systems and devices. In terms of case research and development, China started in the 80's, has formed a certain degree of research and development, with Professor Yang fuqing as the representative of the Peking University computer Science and Technology department in 65, 75, 85 of the issue of case theory, technology and products conducted a concentrated study, Launched the commercialization of the JBⅠand JBⅡCASE platform, in the national "95" program based on heterogeneous platforms, access to multiple information sources of application Systems integrated Environment BluebirdⅡ(JB3) system research and development. JBⅠCASE products, with workstations, microcomputers and laptops three versions, to support the various stages of the software development process, regardless of methodology, technology and supported goals have reached the international advanced level[34].

6.2. **Code generator.** At present, many large software companies at home and abroad have designed their own solutions to solve the problem of code reuse, there are some companies in order to solve the problem of commonality to develop the corresponding code generation tools, such as Rational Rose[35], it is a better model conversion tool, It can design and model some requirements for project development, and generate some code automatically according to the design model, but the tool can only generate some skeleton code, and the operation code of the specific business logic is not realized.

In addition, there are some specific areas of code automation generators, such as the PowerDesigner[36] software proposed by Sybase, which is designed to design a variety of data models in the system, can realize the concept model, physical model and other models of the conversion between the model, and generate database script files; The Visual Studio[37] tool developed by Microsoft, which only needs to drag and drop the corresponding controls, can automatically generate some class or framework code, so it is very convenient for programmers to develop projects.

In addition, there are a number of tools to automate the overall system generation, such as SmartCode, Codesmith, and Codematic. SmartCode[38] is completely based on template code generation, and it can set build properties for each table or field by reading the database schema. Codesmith[39] is a template-based code generation tool that uses a syntax similar to ASP.NET to generate any type of code or text. Dongruan.NET[40] code generator (codematic), which is designed to automatically generate a three-tier web system based on the .NET platform, which can add, delete, change, and check logical business functions and generate database-related code files.

6.3. **Templet engine tools.** Common template engine tools are Freemarker, Velocity, JET. Freemarker and Velocity are java-based template engine tools, JET is an EMF based template engine tool.

①Freemarker[41]

Freemarker is a template engine that is a general-purpose tool for generating text output based on templates. It is a development package provided by Java programs, or a class library, it is not a Web application framework, it can only be a part of the Web application framework. Freemarker is not related to web containers, that is, when the web is running, it does not know about HTTP or servlets. Freemarker is primarily applied to the generation of HTML Web pages, especially in applications that follow MVC (Model-View-Controller) mode. The design idea of Dynamic Web page based on MVC pattern separates the foreground designer (such as the HTML Designer) from the backstage coder, reduces the coupling and improves the reusability. Because of the separation of Business logic (Java program) and view (freemarker template), interface designers can change the foreground display of a page without recompiling the code, that is, the interface generation is not affected by complex program code. This separation is very meaningful, for some HTML page designers and coders are different people or even the same person, it can keep the application clear, and easy to maintain. Although Freemarker has certain programming capabilities, the Java program usually prepares the data to be displayed, and then Freemarker the template to generate a display page showing the data.

②Velocity[21]

Velocity is also a Java based template engine technology, which is an open-source project proposed by Apache. When velocity is introduced into Web project development, interface designers and Java developers can develop synchronously with a Web site with an MVC architecture. Interface designers only to the display of the page to do processing, design a good appearance; Java developers focus only on writing efficient and concise code to achieve specific business logic. So velocity separates the Java code from the presentation Layer page and makes it easier to maintain the Web site. The template language it uses is the VTL (Velocity Template Language), which makes it easy to create or modify the target code template file to generate code that meets the requirements as needed. Similarly, it is not only applied to the web domain, it can also generate other types of code files based on custom template files, such as database files, script files, other source code, or some documents.

③JET[42]

JET (Java Emitter Template) is an open source tool for generating code in the EMF (Eclipse Modeling Framework). It is a generic template engine that is included in the EMF run plug-in. The syntax for JET and JSP is very similar, are compiled into Java programs in the background, both to separate foreground presentations from data models and controllers, and to use Java objects as input parameters and both to represent the structure of the generated objects (Web pages, Java classes, or files). The difference is that JET can support different languages to generate various types of files, such as database files, XML, Java code, through the structure of the transform tag. The basic model of the JET engine is to combine the parameter file (sample.xml) with the template file (xxx.jet) to generate the corresponding target file. In order to differentiate the markup, the tags "<%" and "%>" in the JET template need to be converted to " %" and "% " when the target generation file is the JSP file. The generic Jet template file has the suffix name of .Jet, in order to enable users to clearly distinguish between generated target codes, you can use the extension to add the generated file class (such as. textjet) before jet.

7. **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.

(1)According to the above ideas, the exploration steps of automatic syntactic analysis are as follows:

The first step, investigate a large number of syntactic analysis software (consider only the Java programming language), such as Stanford Parser, Fudannlp, Hanlp, Berkeley parser, which meets our needs now. In the corresponding official website or gitub download software source. Standford parse[43] developed a CVG (Compositional Vector Grammar) method, combined with PCFG (Probabilistic Context Free Grammar) and a recursive neural network for learning grammatical semantics and combinatorial vector expressions to parse and support multiple languages; Berkeley parser[44], a syntactic parser based on probabilistic context-free grammar (PCFG), developed at the University of California, Berkeley NLP Laboratory, supports a wide range of languages such as English, Chinese, and German, and is widely used in the papers of mainstream international conferences such as ACL,NAACL; Fudannlp[45] is a toolkit developed for Chinese natural language processing at Fudan University, and also includes machine learning algorithms and datasets for implementing these tasks; Hanlp[46] is a Java toolkit consisting of a series of models and algorithms that aims to popularize the application of natural language processing in production environments. Not only participle, but provides lexical analysis, syntactic analysis, semantic understanding and other complete functions.

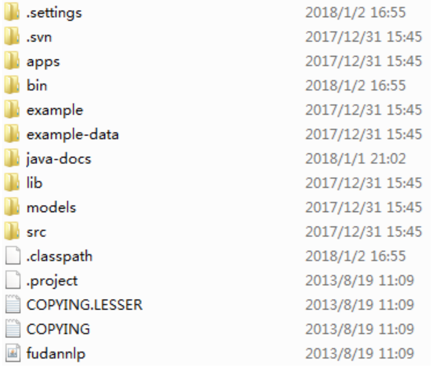
The second step, each software will have a corresponding demo, first in Eclipse in the demo run successfully, from the results of the operation of the general summary of the corresponding software steps, and then take the demo as the main line, to find the demo involved in the specific classes and methods, extracted from the source, to form a complete project The logic structure of software has been extracted in this process. During the operation of the whole Standford parser, it can be divided into main method entrance module, model loading module, judgment module, word segmentation module and syntactic analysis module, and there are many classes and methods in each module. In the Fudannlp running process, can be divided into private static variable declaration module, the main method entrance module, model loading module, input module, method module and output module are included in the main method loading module, calling method module, input module, traversing module and output module during Hanlp operation.

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.

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 Freemaker tool is used to generate the code with strong practicability, and its generation process is more flexible.

(2) Here's an example of Fudannlp, showing the steps above:

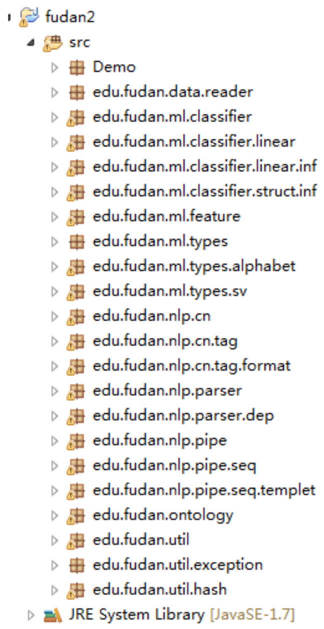
The first step, the site download the source code, as shown in FIGURE 13.



.settings	2018/1/2 16:55
.svn	2017/12/31 15:45
apps	2017/12/31 15:45
bin	2018/1/2 16:55
example	2017/12/31 15:45
example-data	2017/12/31 15:45
java-docs	2018/1/1 21:02
lib	2017/12/31 15:45
models	2017/12/31 15:45
src	2017/12/31 15:45
.classpath	2018/1/2 16:55
.project	2013/8/19 11:09
COPYING.LESSER	2013/8/19 11:09
COPYING	2013/8/19 11:09
fudannlp	2013/8/19 11:09

FIGURE 13. SOFTWARE STRUCTURE OF FUDANNLP

The second step, set up a good logical structure of the complete project and software operation called fudan2. As shown in FIGURE 14 and 15.



```
fudan2
  src
    Demo
    edu.fudan.data.reader
    edu.fudan.ml.classifier
    edu.fudan.ml.classifier.linear
    edu.fudan.ml.classifier.linear.inf
    edu.fudan.ml.classifier.struct.inf
    edu.fudan.ml.feature
    edu.fudan.ml.types
    edu.fudan.ml.types.alphabet
    edu.fudan.ml.types.sv
    edu.fudan.nlp.cn
    edu.fudan.nlp.cn.tag
    edu.fudan.nlp.cn.tag.format
    edu.fudan.nlp.parser
    edu.fudan.nlp.parser.dep
    edu.fudan.nlp.pipe
    edu.fudan.nlp.pipe.seq
    edu.fudan.nlp.pipe.seq.templet
    edu.fudan.ontology
    edu.fudan.util
    edu.fudan.util.exception
    edu.fudan.util.hash
  JRE System Library [JavaSE-1.7]
```

FIGURE 14. COMPLETE PROJECT OF FUDAN2

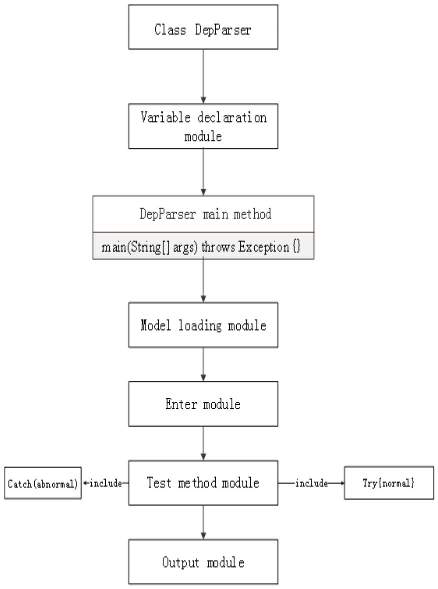
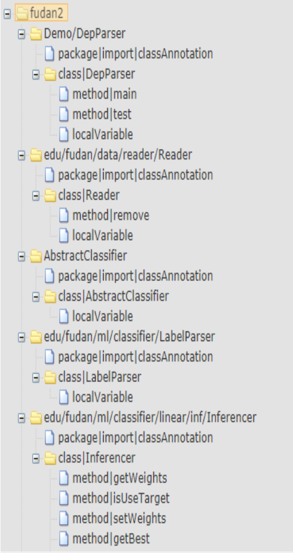


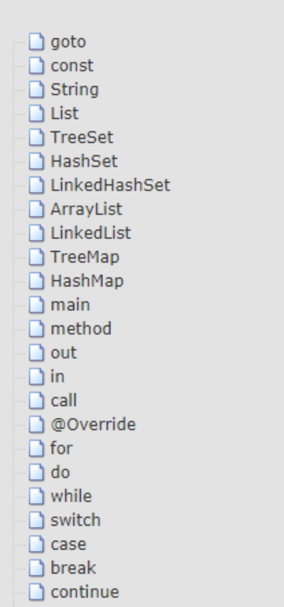
FIGURE 15. LOGICAL STRUCTURE

The third step, upload the projects and dictionaries to the platform, as shown in FIGURE 16 and 17.



```
fudan2
  Demo/DepParser
    package|import|classAnnotation
    class|DepParser
      method|main
      method|test
      localVariable
  edu/fudan/data/reader/Reader
    package|import|classAnnotation
    class|Reader
      method|remove
      localVariable
  AbstractClassifier
    package|import|classAnnotation
    class|AbstractClassifier
      localVariable
  edu/fudan/ml/classifier/LabelParser
    package|import|classAnnotation
    class|LabelParser
      localVariable
  edu/fudan/ml/classifier/linear/inf/Inferencer
    package|import|classAnnotation
    class|Inferencer
      method|getWeights
      method|isUseTarget
      method|setWeights
      method|getBest
```

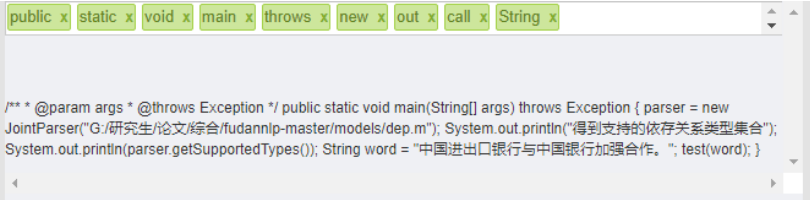
FIGURE 16. PRELIMINARY ANALYSIS OF THE PROJECT



```
goto
const
String
List
TreeSet
HashSet
LinkedHashSet
ArrayList
LinkedList
TreeMap
HashMap
main
method
out
in
call
@Override
for
do
while
switch
case
break
continue
```

FIGURE 17. DICTIONARY

The fourth step, label the lines of code and optimize them to get the optimized model, as shown in FIGURE 18.



```
/** * @param args * @throws Exception */ public static void main(String[] args) throws Exception { parser = new
JointParser("G:/研究生/论文/综合/fudannlp-master/models/dep.m"); System.out.println("得到支持的依存关系类型集合");
System.out.println(parser.getSupportedTypes()); String word = "中国进出口银行与中国银行加强合作。"; test(word); }
```

FIGURE 18. LABEL DIAGRAM

8. **Conclusion and Expectation.** This paper introduces the existing code generation technology, the use of DOM generators and code generation based API is not always used; the technology of automatic generation based on contract is immature, but the stability is poor, and the automatic generation technology based on model has a wide application prospect, which is not in line with our needs. The code generation technology based on the template engine parsing is most in line with our requirements, because this method produces a unified code style, easy to reuse, code to run high efficiency, modify the code only need to modify the template. Although code generation technology has achieved some success, in many ways it wants to improve and improve.

Syntactic analysis software is an indispensable tool for intelligence researchers and researchers in the field of natural science, 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 has already had the direction in this direction, next we will synthesize each kind of technology, unceasingly carries on the research, and adapts the society unceasing development the demand.

REFERENCES

[1] Zhao Xiaohua. Software Development Automation Technology M. National Defense Industry Press, 1997.

[2] Zhou Yanming. Research on Automatic Generation of GIS Application System D.Beijing: Beijing University of Technology, 2002.

[3] Tong yujun, Chen wenshi, zhao yue, shen qingsong. CASE tool and its application analysis J. Liaoning institute of technology, 2003(1):26-28.

[4] Liu Manyun, Li Wanxue. Automatic Generation of Code in Object-Oriented CASE Tool J. Computer Science and Modernization, 1999, 61 (3).

[5] Zhou Yanming. Research on Automatic Generation of GIS Application System D.Beijing: Beijing University of Technology, 2002.

[6] Zhou Yanming.Automatic Generation of Application Software Based on Domain Specific Language J. Computer Engineering and Applications, 2003 (10):124-127.

[7] Ralf Gitzel, Martin Schader. Generation of XML-based Web Applications Using MetamodelsM. Anaheim: Acta Press, 2003.

[8] Zhou li, zheng qiuhua. Automatic generation of code based on domain modeling J. Technology bulletin, 2011, 27(2):200-202.

[9] Ledeczi A,Bakay A,Maroti M,et al. Composing Domain specific Design EnvironmentsJ. Computer, 2001, 34(11):44-51,24.

[10] Edwards G, Medvidovic N,et al. Modle Interpreter Frameworks:A Foundation for the Analysis of Domain Specific Software ArchitecturesJ. Journal of Universal Computer Science, 2008, 14(8):1182-1206.

[11] Zhou li, zheng qiuhua. Automatic generation of code based on domain modeling J. Technology bulletin,

2011, 27(2):200-202.

[12] Georgescu C. Code generation templates using XML and XSLJ. C/C++ Users Journal, 2002, 20(1): 6-19.

[13] Kong deyu, Luo feng, Lin weibo, et al. The research on automatic generation technology of code based on Velocity J. Computer application and software, 2014, 31(10):20-23.

[14] Dong yu-ming. Research and application of code generation technology in the development of management information system D. Jilin: jilin university, 2012:11.

[15] Xu aichun, zhang jianmin. Research on automatic generation of XML/XSLT code J. Journal of Hangzhou institute of electronic technology, 2004, 24(4).

[16] Zhang yanwei, Chen zhengming, lu jia, zhang xia. Analysis and comparison of mainstream code generation methods J. Computer and modernization, 2015(11):103-108.

[17] Yu liangwei, yuan quan, huo jianqing, etc. Model-driven architecture based on XML and XSLT J. Computer engineering, 2010, 6(6):49-51.

[18] Guan taiyang. Research on template based automatic code generation technology D. Chengdu: university of electronic science and technology, 2007.

[19] Guo dongyang. Template-based UML Statechart Code Framework Generation D. Xian: Xian Electronic Science and Technology University, 2013: 18-19.

[20] Ling dehua. Research and implementation of automatic generation of code based on MDA D. Shanghai: east China normal university, 2006.

[21] Feng yingying. Research and application of code generation technology based on MDA D. Xian: xian university of electronic science and technology, 2011.

[22] Rahmouni M’hamed, Mbarki Samir. MDA-based ATL Transformation to generate MVC 2 Web modlesJ. Internationnal Journal of Computer Science & International Technology, 2011, 3(4):57-70.

[23] Frohner Á, Porkoláb Z, Varga L. Code Generation from UML modelsJ. Electrical Engineering, 2000, 44(2): 141-157.

[24] Wang xiaoyu, qian hongbing. Research on automatic generation method of C++ code based on UML class diagram and sequence diagram J. Computer application and software, 2013, 30(1).

[25] Mathupayas Thongmak, Pomsiri Muenchaisri. Design of roles for transforming UML sequence diagarams into java codeC//Proceeding of Ninth Asia-Pacific Software Engineering Conference.Gold Coast, Australia, 2002.

[26] He biao. The document object model (DOM) and its application J. Modern commercial industry, 2009(11):291-293.

[27] Chen xiaofeng, wang shian. Technology of Java based document object model J. Computer and automation, 2004, 22(2):49-50.

[28] Cleaveland J Craig, cheng hu jun, liu jiqiang. Using XML and JAVA to create program generator M. Beijing: science press, 2002.

[29] Dong yuming. Research and application of code generation technology in the development of management information system D. Jilin: jilin university, 2012:14.

[30] Zhang cheng. Research and implementation of AOP code generators for supporting contracts J. Microcomputer applications, 2007, 23(1):1-4.

[31] Zhang jin. Research on the code generation platform based on EJB components. Design of code generator and template library D. Zhejiang: Zhejiang university of technology computer application,

2004:18-19.

[32] Matt Stephens. Automated Code GenerationEB/OL. (2002-06)2018-1-23. Articles.softwarereality.com.

[33] Liu kaifeng. Research on automatic generation of application software based on business platform --code generator and rules engine design D. Zhejiang: Zhejiang university of technology, 2005:6.

[34] Yang fuqing, mei hong, li keqin. Software reuse and software component technology J. Electronic journal, 1999(2).

[35] Yin yanjun. Research and implementation of code generation system in Web application code generation platform D. Beijing university of technology, 2007.

[36] Quatrani T. Visual modeling with rational rose 2002 and UMLM. Addison-Wesley Longman Publishing Co., Inc., 2002.

[37] Wang jinheng, wang pu, li yafen. Research on the transformation method of business logic model to code for MDA J. Computer technology and development, 2012, 22(11): 36-40.

[38] Smart Code. Smart Code Generator (Asp.Net) EB/OL. (2018-01-29) 2018-01-29. http://smartcodegenerator.codeplex.com/.

[39] CodesMith. CodesMith ToolsEB/OL. (2018-01-29) 2018-01-29. http://www.codesmithtools.com/.

[40] Codematic. Dynamic soft code generator EB/OL. (2018-01-29)2018-01-29. https://baike.baidu.com/item/ Dynamic soft code generator/.

[41] Liang xingbo. Application of FreeMarker template engine in Java development J. Silicon valley, 2013(21):46.

[42] Gao xiaoqing. Research on embedded system design method based on JET technology J. Microcomputer information, 2009 (32):59-61.

[43] Stanford. The Stanford Parser: A statistical parserEB/OL . (2017-10-20) 2018-01-24. https://nlp.stanford.edu/software/lex-parser.shtml.

[44] Berkeley. The Berkeley NLP Group/OverviewEB/OL . (2017-11-01) 2018-01-24. http://nlp.cs.berkeley.edu/index.shtml.

[45] Fudan. Fudan Natural Language Processing Group EB/OL . (2017-11-15) 2018-01-24. http://nlp.fudan.edu.cn/software/.

[46] HanLp. HanLp:Han Language ProcessingEB/OL . (2017-11-20) 2018-01-24. http://hanlp.linrunsoft.com/.