# A Requirements Based Approach for Automating Enterprise IT Architecture Modeling Using Multiple Data Sources

Margus Välja, Robert Lagerström, Mathias Ekstedt, Matus Korman

**ModTools15**

# Outline

- Motivation
- Methodology
- Requirements and Approach
- Study
- Discussion

# Motivation

Too **complex environments** to create models manually

(hundreds of devices which each have tens of instances of software)

If we would succeed,

**Timeliness** and **scalability** still problematic

# Motivation

Too much heterogeneous data and difficult to make sense of it

# Motivation: Model

| Collected | Known | In progress |
|---|---|---|
| Application protocols | Known vulnerabilities in existing software | Configuration methods used for web applications and similar |
| Computer and network hardware with addresses | Patch levels of clients, servers and software products | IT management processes' characteristics like for example for zone management process |
| Network zones | Access control points and password authentication mechanisms | Social aspects like social zone, security awareness program, and developer training |
| Software (also firmware) including system software and operating systems | Data flows | Software architecture and software assurance methods like static code analysis |
| User accounts | | Types of security controls present like cryptography methods and port security |

# Motivation: Data sources

| Type of tool | Examples | Type of data | Data acquiring method |
|---|---|---|---|
| Active scanners | Vulnerability scanners, network scanners | Hardware devices, software, vulnerabilities | Scanning network, computer nodes and application servers |
| Passive scanners | Vulnerability scanners, network scanners, packet analyzers | Hardware devices, software, vulnerabilities, network communication | Listening to existing network traffic |
| Enterprise architecture management | Business, information, IT architecture | Models of organization and its IT (in different views) | Manual input, scanning |
| System management | Change, release, license management, directory services | TO-BE to AS-IS elements | Manual input, scanning |
| Security monitors | IDS, IPS, firewalls, SIEM solutions | System, network, process state information | Scanning, listening, registering security events |

# Methodology: Earlier work

- Model to model transformation standards Extensible Stylesheet Language Transformations (XSLT), Query/View/Transformation (QVT) etc.
- Attempts to create EA models (Archimate, CySeMoL)
- Business process modelling

# Methodology

- Requirements
  - EA model maintenance
  - Enterprise information credibility
  - Data cleaning (DW)

- IT operational model
  - EAAT & CySeMoL
  - XSLT

# Approach: Architectural requirements

1. Data collection process needs to be supported by tools to limit manual inspection.
2. Data collection process needs to be extensible to cover additional sources easily.

# Approach: Integration requirements

3. The system needs to be able to detect changes in real world enterprise architectures.
4. The system needs to provide a mechanism to define mapping from incoming data to the internal data structure.
5. The system needs to have a machine understandable internal structure.

# Approach: Data quality requirements

6. The system needs to provide mechanisms to ensure data quality that is sufficient for the modeling goals including the classification of data sources
7. The data needs to be of appropriate granularity, consistency, completeness and actuality (time), all of which must be measurable.
8. The system needs to allow for the automated propagation of changes.
9. The system needs to be able to identify and resolve data identity conflicts from different sources via reconciliation.

# Approach: Data quality requirements

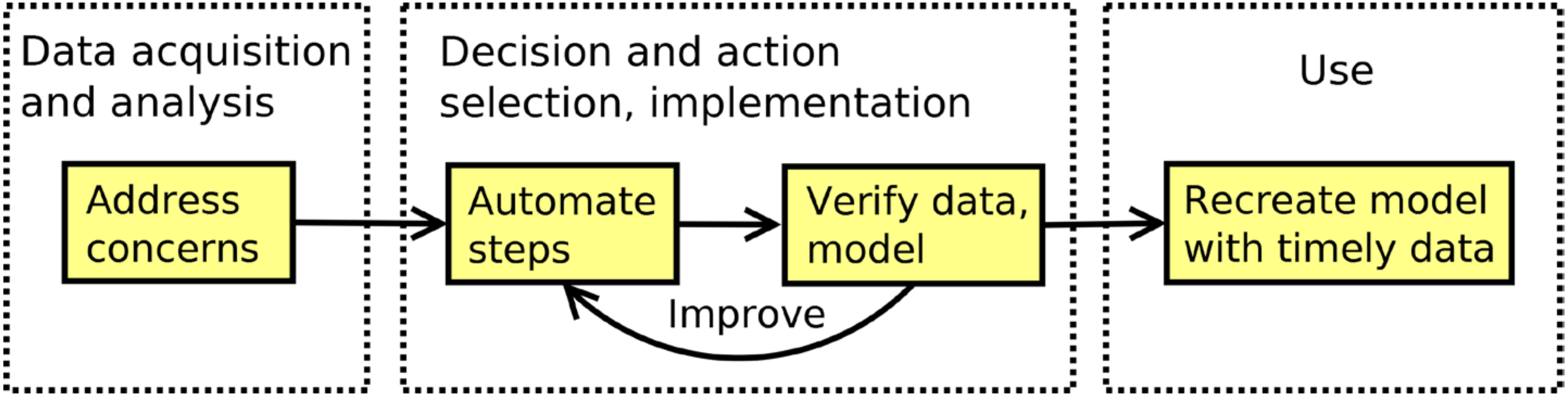
10. As more sources agree an answer, the credibility of that answer should be monotonically non-decreasing.

11. The data source must be known.
12. The data source’s historical credibility must be known.

# Approach: human-machine approach

| One source | Multiple sources | Description |
|---|---|---|
| Data acquisition | Data acquisition | Collecting data for a model |
| | Data analysis | |
| | Decision and action selection | |
| | Action Implementation | |
| Data analysis | Data analysis | Model based analysis |
| Decision and action selection | Decision and action selection | Model based action selection |
| Action Implementation | Action Implementation | Real life consequences |

# Approach: High level view



Data acquisition and analysis

Address concerns

Decision and action selection, implementation

Automate steps

Verify data, model

Improve

Use

Recreate model with timely data

# Approach: Concerns: Data acquisition

| | Concern | Process step | Automation |
|---|---|---|---|
| 1 | What data are needed for modeling? | Define data needs based on the metamodel as a common data structure | No |
| 2 | Where do the data come from? | Define data sources | No |
| 3 | What kind of tools and machine understandable structure should be used? | Decide tools and machine understandable format | No |
| 4 | Which data sources can be trusted and to what extent? | Define credibility calculation rules | Partial |
| 5 | How are the data acquired? | Create adapters to common data structure | Yes |

# Approach: Concerns: Data analysis

| | Concern | Process step | Automation |
|---|---|---|---|
| 6 | Are there any data quality problems and can these be solved? | Identify common data quality problems and define known techniques to solve them (data cleaning workflow and mapping rules) | Yes |
| 7 | Which data are still missing from the model and can missing data be derived from existing data? | Identify missing data and define rules for deriving missing data | Partial |
| 8 | Are there any patterns that can be reused to improve future models? | Define patterns and reuse them | Yes |
| 9 | How can the processed data be transformed into a model? | Apply a transformation method | Yes |

# Empirical study

- SCADA lab
  - 5 servers
  - 2 Red Hat Systems
  - 3 Windows

- Security analysis
  - EAAT and CySeMoL
  - Automatic model generation

# Empirical study: Addressing concerns

- Data sources Nexpose and Wireshark
- Nexpose prioritized over Wireshark
- Data exported to XML format and abstracted to right level with adapters

# Empirical study: Addressing concerns

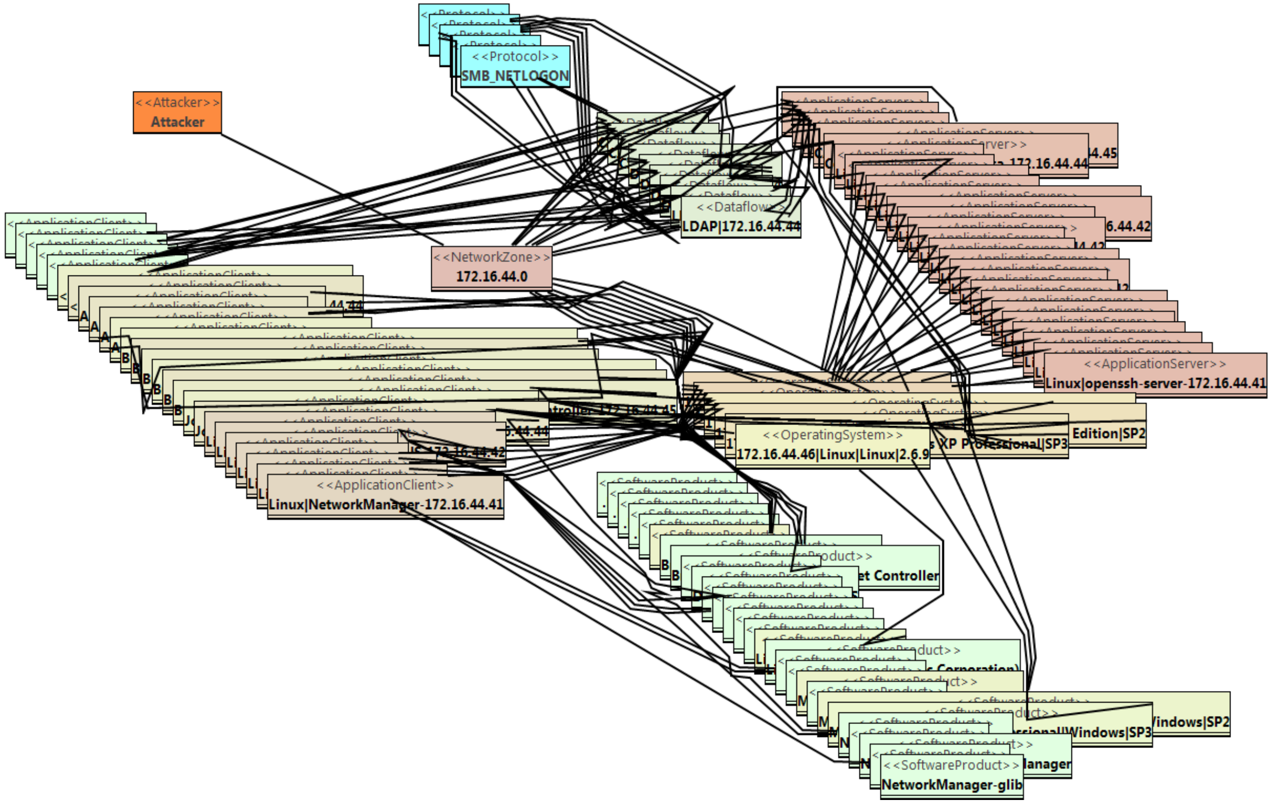
- Common data structure represents CySeMoL  ontology, data quality attributes
- Unique identifiers chosen like IP address to merge data
- Outside dictionaries needed to complement data
- Some data needs to be omitted from the model
- EAAT transformation through XLST

# Results

Calculable model with more than 10000 elements

Process can be repeated with new data in minutes

# Results: Model



# Limitations

- Implementation specific details missing
- Only certain type of models supported
- Manual implementation work

# Conclusion

- Model creation with multiple sources automated
- 9 Concerns need to be addressed

# Questions

?