

Historization of Enterprise Architecture Models and Analysis Results for Optimization

Robin Bråtfors - CS Master Thesis at KTH



EA Debt

“ Enterprise Architecture Debt depicts the deviation of the currently present state of an enterprise from a hypothetical ideal state. ”

EA Smell

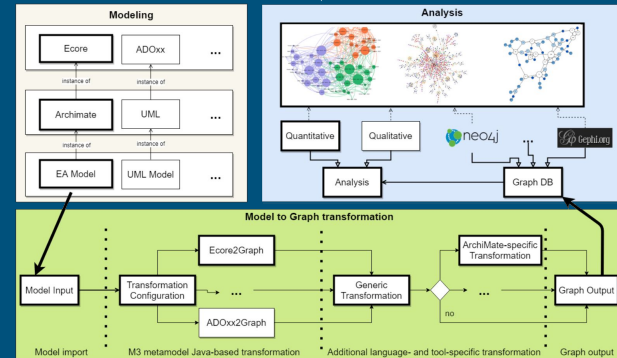
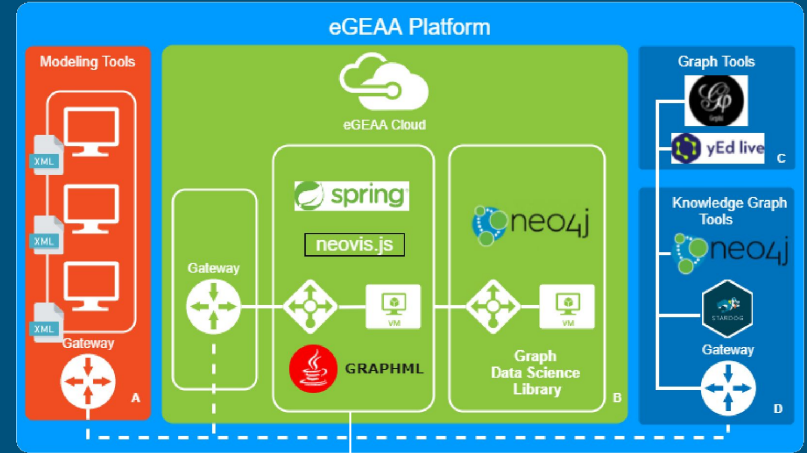
- Derived from Code Smell
- The symptoms of EA Debt
- Catalog of 63 smells

The screenshot displays the 'Enterprise Architecture Smells' website. The interface includes a search bar, a sorting dropdown, and a page indicator '63/63'. A left sidebar lists categories such as 'technology', 'business', 'application', 'Understandability Problem', 'Semantic Error', 'Rule-related defects', 'Data-flow related defects', and 'Control-flow Problems', along with tags like 'All', 'Process', and 'The Enterprise'. The main content area features a grid of six smell cards, each with a title, a description, and a set of icons for sharing and navigation.

Ambiguous Viewpoint	Architecture by Implication	Big Bang
Analysis and design models are often presented without clarifying the viewpoint represented by the model. Mixed viewpoints don't allow the fundamental separation of concerns, confusing blueprints of abstractions and implementation details.	This antipattern is characterized by the lack of architecture specifications for a system under development. Usually, the architects responsible for the project have experience with previous system construction, and therefore assume that documentation is unnecessary.	A strategy often preferred by large vendors where an entire Enterprise Architecture Model is built 'at once'.
Business Process Forever	Chatty Service	Combinatorial Explosion
Business processes have been strictly defined and are now static and cannot be easily changed.	A high number of operations is required to complete one abstraction. Such operations are typically rather simple tasks that needlessly slow down an entire process.	A subtle form of Duplication; this smell exists when numerous elements do the same thing using different combinations of data or behavior.

The CM2KG Platform

- Transforms EA models into key graphs
- Automatic EA Smell detection
- Graph Visualisation and Analysis
- No memory



Why do we want historization?

- Track changes
- Navigate through time to gain context and understanding
- Track dependencies
- Temporal queries
- Backup



Problem Statement

- How should the transformed graphs of EA models be stored and structured to preserve their history?
- How should an interface be designed to allow the models and analysis results of the CM2KG tool to be viewed, queried and extracted in an accessible way?

Graph Historization in previous research

Storage

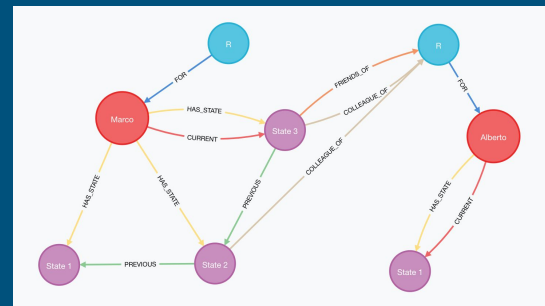
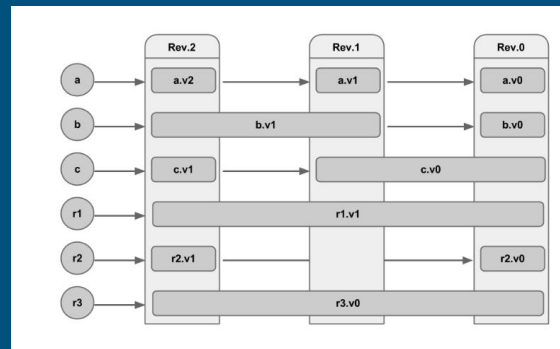
- Snapshot
 - Costly storage
 - Fast retrieval
- Delta
 - Efficient storage
 - Slow retrieval
- Granularity



Graph Historization in previous research

Representation

- Separation
 - Mixed
 - Isolated
- Structure
 - Linked Lists
 - Entity-State model
 - Tables
 - Whole graphs



Implementation (current focus)

- Check ID of the model
- Compare the uploaded model with its most recent version
- Add or update the timestamp of any changed element
- Add any change to an entity-state graph
- Save the uploaded model as the most recent version
- Present the entity-state graph as well as all singular versions in a queryable way

The screenshot displays the CM2KG web interface. At the top, it reads "CM2KG Conceptual Model to Knowledge Graph" and "Transform your conceptual model (Archi, ADOxx, Papyrus ...) to GraphML, and gain insights about your model by apply graph based-analysis." Below this, there is a "Version" dropdown menu currently set to "Version 2 - 2021/10/12". To the right, there is a "Time Range" selector and a search form with fields for "Name" (containing "operation"), "ClassName", "Label", "Smell", and "Function". Below the search form is a "Custom query" input field. At the bottom right, there are two buttons: "Query current" and "Query all". The main content area displays XML snippets for two nodes, each with keys for "LastUpdate", "ClassName", "Label", "name", and "documentation".

```
<node id="ff61d10e">
  <data key="LastUpdate">2021/11/25 11:37</data>
  <data key="ClassName">ApplicationFunction</data>
  <data key="Label">operation</data>
  <data key="name">operation</data>
  <data key="documentation"></data>
</node>
<node id="ff61d10e">
  <data key="LastUpdate">2021/11/27 10:23</data>
  <data key="ClassName">ApplicationFunction</data>
  <data key="Label">operation</data>
  <data key="name">operation</data>
  <data key="documentation">This needs documentation!</data>
</node>
```

Demonstration & Evaluation



- The historization will be demonstrated by using modified versions of pre-existing EA models and then running queries on parts of and the whole history
- The implementation will be evaluated on the accuracy of the historization and query results, as well as how practical it is to use based on feedback from EA practitioners

Feedback

Is there any specific historical data that you would find useful?
