PL/SQL code fragments into Spring Batch configuration via Xtext

Peter Sasse
07.11.2014
Objectives and Purpose

- State the purpose of one proof of concept
- What was the plan?
- Demo
- Look back
- «Architecture of uncertainty»
- Conclusion
From issues to proof of concept

- Poor performance due to
  - Lack of parallel programming in PL/SQL code
  - Data formatting on database server (network latency)

- Poor maintainability because of
  - Missing central source container (e.g. as SVN for Java)
Poor performance: Data formatting

Database

server

DB

PL/SQL

Batch

server

POJO

text

files

select LPAD('ID', 8, '0') from dual
Poor performance: Data formatting

Database server

PL/SQL

Format: '0000000ID'

Batch server

POJO
text files
So what was the plan? (I)

- Requirement: Generation of Spring Batch configuration
- For grouping Spring Batch jobs
  - Get overview of jobs and their dependent objects
  - Extract table names from PL/SQL code
  - create flowchart
So what was the plan? (II)

- Extract exporting SELECT statements:
  - Go down the recursion
  - Built Abstract Syntax Tree (AST) as EMF model
- Model-to-Text transformation:
  - Generate Spring Batch Job.cfg files with the use of Xtend
One proof of concept (I)

Xtext -> Ecore

EMF & Xtend

Data- & Control Flow diagram

Spring Batch

Oracle PL/SQL

SQL selects

Job.cfg
One proof of concept (II)

Xtext → Ecore → Data- & Control Flow diagram

Our first goal/milestone!

Oracle PL/SQL → SQL selects → Spring Batch Job.cfg
One proof of concept (III)

Our second & final goal/milestone!

Xtext

Ecore

Data- & Control Flow diagram

Oracle PL/SQL

SQL selects

Spring Batch

Job.cfg
Demo
Look back

- We started with Xtext before it moved
  - from openArchitectureware
  - to the Eclipse Modeling Project
- For us this project was great exercise, since 2010:
- We were uncertain, if for example memory consumption will explode
- We knew about the impact, if Spring Batch configuration will change (=> changes to Xpand files)
Conclusion

- “Uncertainty is an uncomfortable position, but certainty is an absurd one.” (@2nd Annual PHP North West Conference: Kevlin Henney)
- First know exactly the target (“know, where the finish line is”)
- Uncertainty: Find out impacts (then no probability of occurrence) on your “way”
- Be part of early design rendering in code to reveal additional design efforts
- We will carry on
  - Model-to-Model transformation (M2M): BMC Control-M model to Credit Suisse AG Java batch model
  - Eclipse Graphical Modeling Framework (GMF): Graphical Credit Suisse AG Java batch model editor
Q & A

- Please let me answer your questions
- Thank you for your time!
Backup slides