Abstract

To match the needs of the fast paced generation, the speed of computing has also increased enormously. But, there is a limit to which the processor speed can be amplified. Hence in order to increase productivity, there is a need to change focus from processing time to programming time.

Reduction in programming time can be achieved by identifying the domain to which the task belongs and using an appropriate Domain Specific Language (DSL). DSLs are constrained to use terms and concepts pertaining to an explicit domain making it much easier for the programmers to understand and learn, and cuts down the development time drastically.
In this paper, we will understand what a DSL is; explore a number of DSLs spanning various phases of software development life cycle in terms of features that elucidates their advantages over general purpose languages and perform in depth study by practically applying a few open source DSLs: 'Cascading', Naked Objects Framework and RSpec.

Reference


Index Terms

Computer Science Programming Languages

Key words

Domain Specific Language
Fluent Interfaces
Method Chaining
Domain Specific Languages

- RSpec
- Cucumber
- RGen
- Graphviz
- 'Cascading'
- Naked Objects Framework
- Maestro
- ScalaModules
- Make
- Rake
- Twill
- Twist
- SmartFrog
- Chef
- EC2 Deploy Framework