Summary of "Overview of Generative Software Development"

By Jason Christian

Generative Software Development is a software development technique which allows the rapid creation of new type of the systems based on existing, reusable system units, models, and architectures. Generative Software Development makes use of Domain-Specific Language (DSL), which are languages that are very powerful and expressive to use in a specific problem with the cost of losing generality of building any kinds of application. By using DSL, a new variant of the system can be expressed or created automatically and rapidly.

Defining the problem and solution space is one of the main concepts of Generative Software Development. Problem mapping changes the problem into expressions that engineers can use to define the solution, such as DSL, systems, models, and what level of abstraction to be used. They are implemented into components, sometimes also a higher form of *domain-system language* generated by the problem space.

This paper includes important points such as:

1. Defining the term "Generative Software Development" and why it is important in building software systems
2. Key concepts on how "Generative Software Development" can be done in the analysis and design phase of a software development lifecycle

What this paper lacks are:

1. Detailed technical aspect of "Generative Software Development" to help the readers understand more on how it can be done in practice
2. Real-world examples of software or systems which are developed using "Generative Software Development" method

The important aspect we need to discuss is how we could apply this method of developing software in every phase of Software Development Lifecycle.