



MyBatis

基于MyBatis的自适应代码生成工具的研究与实现

付荣
21451026



MyBatis

MyBatis的缺点

- 需要维护**SQL**和结果映射**ResultMap**，工作量大
- 依赖于开发人员对**SQL**的掌握程度
- 不适应快速数据库修改：修改表结构，更换数据库



MyBatis

MyBatis Generator

- **MyBatis**官方提供了代码生成工具**MBG**
 - 基于**XML**配置
 - 自动生成**DAO**层，**Model**层，**Mapper**映射文件
 - **Example**类用于构造复杂的筛选条件，以满足复杂查寻
- **MBG**的缺点
 - 仅支持一些简单的**CRUD**（插入、查询、更新、删除）操作
 - 开发人员需要对分页查寻和多表级联查寻，手写**SQL**和对象
 - 一旦表结构修改或者替换别的数据库，这些手写代码都会被重新刷掉



MyBatis

课题的目标

- 结合领域特定语言（**DSL**）、模型驱动开发（**MDD**）、代码生成技术（**CG**），提出一个**MyBatis**自适应代码生成的解决方案
- 开发人员只需编写**DSL**脚本，即可完成**MyBatis**所有相关代码的全自动生成
- 不仅支持**MGB**的基础功能，还支持分页查寻，多表级联查寻
- 适应数据库的快速修改：替换数据库，修改表结构



MyBatis

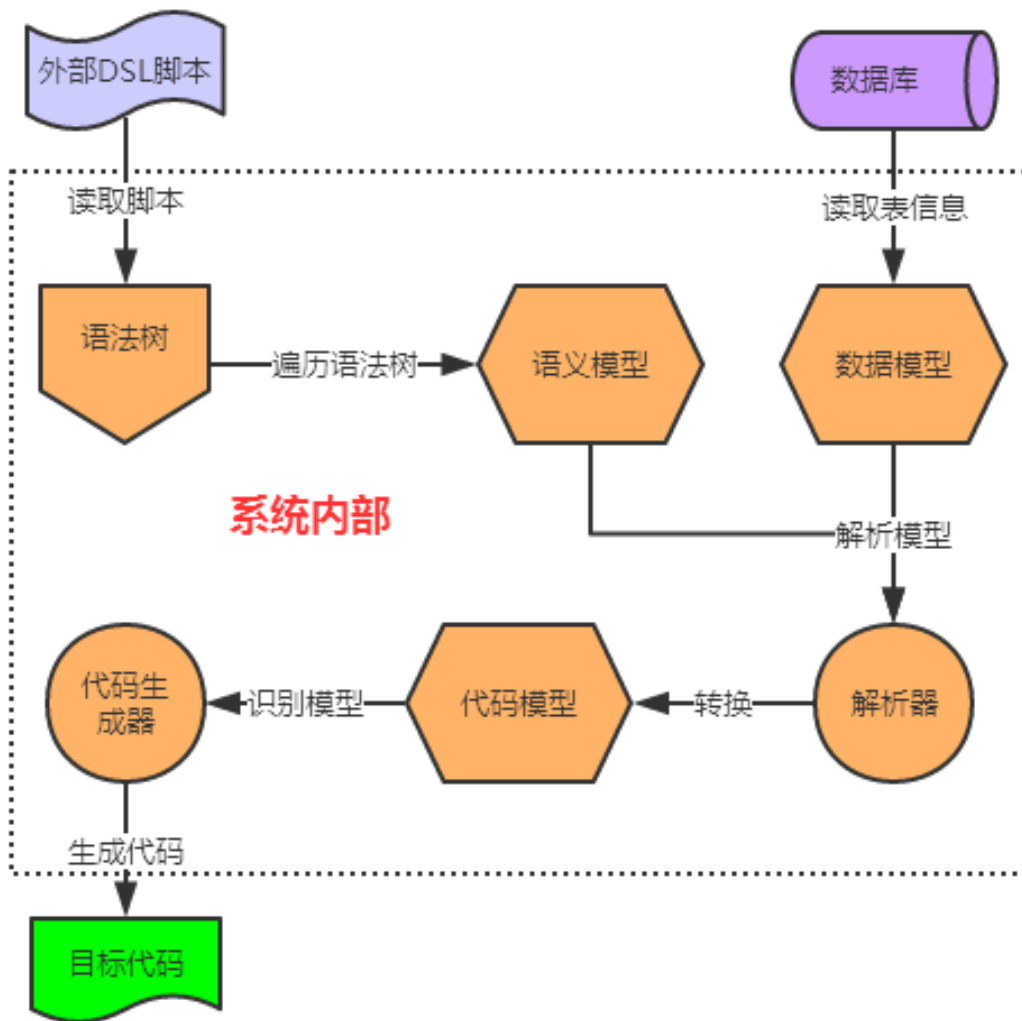
课题的意义

- 开发人员可以不知道**MyBatis**的使用细节，便可以快速地参与项目开发
(：新来的，对**Hibernate**产生依赖)
- 减少开发人员的工作量，节省时间，可以更专注于业务流程
- 降低编码的出错率，提高开发效率
- 提供统一的代码风格以及接口 (：**get**、**select**、**search**、**load**、**query**...)
- 促进**MyBatis**的推广和使用



设计方案

- 模块：外部DSL，语法树，语义模型，数据模型，解析器，代码模型，代码生成器。





MyBatis

计划

- 研究**MyBatis Generator**的源码，学习它的模型以及架构
- 学习领域特定语言，制定自己的**DSL**格式
 - Martin Fowler: 《Domain Specific Language》
 - SSV (SQL Syntax Validator) 实际开发经验
- 学习代码生成技术，找到合适当前项目的生成策略
 - Markus Voelter: 《A Catalog of Patterns for Program Generation》
 - Jack Herrington: 《Code Generation in Action》



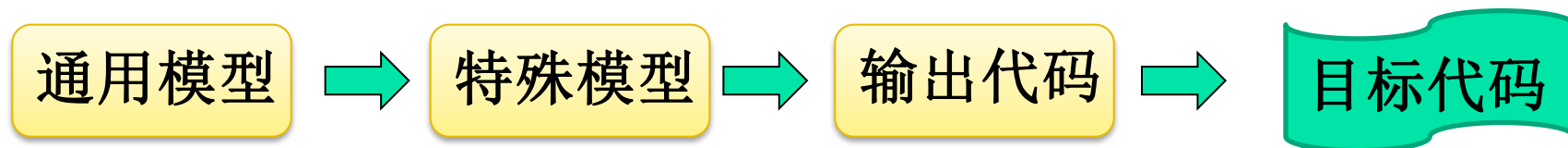
现有的思路-DSL

- **DSL**
 - 采用XML
 - XSD (XML Schemas Definition)
 - JAXB将XSD生成java类
 - 采用JAXB或者Ant读取XML



MyBatis

现有的思路-代码生成



- 1.应对**MyBatis**版本的变化
- 2.应对**Java**版本的变化
- 3.应对架构的变化，针对特殊项目结构，可以做优化
 - **Spring + MyBatis**



MyBatis

结束

谢谢