Domain Specific Languages

Where the Business Rules ARE the Code

The path from requirements to functioning code is often full of twists and turns, of opportunities for confusion and misinterpretation. The predictable outcome is that the resulting code does not match the original requirements and cannot keep up with the requirements as they inevitably change.

Domain Specific Languages (DSLs) help by providing a formal, concise, and clear structure for describing business rules. But in their usual form, DSLs are textual, meaning that you still have to write code, and the Business Analyst needs to learn the intermediary formal language.

What if you could --

* provide the Business Analysts with an easy-to-use business-rules editor that uses terms commonly used in the domain
* Directly generate code based on the business rules

In this session we’ll look at implementing a lightweight Projectional Editor that Business Analysts can use to describe business rules and the data elements that the rules interact with.

The changes made by the BA are used to update an Abstract Syntax Tree (AST) structure. A code generator walks the AST and creates the code to implement the business rules.

When business needs change, it’s a simple matter to edit the rules and re-generate the code. As a bonus, the same edits to the rules can be used to generate regression tests or a document that describes the business rules in shareable form.