

## **Lecture 4 : Synthesis of Research Paper:"An Approach and a Software Tool for Automatic Source Code Generation Driven by Business Rules"**

### **Introduction:**

The research paper titled "An Approach and a Software Tool for Automatic Source Code Generation Driven by Business Rules" addresses the critical need to improve time efficiency in software development through automatic code generation. Recognizing the need of adhering to low-code principles and catering to users with limited programming skills, the paper suggests an approach based on Semantic Business Vocabulary and Rules (SBVR) for intelligible specifications. The primary goal is to bridge the gap between business analysis and engineering by transforming SBVR business rules into an automated source code generating solution.

### **Methodology:**

The methodology is implemented in a systematic manner, with several stages. The process begins with the conversion of SBVR business rules into a triplestore model using SBVR Structured English. This conceptual representation of the subject domain is subsequently turned into a data model, which includes entities, characteristics, and relationships. The study describes a novel application of association rules to suggest attribute data types based on the SQLite databases' existing "Spider" dataset. The suggested data types are modified for interoperability with various software development technologies, such as common programming languages and database management systems.

### **Results and Discussion:**

The proposed approach is implemented using a Python-based software tool. External dependencies like json, sqlite3, pandas, and nltk are used for efficient processing. Using the derived data model, the software tool successfully generates a variety of software components, including classes, database scripts, and smart contracts. The article demonstrates the tool's capabilities by automatically generating Java Beans and SQL scripts for an e-commerce site. However, the debate recognizes limits, such as name concerns and generic exceptions in the resulting Java code. Despite these constraints, the created artifacts are considered genuine and functioning.

### **Conclusion:**

Finally, the study article summarizes the important conclusions and findings of the suggested approach to autonomous source code generation. It underlines the approach's suitability for a wide range of software implementations, such as SQL scripts, Java Beans, and smart contract code. The explanation of limits is followed by a positive outlook that encourages further exploration and improvement of the approach. The conclusion lays the groundwork for future work, recommending improvements to solve noted constraints and broadening the types of created software components. Overall, the study emphasizes the suggested approach's contribution to Intelligent Software Engineering methods, which promote a more fluid integration of software requirements and system design.

