Future Research Topics in Enterprise Architecture Evolution Analysis

Design For Future (DFF) Workshop, Software Engineering Konferenz, Aachen
1. März 2013

Sascha Roth, Florian Matthes
Software Engineering for Business Information Systems (sebis)
What is an enterprise architecture?

Common language for business and IT
- Technical, social, and economic aspects
- Layers and crosscutting concerns
- Relationships are more important than element details
Current practices used for documenting an EA

Current practices for gathering information about an EA
- Workshops,
- Interviews,
- Questionnaires,
- ...

Problem: Documenting an EA is a manual task [Ha12a, Ha12b, Ro13]
- Time consuming
- Error-prone
- Cost intensive
- Information is incomplete/invalid
- Information is soon outdated
- People don’t cooperate
- ...

At the same time:
- Decision makers must be supported with up to date information in a high quality
Our current research endeavor: Automated EA Documentation

Research Hypothesis
Relevant information for EA already is contained in operative IT environments!

Goals
- Gather data from operative IT systems
- Combine, harmonize, and integrate information in a common model
- Manage information (lifecycles, processes)
- Communicate results to stakeholders
- ...

Old and new questions
Over time, a plethora of 3rd party data (multiple versions) within EA repository. Validate data of different information sources, compare versions of an EA, etc.
Managed Evolution: Evolutionary EA Design controlled by Principles and Standards

IT development efficiency

Business benefit

cf. [Mu10, Bu10]
System Cartography: Timeinterval Map

- An analogy to a Gantt-Diagram
- Versions of:
  - Applications,
  - Projects,
  - Programs, or
  - Organisational Units
The dimensions *modeled at, planned for, and variants* may be combined.
How to analyze the evolution of an enterprise architecture?

Common language for business and IT
- Technical, social, and economic aspects
- Layers and crosscutting concerns
- Relationships are more important than element details
Versioning of Maps

- Current, Planned, and Target-States
- Create variants and merge
- Tracable history

- Compare multiple versions in a single visualization, e.g. overlay (different opacity)
Comparing Different Versions of an EA: A Brief Example

Legend
- Business Unit
- Business Unit (consolidated)
- A gets absorbed by B

Munich
- HR
- R&D
- Production
- IT

Berlin
- HR
- R&D
- Production
- IT

Frankfurt
- HR
- R&D
- Production
- IT

New York
- HR
- R&D
- Production
- IT

Munich
- HR
- R&D
- Production
- IT

Legend
- Business Unit
- Business Unit (consolidated)
- A gets absorbed by B
Challenges: Scalability

- Comparing multiple entities (more than two) and respective instances with each other
- Comparing structural information (relationships) over time
- Avoid Information overflow

- re-evaluation of concepts
- EA management visualizations (cf. e.g. [Bu08])
- information visualization (cf. e.g. [Tu01])
- Methods & principles of system cartography (cf. [Wi07, Ma08])
Challenges: Scoping vs. “Big Picture”

- Different levels of granularity to be analyzed
  - strongly depends on the actual stakeholder
  - a high-level overview sometimes is very beneficial
  - history (versions) of a particular business unit or business application could also be subject of interest for a deeper analysis ➔ semantic zoom?
Challenges: Layout algorithms

- Known layout algorithms aim at esthetical pleasing layouts
  - e.g. NFDH algorithm: decreases the overall height of boxes with respect to an aspect ratio
  - would close any open space (white-spots) that are created, e.g. during migration.
  - White-spots are actually useful to communicate (the amount of) change when analyzing EA evolutions.
Challenges: Generic vs. special purpose solutions

- Some solutions will be generic (information visualization)
- Others will be very specific for the domain of EA and EA management (similar to [Bu10a])

The applied method must be (at least partially) included in EA information model
Challenges: Communication of timespans

Visual feedback on the actual timespan (and time differences) shown is useful but sometimes hard to achieve.

Existing visualization approaches, e.g. [Tu01], may be helpful for this purpose.
MIT Timeline

http://www.simile-widgets.org/timeline/
Version control systems (Mercurial, GIT)
Further Questions

- What is the experience of the audience concerning Visual representation/communication of change
  - Techniques
  - Layouts and algorithms
  - Tool support

- Visual support for code evolution
  - Are tools utilized in practice? Why, why not?
  - e.g. Software in the city (Schreiber et al. SE 2013)
    - Comparing of binaries
  - Do 3D visualization help? (cf. e.g. [Te09])

- Visualization/Analysis of the Evolution of (distributed) Software Architectures?
- Which tools are used?
- Are differences of the information model relevant to practice?
Thank you for your attention.

Contact: Sascha Roth (roth@tum.de)
References


