

Conception and Realization of the Versioning of Databases between Two Research Institutes



S. Mueller, R. Mueller
GSI, Darmstadt, Germany

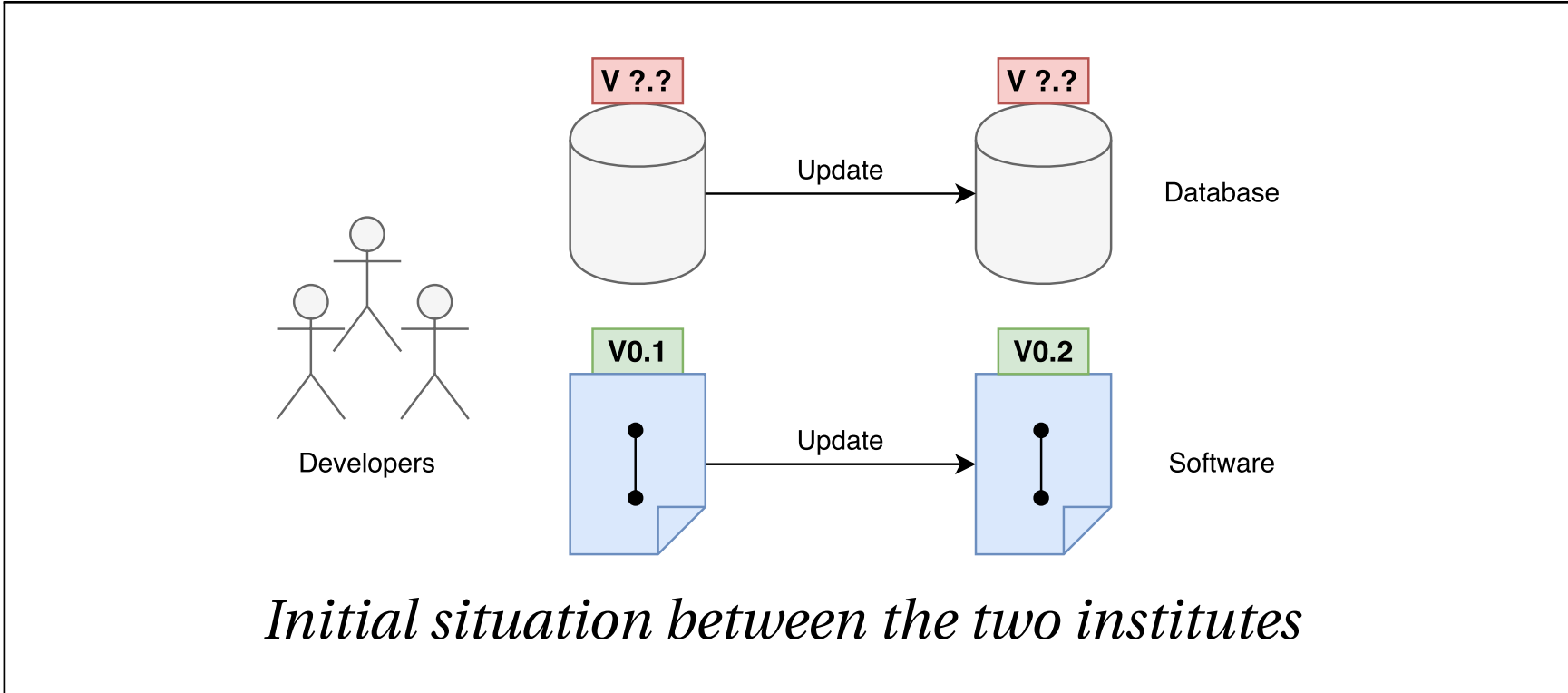


Abstract

This paper describes the version control of oracle databases across different environments. The basis of this paper is the collaboration between the GSI Helmholtz Centre for Heavy Ion Research (GSI) and the European Organization for Nuclear Research (CERN). The goal is to provide a sufficient and practical concept to improve database synchronization and version control for a specific database landscape for the two research facilities. First, the relevant requirements for both research facilities were identified and compared, leading to the creation of a shared catalog of requirements. In the process database tools, such as Liquibase and Flyway, were used and integrated as prototypes into the Oracle system landscape. During the implementation of prototypes several issues were identified, which arise out of the established situation of two collaborating departments of the research facilities. Requirements on the prototype were, to be flexible enough to adapt to the given conditions of the database landscape and easy integration without too many changes into the existing development environment. The creation of a flexible and adjustable system enables the two research facilities to use, synchronize and update the shared database landscape.

Current Status

- Collaboration between two institutes - CERN and GSI
- The applied frameworks - FESA and LSA
- Frameworks have dependencies on each other
- Software is versioned, database is not versioned



Procedure

First, a requirements catalog of the two institutes was created and taken as basis for the choice of the best method to collaborate and later on for evaluation of a suitable tool. The second step was to make a model of each possible method and test these models in a simulation environment. This approach helped to discover possible issues, which, based on the concept only, could not be detected.

- Asking each institute for the requirements
- Generate and evaluating the requirements catalog

Requirements

- Java-Code-Migration - For running Java code during the migration
- Separate Database - To generate a database from the scratch
- Traceability - Traceability of the code migration
- Repeatable Jobs - Repeatable code migration
- Existing Database - Setup the migration tool on a existing database
- Expandability - Expandability of the migration tool via plug-ins
- Open source - Open source and no cost
- Existing Environment - Included in the existing environment
- Cherry picking - Selection of the changes
- Data import - Import static data

Requirements	CERN	GSI
Java-Code-Migration	•	•
Separate Database	•	•
Traceability	•	•
Repeatable Jobs	•	•
Existing Database	•	
Expandability	•	•
Open source	•	•
Existing Environment	•	•
Cherry-picking		•
Data import	•	•

Requirements of each institutes

Evaluation

After collecting the requirements, the next step was to select tools which will fit theoretically. As it had to be an open source tool, there are two popular migration tools on the market - Flyway and Liquibase. Each of these tools fits theoretically to the requirements catalog. For a better exclusion process, an environment was created, which is similar to the production environments of both departments. The goal was to simulate the database landscape of both institutes, to be able to test the chosen migration tools in an environment close to the real-world.

Flyway - Advantages:

- Lightweight - no config file is needed
- K.I.S.S - principle
- Continuous delivery
- More compatible with Oracle database than Liquibase

Liquibase - Advantages:

- XML based
- Convert the XML to SQL during run-time
- Continuous delivery
- Expandable by plug-ins
- Liquibase has a few more articles on Stackoverflow than Flyway

Example of a file for Flyway

```
V00_01_00010__INSTALL_COMMONS_BASIC.sql
V00_01_00020__INSTALL_LSA_TABLES.sql
V00_01_00030__INSTALL_LSA_VIEWS.sql
```

Example of including a config file in Liquibase

```
<include file="initial_setup/c4o/c4o_log.xml"/>
<include file="initial_setup/lsa_common/lsa_common_log.xml"/>
<include file="initial_setup/lsa_gsi/lsa_gsi_log.xml"/>
```

Example of an XML config file from Liquibase

```
[...]
<changeSet id="2017-07-05_16-22_002" author="rmueller">
  <comment>The procedure 'clone_trims_settings' was
  lacking the possibility to clone settings for
  function arrays</comment>
  <sqlFile splitStatements="false"
  path="r009/lsa_gsi/
  2017-07-05_16-22_002__PB_SETTINGS_MANAGEMENT.sql" />
</changeSet>

<changeSet id="2017-08-08_15-48_001" author="smueller">
  <comment>Change Email-Server</comment>
  <sqlFile endDelimiter="/"
  path="r009/lsa_gsi/
  2017-08-08_15-48_001__PB_ADMIN.sql" />
</changeSet>

<changeSet id="2017-07-24_14-05_001" author="smueller">
  <comment> Update Commons 1.15 to 1.16 </comment>
  <sqlFile splitStatements="false"
  path="r009_1/c4o/2017-07-25-14-40_001__PB_com_per.sql"/>
  <sqlFile splitStatements="false"
  path="r009_1/c4o/2017-07-25-14-40_002__PB_com_util.sql"/>
  [...]
</changeSet>
[...]
```

Results

Because of the desire for "cherry-picking" and the versioning of two databases in one, both tools reach their limits. Liquibase is in this case better, it gains flexibility and has more advantages through the XML-configuration approach than Flyway.

Flyway:

- ✗ Can't handle two databases with one config file
- ✓ Can better handle PL/SQL and SQL in only one file
- ✓ Can delete the whole Oracle database
- ✗ Circular dependencies

Liquibase:

- ✓ More manageable through XML
- ✓ Can handle C4O and FESA or LSA at once
- ✗ Has problems parsing PL/SQL and SQL files in one step
- ✗ Can only delete general database objects
- ✗ Different ways to setup various database
- ✗ Circular dependencies
- ✓ Commons 4 Oracle (C4O)

Requirements	Flyway	Liquibase
Java-Migration	✓	✓
Separate Database	✓	✓
Traceability	✓	✓
Repeatable Jobs	✓	✓
Existing Database	✓	✓
Expandability	✗	✓
Open source and free	✓	✓
Cherry-picking	✗	✓
Data import	✓	✓
PL/SQL and SQL	✓	✗
Oracle compatible	✓	✗
C4O and Databases	✗	✓

Requirements of each institutes

Current Status - Outlook

At the moment GSI uses Liquibase for the database migration. CERN is in the phase of testing Liquibase. Since GSI was starting with an empty database from scratch they did not have to care about the content, so CERN is investigating at the moment how to handle the existing content and looking into baselining its database.

CERN:

- Evaluating Liquibase

GSI:

- Using Liquibase
- Changing existing database more and more to Liquibase

