Juggling with Data: On the Lack of Database Monitoring in Long-Living Software Systems

EMLS 2017
4th Collaborative Workshop on Evolution and Maintenance of Long-Living Software System

Christian Zirkelbach, Kiel University

February 21, 2017
Example Software Landscape (Banking Industry)

Systems, Applications, and Databases

Evolution also affects databases!
Challenges

• Handling large-scale, enormous-in-size data repositories [Cuzzocrea et. al]

• Changing requirements or increasing workload

• Performance issues or customer requests [Zirkelbach et. al]
  → inevitable software updates or refactoring

• Legacy systems: often based on outdated technologies and poorly documented [Godfrey and German]

• Insufficient knowledge of the (actual) systems hamper the process [LaToza et. al]
Where to start?

Evolution

1. Gathering information
2. Evaluation

Extended system development life cycle (SDLC) based on [Avison and Fitzgerald]
Envisioned Approach
Envisioned Approach

Legend

Existing Application
Existing Database

Monitoring Data
132743373;createStatement;SELECT..
132743373;createStatement;SELECT..
132743377;createStatement;INSERT..
132743377;createStatement;INSERT..
132743380;databaseQuery;showTab...

Traces

Data Model

Visualization

Landscape-Level Perspective
Systems, Applications, and Databases

Architecture-Level Perspective
Entity Types and Relationships

Usage-Level Perspective
Entities, Relationships, and Joins
Conclusions & Open Questions
Conclusions

• Lack on database monitoring in long-living systems
  • Based on...
    • obsolete technologies and platforms
    • poor documentation
    • insufficient knowledge

• Presented an approach as a solution
  • Live database trace visualization for large software landscapes
  • Adresses developers and operators
  • Only first draft – open for feedback and suggestions
Open Questions

• Which visualization and layout is suitable for our landscape-level perspective, that comprises the complete software landscape including the databases?

• Does our 3D visualization within our usage-level perspective offer an advantage over a traditional 2D visualization like the architecture-level perspective?

• Which related approaches or tools could be employed, when evaluating our approach within a controlled experiment?


