



A RESEARCH ON JAVA DYNAMIC CODE GENERATION

1st DHARSHINI KUMAR
Dept. of Computer Science and Engineering
Meenakshi Sundararajan Engineering College
Chennai, India
dharshini.9101@gmail.com

2nd Dr.B.MONICA JENEFER Associate Professor
Head of Dept, Computer Science and Engineering
Meenakshi Sundararajan Engineering College
Chennai, India
monicamaheshwaran@gmail.com

Abstract- Java, being a highly versatile programming language, is widely used across various industries and applications. Its popularity in modern software development can be attributed to its platform independence, object-oriented programming approach, and strong security features, making it a vital tool for creating scalable and efficient applications. Java is the backbone of many renowned software programs. Among the different phases of the software development life cycle, the code writing phase tends to be the most time-consuming due to the meticulous planning and attention required to ensure that the code is functional, efficient, and scalable. However, if this phase takes too long, it can result in difficulties during testing. To tackle this issue, Java dynamic code generation was introduced, which is the process of creating Java source code based on simple inputs. This efficient code generation approach is a game-changer in the software development industry. The objective of this paper is to provide an overview of the different techniques available for generating Java code and evaluate their advantages and disadvantages.

Keywords -Java code generation, POJO jars, Code generation tools, String templates, Velocity template, Code smith, Jcode Model, Java poet, MyBatis Generator, Celerio, Json, xml, NLP

1.INTRODUCTION

Dynamic code generation refers to the process of generating executable code during runtime rather than during compile-time. This approach offers the flexibility to create code based on input data or conditions, resulting in more efficient software systems. In Java, dynamic code generation is commonly used to create new classes and methods in real-time or modify existing ones, based on the program's runtime requirements. This technique enhances code reusability, eliminates redundancy, and optimizes the implementation of complex algorithms. However, current tools and web applications for dynamic code generation are not highly efficient. This paper evaluates the available tools and various methods for code generation and proposes techniques to enhance the technology.

2.LITERATURE REVIEW

[2]describes the code generation process which includes the system modeling in UML state chart diagram, generation of XML for the model, parsing XML and then generating code.

In [3] Automatic java code is generated by defining the xUML (Activity diagram) formal definition which is used to create the hierarchical syntax chart(HSC).This HSC is used as a source for generating java code.

The proposed Document Responsibility Collaboration method in [4] uses the domain concept in which domains are made up of documents, and a document defined in a domain can work with another document in any domain to fulfill its responsibility. Document Responsibility Collaboration

method defines a process, which starts at the transition from analysis to design phase and continues to the code generation phase.

[5] uses business level use case for generating code. Business level use case was generated as CIM model followed by system use case generation as PIM model. PIM model is then transformed into class diagram (PSM). The tool called xGenerator is used to convert the PSM into java code.

The proposed approach in [6] considers business rules as input, uses the triplestore model for knowledge representation based on business rules, utilizes association rules to suggest attribute data types, and produces an abstract data model. This abstract data model is a framework for software components generation of various purposes and syntax, such as SQL scripts for database tables creation and Java Beans for server-side implementation.

In [7] the activity diagram and sequence diagrams are defined as part of the UML definition to define the object flow of the system and interaction between the objects, respectively. An XMI schema is a text representation of any software model that is exported from a modeling tool. The modeling tool BoUML exports the required schema from the given input models such as sequence diagrams and activity diagrams. The proposed JC_Gen extracts artifacts from the XMI schema of these two models to generate the code automatically.

The proposed system in [8] uses NLP to convert Natural Language into programming Language independent pseudo code. Then, using traditional programming, this system converts the generated pseudo code into programming language-dependent source-code.

In [9] JaCoText, a pretrained model based on Transformers is used to convert the NL to java code. First, Initialize our model from pretrained weights of CoText-1 CC and CoText-2CC, instead of performing a training from scratch. Later, an additional pre-training step is done using data that belongs to a specific programming language. Then conversion takes place.

[10] describes the metric model for testing the generated java code. In this approach .The metric model was created based on quality and efficiency of source code. Automatically generated code was tested for the created characteristics .The Results are analyzed to compare the various automatic code generation tools.

3.METHODOLOGY

The study was conducted using three different methodologies. Firstly, extensive online research was carried out to explore various methods for generating Java code. Secondly, numerous literature papers, research papers, and journals were analyzed to gather further insights. Lastly, various online tools for generating Java code were analyzed and evaluated to obtain a comprehensive understanding of the topic.

4.RESULTS

After conducting the study, it was found that there are several tools and techniques available for generating Java code. These methods have their respective advantages and disadvantages, which are outlined below.

FROM ONLINE RESEARCH:

There are two methods for generating Java code, namely using templates or POJO (Plain Old Java Objects) jar files. The use of templates is considered the optimal approach for generating a generalized form of code as compared to POJO jar files. While the predefined nature of POJO jars makes them easy to implement, generating customized classes using this approach can be challenging. Additionally, implementing inheritance features using POJO jar files can be complex, whereas it is relatively straightforward with templates.

FROM RESEARCH PAPERS:

REFERENCE NUMBER	ADVANTAGES	DISADVANTAGES
2	This method of code generation supports state concurrency and history without Compromising the expandability, reusability and understandability	This paper proposed code generation from UML state chart diagrams. Since the state chart diagram shows the state of the entity classes (objects), the method proposed in this paper is best suited for entity classes and not for boundary and controller classes.
3	This paper proposes an extended xUML language based on the UML language and solves the problem of incomplete semantics of UML.	The method proposed in this paper involves complex conversion (two step).This paper does not address the problem of model integrity and consistency check problems
4	This paper proposed the Streamlined design process for a project with a document concept arises.Code quality with automatic code generation and standardization is achieved.	This paper does not define the standard tool to check the generated code (Manual testing is done). Proposed model is inefficient for larger systems
5	The architecture of the generated Web applications respects a variant of the well-known Model-View-Controller (MVC) pattern	With respect to the general case, the current version of xGenerator proposed in the paper is not able to generate the control flow inside the body of the subclasses of because it is not possible to model it with sequence diagrams.Expensive approach
6	Produced SQL scripts and Java classes are completely operable, tables with foreign key constraints were created and Java Beans were compiled. Generated artifacts can be used in an information system software development project after minor tuning.	Proposed method of generating source code shows naming issues and generic exceptions misuse in the verification and validation process.

7	The procedures proposed in the paper can also be extended to develop GUI programs using java without the knowledge of syntax. The final code produced by the work is 95% compilable code.	The proposed approach may have the risk of error propagation .Small change in XMI affects the entire process of code generation.
8	Time taken to generate input data statements from Seed statements is less hence increases speed of execution. The proposed system can be used as a suggestion system where the mail is analyzed and a suggested solution is provided to the programmer.	The proposed solution can support for 'python' and 'C' only.
9	Proposed paper proved that, each modification in this models, such as the additional training, allows JaCoText to have better comprehension of the java programming language.	It is difficult to find the efficient pre trained model. Additional training will increase the cost and time
10	This approach reduces the cost of software development to increase the efficiency of software generation and to improve the quality of software products. The analysis measurement results can show that the metric model of automatic code generation is reasonable, which can be helpful to measure the automatic code generation.	There are some metric characteristics that are not reflected in the above experimental samples, such as the error number of source code generation at run-time. The proposed system is static in nature.

FROM CODE GENERATION TOOLS:

CodeSmith

CodeSmith is a software tool used for code generation in various programming languages, including Java. It allows developers to automate the repetitive and time-consuming task of writing code by creating templates and generating code based on those templates.

CodeSmith is a popular code generation tool that comes with many advantages. Firstly, it allows developers to generate code quickly and efficiently, saving them time and effort. The tool also supports multiple programming languages, including C#, Java, and VB.NET, making it versatile and widely applicable. CodeSmith has a simple and user-friendly interface, making it easy for developers to learn and use. Additionally, the tool allows for customization of templates, enabling developers to

tailor their code generation process to their specific needs. CodeSmith also offers great customer support, with a helpful community and technical support team.

However, like any tool, CodeSmith has its disadvantages. One of the main drawbacks of the tool is that it can be expensive, particularly for small businesses or individual developers. Additionally, while the tool allows for customization, it can be complex and difficult to understand for those who are not experienced with code generation. Finally, CodeSmith has a limited set of templates and may not offer the flexibility that some developers need for their projects.

JCodeModel

JCodeModel is a Java library that provides a way to generate Java source code in a programmatic way. It allows developers to create a model of the source code they want to generate and then output the source code in various formats such as plain text, in-memory representation, or a file.

The library's biggest advantage is its ability to generate code from any source, whether it is from a database schema or an XML file, which saves developers a lot of time and effort. JCodeModel is also open source, with an active community of contributors, which means it is constantly updated and improved.

Although JCodeModel offers numerous advantages, it also has some limitations that must be considered. One of the main challenges is that it can be challenging to understand for developers who are not familiar with Java code generation. The API can be complex, and there is a learning curve associated with using JCodeModel effectively. Furthermore, for large-scale projects, JCodeModel may not be the most appropriate tool since it can become difficult to manage and unwieldy as the codebase grows. In such instances, it may be preferable to use a more specialized code generation tool or to hire a dedicated code generator.

MyBatisGenerator

MyBatis Generator takes an XML configuration file as input, which specifies the database schema and the desired output for the generated code. The configuration file also includes instructions for mapping database tables to Java objects and specifying the SQL statements to be used for database operations. Additionally, MyBatis Generator requires a JDBC driver to connect to the database.

MyBatis Generator offers several advantages, such as its extensive support for different databases and data types. It allows developers to generate code for popular databases like MySQL, Oracle, and SQL Server, and supports various data types including integers, strings, and dates. Additionally, MyBatis Generator works seamlessly with the MyBatis framework, making it a valuable tool for developers who are already utilizing MyBatis in their projects.

Although MyBatis Generator has its advantages, there are also some limitations that should be considered. One disadvantage is that the generated code can be challenging to modify or extend. This could present a significant challenge for developers who need to make changes to the code, as they may need to invest significant time and effort understanding the code before they can make the necessary modifications. Additionally, MyBatis Generator may not be the best option for projects that require complex or highly customized SQL queries, as the tool may not be able to generate the required code to handle such queries.

Celerio

Celerio is a code generation tool that offers several advantages. First, it is a very versatile tool that supports a wide range of technologies, including Java, Spring, Hibernate, and AngularJS. This

makes it an ideal choice for developers who work with a variety of technologies and need a tool that can handle different types of code generation tasks.

Celerio is easy to use and offers a simple, intuitive interface that makes it easy to get started with code generation. The tool provides a comprehensive set of templates and pre-built code snippets that can be used to quickly generate code for common use cases, such as database access, CRUD operations, and UI components.

There are also some limitations to using Celerio. One of the main drawbacks is that the tool can be slow and resource-intensive, particularly when generating large amounts of code. This can be a problem for developers who need to generate code quickly and efficiently. Additionally, Celerio may not be the best choice for projects that require highly customized or complex code, as the tool may not be able to generate the necessary code to handle these scenarios. Finally, Celerio is a proprietary tool that requires a license, which may be a barrier to adoption for some developers or organizations.

Java poet

JavaPoet is a Java library for generating Java source code. It provides a simple and intuitive API that allows developers to programmatically generate Java code, including classes, methods, fields, annotations, and more. JavaPoet is designed to be easy to use and to produce readable and maintainable code. It also supports the latest Java language features, such as lambdas, method references, and default methods, making it a versatile tool for modern Java development.

Additionally, JavaPoet allows for the creation of complex Java code with minimal boilerplate code, thus saving developers valuable time and effort. The tool also provides support for various Java language features, including annotations, generics, and lambda expressions, and generates code that is highly readable and maintainable.

Java poet requires a solid understanding of Java programming to use effectively, meaning it may not be the best choice for beginner developers. Additionally, while JavaPoet is highly customizable, it may not provide the level of flexibility needed for highly specialized code generation tasks. Lastly, JavaPoet generates code at runtime, which may impact performance and memory usage for large-scale projects.

5.DISCUSSION

Java code generation tools commonly rely on UML diagrams as a means of input and typically produce either .java or .class files as output. While the creation of UML diagrams may appear to be a straightforward task, it can become complex when transforming them into Java code. This is because another tool is often necessary to convert the diagrams into text, which is then utilized as input for code generation. Consequently, the usage of UML diagrams is not an efficient approach. In lieu of this, XML or JSON files are recommended due to their ease of parsing. JSON is gaining more popularity due to its simplistic structure and easy-to-understand nature. Thus, it is suggested to employ a JSON file as input to generate code.

Despite the benefits of using JSON, there are also more advanced techniques such as machine learning and natural language processing (NLP) that can be utilized for creating Java code. However, these techniques require a significant amount of training data to create the model. This makes the process of code generation using these methods much more expensive and time-consuming compared to using UML, XML, or JSON files. Therefore, when considering the approach to code generation, it is important to weigh the pros and cons of each technique and determine which approach best suits the project's requirements.

6.CONCLUSION

To sum up, Java code generation tools have become indispensable in the software development process as they enable developers to automate code writing, thereby saving time and effort. While each tool has its own advantages and disadvantages, JCodeModel offers flexibility, MyBatis Generator supports multiple databases and integrates well with the MyBatis framework, Celerio is highly customizable, and Java Poet provides a simple and intuitive API. It is crucial to select a tool that aligns with the project's specific requirements. Additionally, using UML diagrams as input for code generation can be an inefficient approach, and it is better to use XML or JSON files instead. With the advancement of technology, more advanced techniques like machine learning and NLP are becoming popular to generate Java code, but these methods require a large training dataset to create the model, which makes the process more expensive. Overall, Java code generation tools play a critical role in modern software development, and choosing the right tool can greatly enhance productivity and efficiency.

7.REFERENCES

- [1]J. Rojas-Pérez, Member, IEEE, O. Fragoso-Díaz, Senior Member. IEEE, R. Santaolaya-Salgado, Senior Member, IEEE, and J. Soto-Orduño ,“Generation of POJOs and DAOs Classes from Metadata Database” , IEEE LATIN AMERICA TRANSACTIONS, VOL. 18, NO. 9, SEPTEMBER 2020.
- [2]Sunitha E. V. And Philip Samuel “Automatic Code Generation From UML State Chart diagrams.” , Digital Object Identifier 10.1109/ACCESS.2018.2890791 VOLUME 7, 2019
- [3]Zhenyu Wang, “A JAVA Code Generation Method based on XUML”,IOP Conf. Series: Materials Science and Engineering 2019
- [4]Tugkan Tuglular , Onur Leblebici “Automatic Code Generation w/th Document Responsibility Collaboration Modelling Method”,IEEE 2020
- [5]Gaetanino Paolone , Martina Marinelli, Romolo Paesani and Paolino Di Felice, “Automatic Code Generation of MVC Web Applications”,mdpi journals -computers-2020
- [6]Andrii Kopp and Dmytro Orlovskiy, “An Approach and a Software Tool for Automatic Source Code Generation driven by Business Rules”,14th International Conference on Advanced Computer Information Technologies (ACIT 2016) in Ceske Budejovice, Czech Republic.
- [7]Anand Deva Durai , Mythily Ganesh , Rincy Merlin Mathew,Dinesh Kumar Anguraj, “A novel approach with an extensive case study and experiment for automatic code generation from the XMI schema Of UML models”,The Journal of Supercomputing 2022
- [8]Aaqib ahmed r.h. ansari, dr. Deepali ,R. Vora, “NLI-GSC: A Natural Language Interface for Generating Source Code”,International Journal of Advanced Computer Science and Applications 2022
- [9]Jessica Lopez Espejel, Mahaman Sanoussi Yahaya Alassan, Walid Dahhane, El Hassane Ettifouri, “JaCoText: A Pretrained Model for Java Code-Text Generation” World Academy of Science, Engineering and Technology International Journal of Computer and Systems Engineering Vol:17, No:2, 2023
- [10]Zhen Li,Ying Jiang1, Xiao Jiang Zhang and Hai Yan Xu, “The Metric for Automatic Code Generation”,3rd international conference on mechatronics and intelligent robotics (ICMIR 2019)
- [11]Salvatore Calanna, Andrea Calvagna, Andrea Fornaia, Riccardo Torrisi, Emiliano Tramontana, “Automatic Generation of Parallel Java Programs and their Validation using Combinatorial Testing Suites”,IEEE the 6th International Conference on Computer and Communication Systems 2021