Empirical Metadata Maintenance in Source Code Development Process

Karol Rástočný

Supervisor: Prof. Mária Bieliková

PeWe@FIIT
personalized web group

PerConIK
perconik.fiit.stuba.sk

Why Empirical Metadata?

- *Source code-based metrics*
  - Static view to source code properties
  - What is good/wrong but not why
- *Empirical software metrics*
  - Dynamic view to source code properties
  - Why has source code calculated properties
- *Problems of empirical metadata*
  - Collecting empirical data
  - Quality of collected data
  - Usability of empirical data

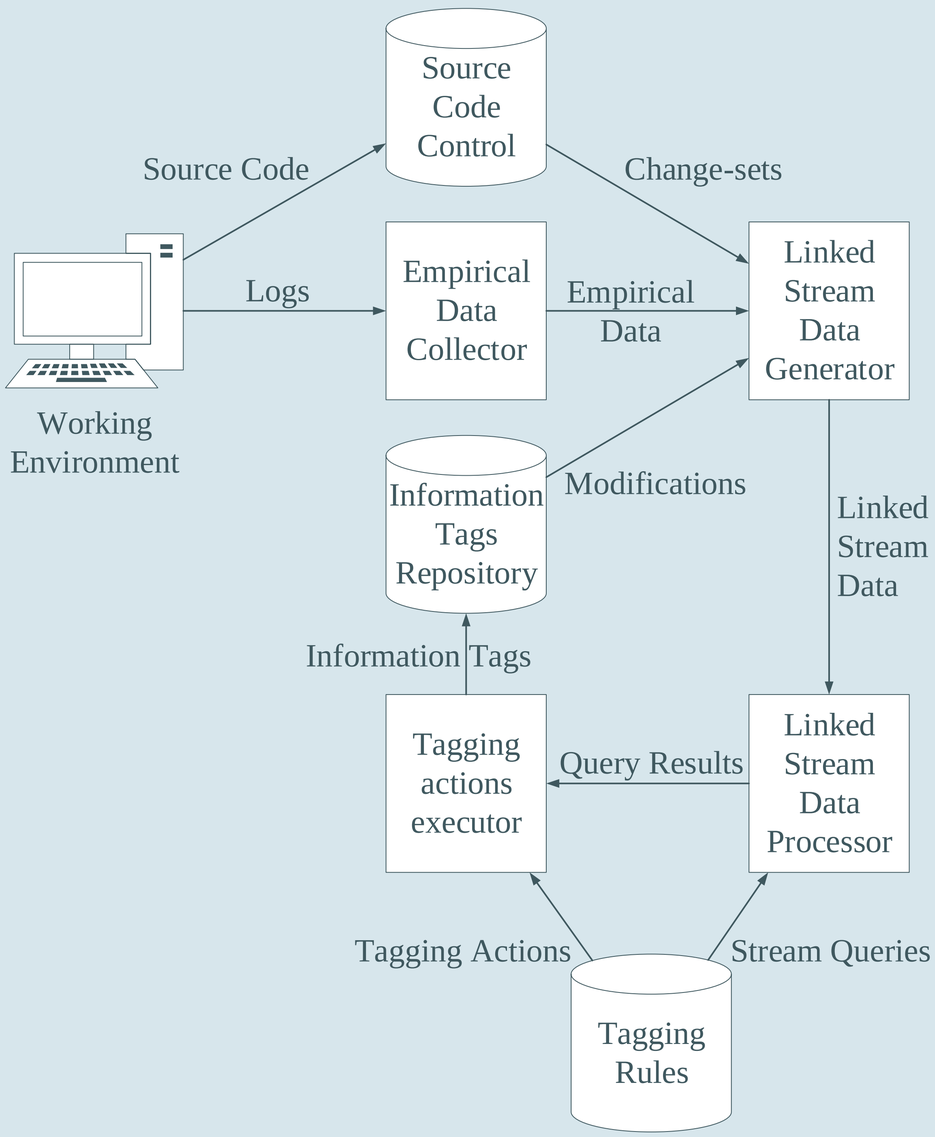
How? Information Tags!

- *Metadata anchored to an artifact*
  - *Type* – type of the stored information
  - *Anchoring* – the target artifact
  - *Body* – the stored information
- *Machine information tags*
  - *Content-based information tags*
  - *User activity information tags*
  - *Aggregating information tags*
- *User information tags*

Dependencies of Information Tags

- *Structure of the source code artifact*
- *Semantic of the source code artifact*
- *Empirical properties*
  - Context of the source code artifact
  - Influence to structure and semantic

Empirical Metadata Processing



Tagging Rules

- *Maintain information tags (empirical metadata)*
  - Create missing information tags
  - Repair invalidated information tags
  - Remove unrepairable information tags
- *Stream queries* – matches specific events and data related to these events
- *Tagging actions* – reactions to matches events, that are based on results of queries

Stream Generator

- *Generates a RDF stream of events about*
  - Activities of developers (empirical data)
  - Modifications in source code
  - Modifications in information tags space

Stream Processor

- *Based on C-SPARQL*
- *Evaluates stream queries from tagging rules in sliding windows over RDF stream*
- *Returns rows with results assigned to identifiers of tagging rules*

Tagging Actions Executor

- *Processes results of stream processor*
- *Executes tagging actions with results of queries*
- *Results are sets of modifications (add, update, delete) over information tags space*

STU FIIT

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava

www.fiit.stuba.sk
maria.bielikova@stuba.sk
karol.rastocny@stuba.sk