Rules are changing everyday in the face of rapidly volatile business requirements. How do we handle this change while keeping our systems maintainable, reusable and extensible? How do we model and handle (represent) rules, for greater reuse and maintainability (and performance)?

Business rules are expected to change more frequently than the rest of the business object. Rules are frequently implemented within the context of the methods of a business object. They refer to other business objects from within another business object's method, creating a web of dependencies that are typically implicit. Thus, changing one business rule would impact a set of objects dependent upon that rule.

We will go over a pattern language for the design and implementation of business rules in object-oriented systems.

Practical step-by-step methods to build robust, flexible and pluggable business rules are demonstrated.