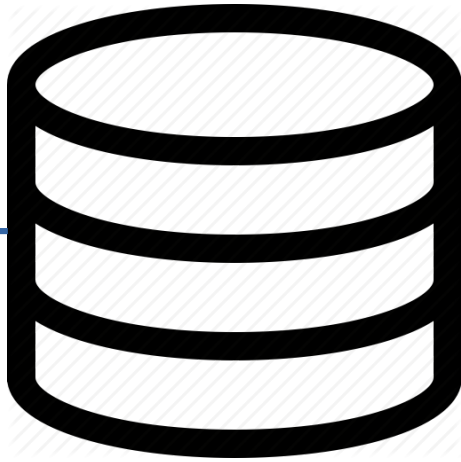


Analýza datových tokov v databázových systémoch

Autor: Richard Eliáš

Školiteľ: RNDr. Pavel Parížek, Ph.D.



```
1 public class Main {  
2     JdbcTemplate jdbcTemplate = ...;  
3     KafkaProducer kafkaProducer = ...;  
4  
5     public void copyUserFromDatabaseToKafka(int id) {  
6         User user = jdbcTemplate.queryForObject(  
7             "SELECT * FROM USER WHERE ID = ?",  
8             new UserRowMapper(),  
9             id);  
10        kafkaProducer.send(new ProducerRecord<>("User", user));  
11    }  
12 }
```

Typy programovej analýzy

- **Dynamická analýza**

- Chovanie programu sa určuje počas jeho behu

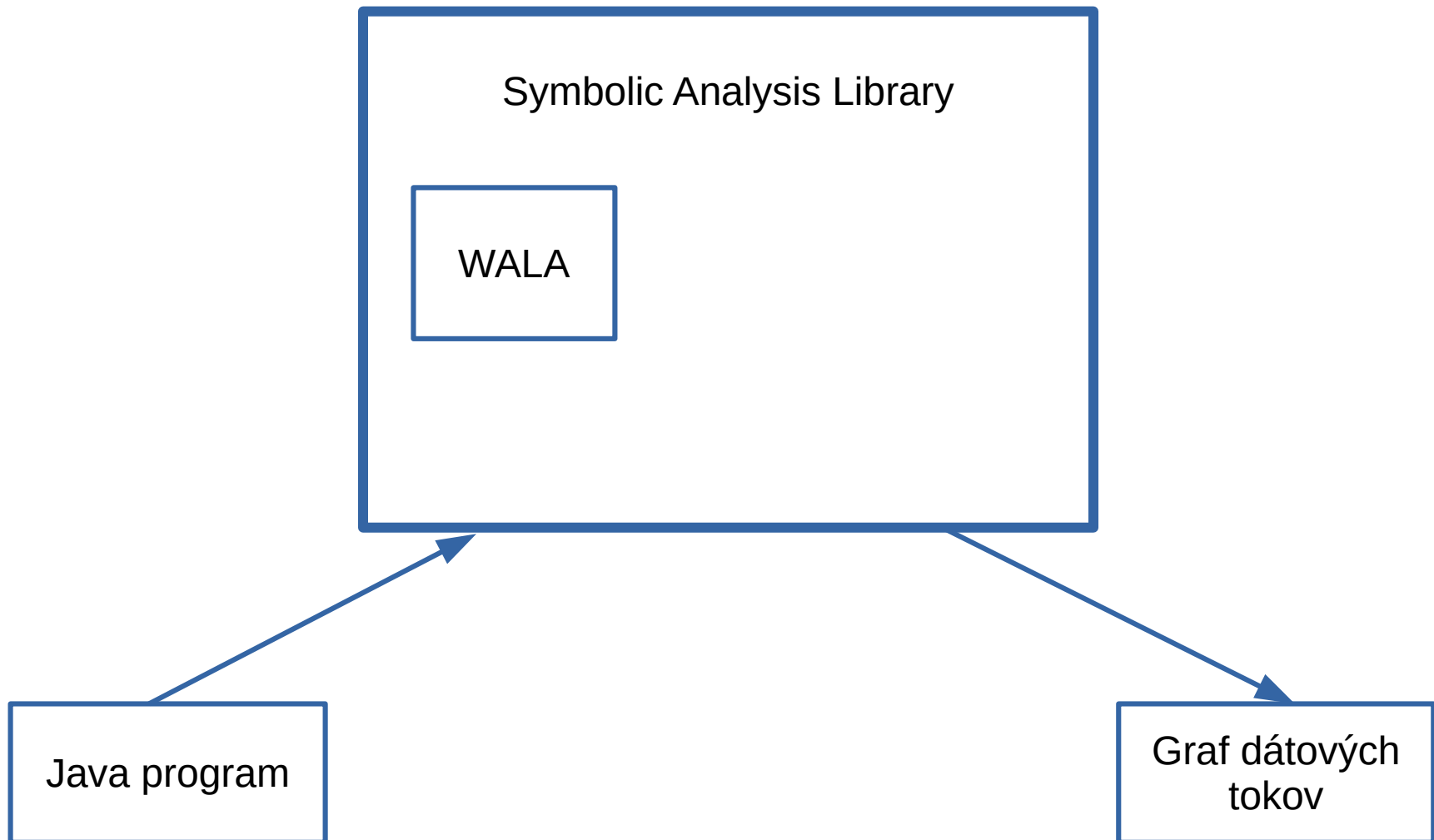
- **Statická analýza**

- Chovanie programu sa určí na základe kódu programu

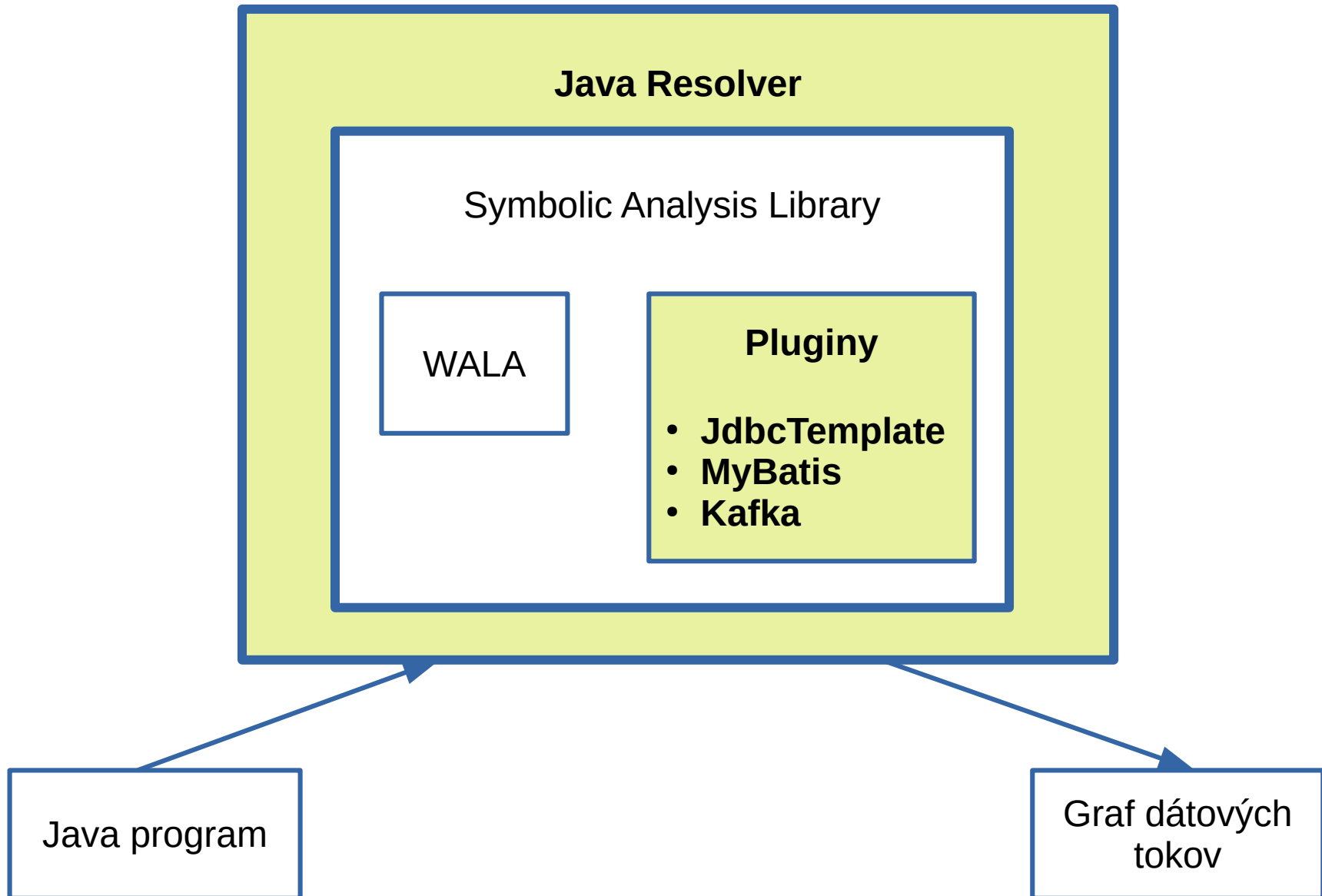
- **Kombinovaná analýza**

- Kombinácia statickej a dynamickej analýzy

Java Resolver



Java Resolver



MyBatis - príklad

```
1 public interface Mapper {  
2     @Results(value = {  
3         @Result(column = "ID", property = "id"),  
4         @Result(column = "VALUE", property = "value")  
5     })  
6     @Select("SELECT ID, VALUE FROM T WHERE ID = #{id}")  
7     DatabaseValue getForId(int id);  
8  
9     @Insert("INSERT INTO T (ID, VALUE) VALUES (#{id}, #{value})")  
10    void insert(DatabaseValue value);  
11 }
```

MyBatis - příklad

```
1 public interface Mapper {
2     @Results(value = {
3         @Result(column = "ID", property = "id"),
4         @Result(column = "VALUE", property = "value")
5     })
6     @Select("SELECT ID, VALUE FROM T WHERE ID = #{id}")
7     DatabaseValue getForId(int id);
8
9     @Insert("INSERT INTO T (ID, VALUE) VALUES (#{id}, #{value})")
10    void insert(DatabaseValue value);
11 }
```

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE configuration
3     PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-config.dtd">
5 <configuration>
6     <environments default="environmentId">
7         <environment id="environmentId">
8             <transactionManager type="JDBC"/>
9             <dataSource type="POOLED">
10                 <property name="driver"
11                     value="oracle.jdbc.driver.OracleDriver"/>
12                 <property name="url"
13                     value="jdbc:oracle:thin:@//192.168.0.16:1521/orcl"/>
14                 <property name="username" value="User"/>
15                 <property name="password" value="Password"/>
16             </dataSource>
17         </environment>
18     </environments>
19     <mappers>
20         <mapper class="Mapper"/>
21     </mappers>
22 </configuration>
```

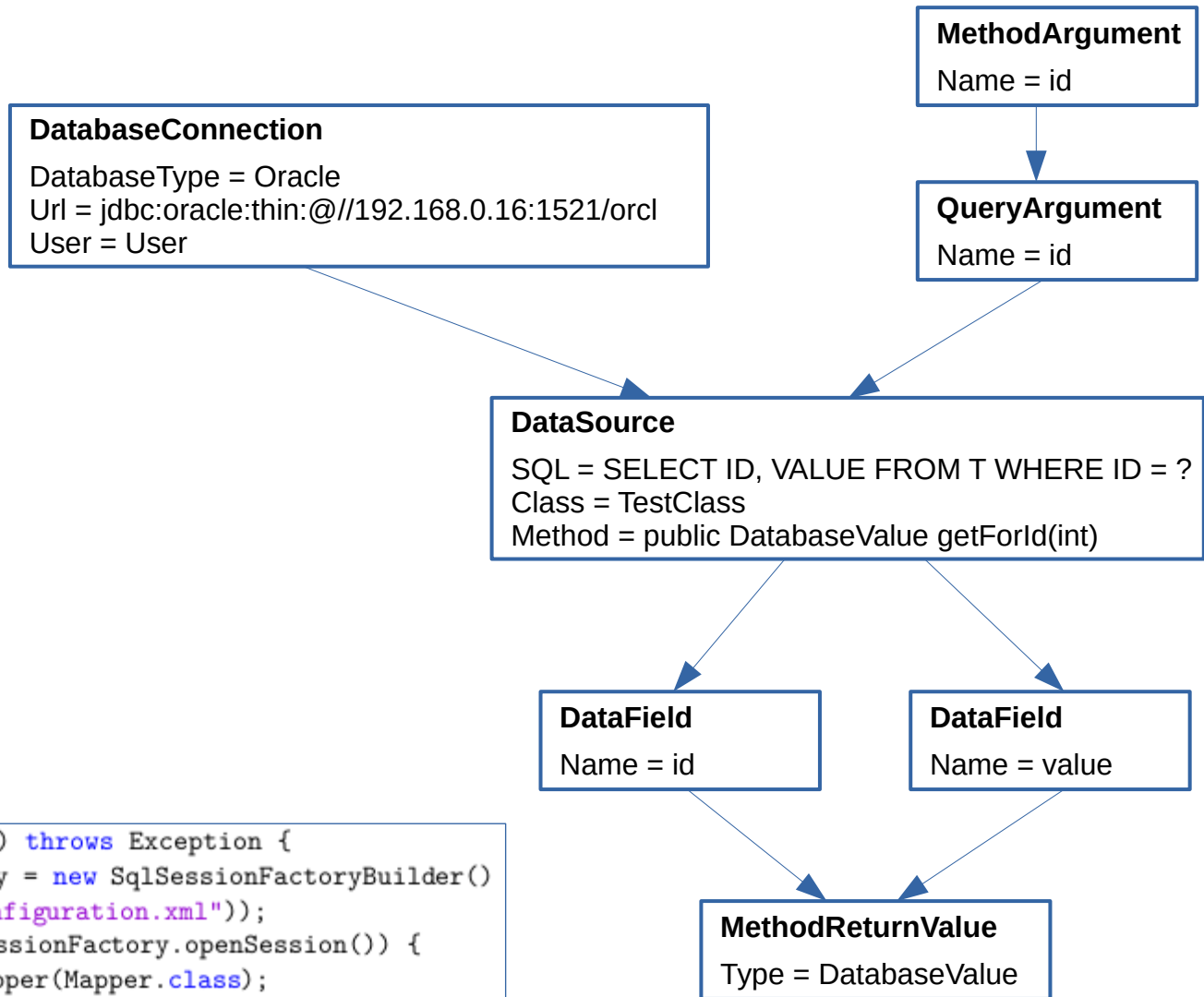
MyBatis - příklad

```
1 public interface Mapper {
2     @Results(value = {
3         @Result(column = "ID", property = "id"),
4         @Result(column = "VALUE", property = "value")
5     })
6     @Select("SELECT ID, VALUE FROM T WHERE ID = #{id}")
7     DatabaseValue getForId(int id);
8
9     @Insert("INSERT INTO T (ID, VALUE) VALUES (#{id}, #{value})")
10    void insert(DatabaseValue value);
11 }
```

```
1 public DatabaseValue getForId(int id) throws Exception {
2     SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder()
3         .build(new FileInputStream("configuration.xml"));
4     try (SqlSession sqlSession = sqlSessionFactory.openSession()) {
5         Mapper mapper = sqlSession.getMapper(Mapper.class);
6         return mapper.getForId(id);
7     }
8 }
9
10 public void insert(DatabaseValue value) throws Exception {
11     SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder()
12         .build(new FileInputStream("configuration.xml"));
13     try (SqlSession sqlSession = sqlSessionFactory.openSession()) {
14         Mapper mapper = sqlSession.getMapper(Mapper.class);
15         mapper.insert(value);
16     }
17 }
```

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE configuration
3     PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-config.dtd">
5 <configuration>
6     <environments default="environmentId">
7         <environment id="environmentId">
8             <transactionManager type="JDBC"/>
9             <dataSource type="POOLED">
10                 <property name="driver"
11                     value="oracle.jdbc.driver.OracleDriver"/>
12                 <property name="url"
13                     value="jdbc:oracle:thin:@//192.168.0.16:1521/orcl"/>
14                 <property name="username" value="User"/>
15                 <property name="password" value="Password"/>
16             </dataSource>
17         </environment>
18     </environments>
19     <mappers>
20         <mapper class="Mapper"/>
21     </mappers>
22 </configuration>
```


MyBatis - příklad



```
1 public DatabaseValue getForId(int id) throws Exception {  
2     SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder()  
3         .build(new FileInputStream("configuration.xml"));  
4     try (SqlSession sqlSession = sqlSessionFactory.openSession()) {  
5         Mapper mapper = sqlSession.getMapper(Mapper.class);  
6         return mapper.getForId(id);  
7     }  
8 }
```

Java Resolver

Obmedzenia nástroja:

- Dĺžka analýzy programov rastie v závislosti na ich veľkosti a veľkosti knižníc
- Overaproximácia
- Neznáme hodnoty premenných závislých na vstupe

Java Resolver

Podporované funkcie nástroja:

- Identifikácia zdrojov a stokov dát
- Podpora pre prácu s externými konfiguračnými súbormi
- Podpora pre prácu s anotáciami
- Podpora pre spracovanie interakcií medzi frameworkom a aplikáciou (callbacky)
- Ľahká rozšíriteľnosť

Ďakujem za pozornosť

Autor: Richard Eliáš

Školiteľ: RNDr. Pavel Parížek, Ph.D.

Komentáre

1. Ignorovanie pokročilých funkcií frameworkov

- Úlohou bolo vytvoriť prototyp aplikácie s podporou základných funkcií frameworkov, nie vyčerpávajúcu implementáciu všetkých jeho zákutí

Komentáre

1. Ignorovanie pokročilých funkcií frameworkov
2. Použitie existujúceho riešenia na modularizáciu
 - Tieto riešenia neboli vhodné kvôli integrácií do výsledného produktu Manta Flow

Komentáre

1. Ignorovanie pokročilých funkcií frameworkov
2. Použitie existujúceho riešenia na modularizáciu
3. Výber frameworkov
 - Výber konkrétnych frameworkov (Spring JDBC, MyBatis, Kafka) bol zvolený najmä pre ich rôzne spôsoby ich používania a prístupu k dátam
 - Konfigurácia pomocou XML a Properties súborov
 - Anotácie
 - Callbacky