Automated Detection of Architecture Patterns in MBSE Models

Matt Cotter (mjcotter@mitre.org)

Michael Hadjimichael (mikeh@mitre.org)

Aleksandra Markina-Khusid (amk@mitre.org)

Brian York (byork@mitre.org)

October 28-29, 2020

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

Introduction

- **The evaluation of a system's architecture is an iterative, essential process within the systems engineering lifecycle, BUT…**

  1. It is difficult to thoroughly understand, assess and evaluate a system's architecture in a consistent and objective manner
     - Even when a system architecture is formally defined using commercially available Model Based Systems Engineering (MBSE) methodologies/languages/tools
  2. Non-functional architecture qualities, -ilities, are difficult to define, and quantify, in meaningful ways across domains

- **This presentation provides an overview for a novel, automated, repeatable method for detecting patterns-of-interest within an MBSE model, to be exposed to help assist the systems engineer in the evaluation and improvement of said architecture**

Challenge: SME-Driven/program-specific/qualitative architecture analysis

Mitigate with a combination of MBSE, design pattern literature, natural language processing, and graph theory

Some Definitions

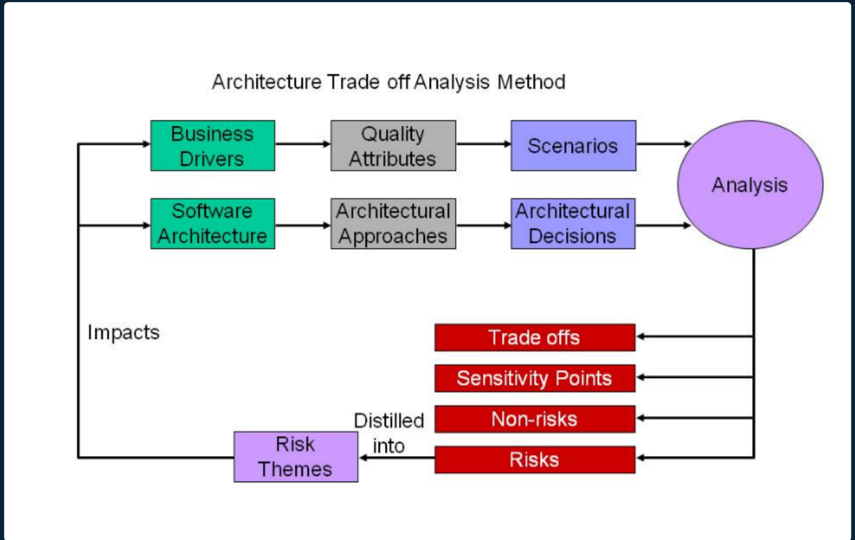
Architecture Evaluation (or Assessment): A series of activities that enable systems engineers to better understand, quantify and improve the potential for said architecture to deliver a system capable of fulfilling stakeholder requirements.

- Not just an assessment of “goodness” or “badness”, but an assessment of whether the architecture fits the needs of the customer or aligns with some desired non-functional properties or requirements (e.g., –ilities).
- Most architecture evaluation activities typically use scenarios to characterize quality attributes, which can be rolled up to evaluate an architecture

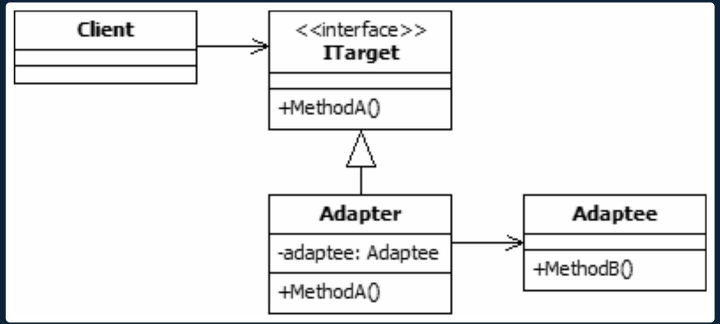
Architecture (or Design) Pattern: A generalized solution to one or more common problems within a given context or domain.

- Pattern application may promote particular -ilities within such as modularity and scalability
- Provides an evaluation team with a concrete set of architectural features that align with a specific capability or requirement

Architecture (or Design) Anti-Pattern: A common response to one or more common problems that is usually ineffective and introduces risk into the system



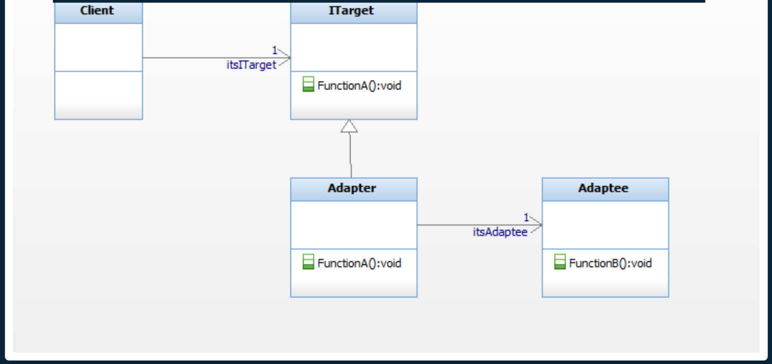
Example of an architecture evaluation method: Architecture Tradeoff Analysis Method (ATAM)



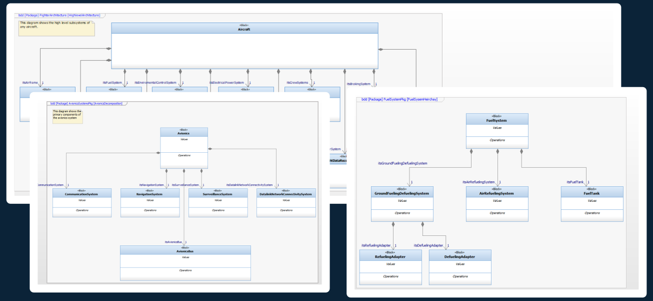
Example of an Adapter Pattern written using Unified Modeling Language (UML) Notation

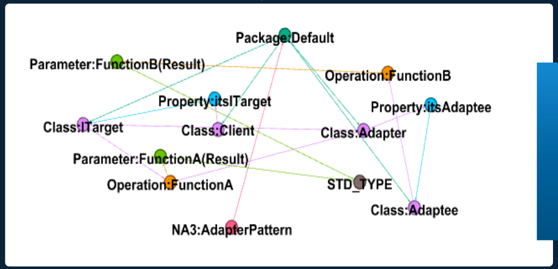
Conceptual Approach

Architecture Pattern Template



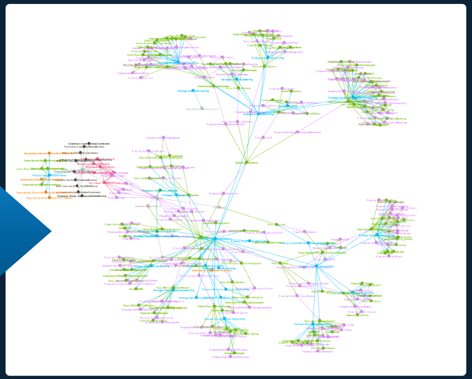
System Architecture





Graph Representation

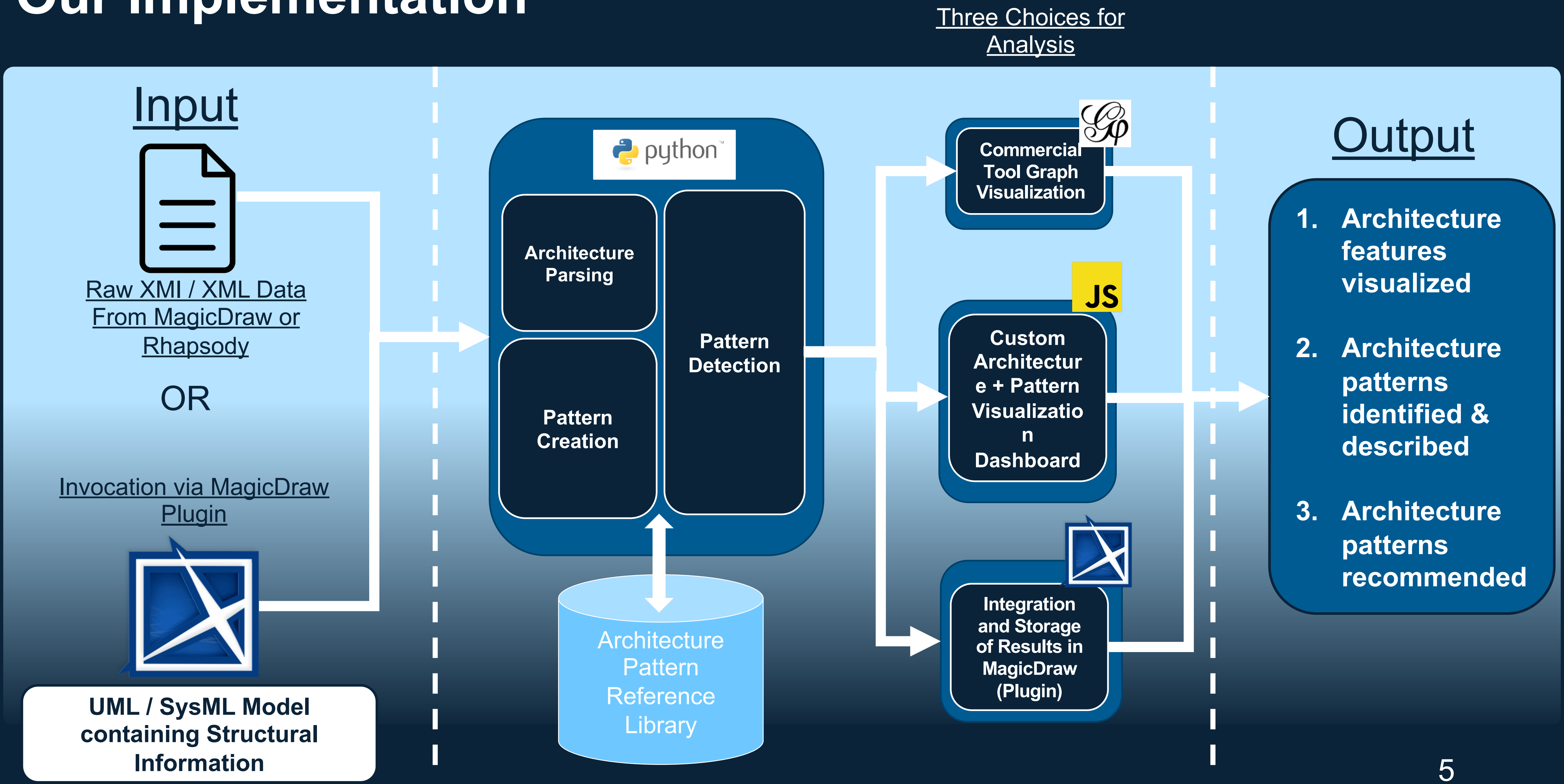
Find full or partial instances of the pattern within the system to assist in evaluation activities



Graph Representation

Not a replacement for existing architecture evaluation methods, but an additional tool in the SE toolbelt

Our Implementation



Three Choices for Analysis

Input

Raw XMI / XML Data From MagicDraw or Rhapsody

OR

Invocation via MagicDraw Plugin

UML / SysML Model containing Structural Information

python

Architecture Parsing

Pattern Creation

Pattern Detection

Architecture Pattern Reference Library

Commercial Tool Graph Visualization

Custom Architectur e + Pattern Visualizatio n Dashboard

Integration and Storage of Results in MagicDraw (Plugin)

Output

1. **Architecture features visualized**
2. **Architecture patterns identified & described**
3. **Architecture patterns recommended**

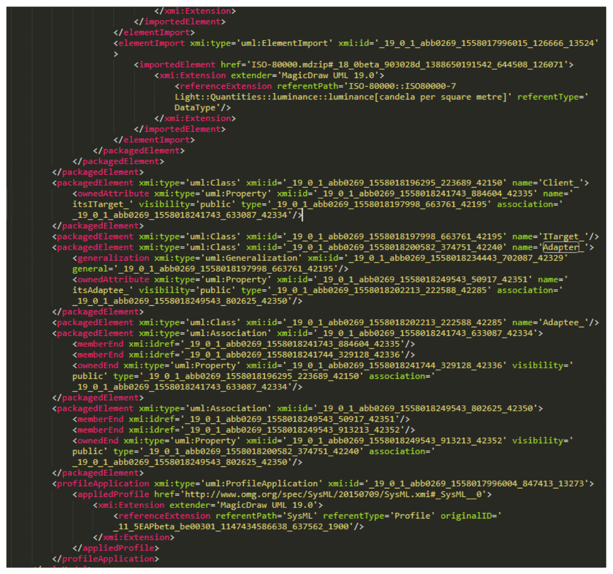
Architecture Parsing

- Defined as the process of translating text-based architecture data that has been generated from a commercially available MBSE tool into a graph-based data structure whose nose and edges correspond to a subset of elements contained within the original MBSE model

- Utilizes the ability for commercial tools to output text-based data that complies with the XML Metadata Interchange (XMI) standard, as defined by the Object Management Group (OMG).

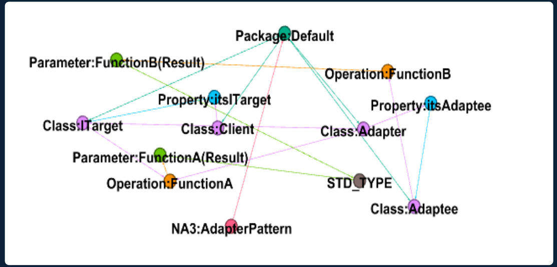
- Initial work focused primarily on *structural* SysML elements instead of *behavioral* elements

- Separate prototypes required for Rhapsody and MagicDraw



```
                    </xmi:Extension>
                </importedElement>
            </elementImport>
            <elementImport xmi:type='uml:ElementImport' xmi:id='_19_0_1_abb0269_1558017996015_126666_13524'
            >
                <importedElement href='ISO-80000.mdzip#_18_0beta_903028d_1388650191542_644508_126071'>
                    <xmi:Extension extender='MagicDraw UML 19.0'>
                        <referenceExtension referentPath='ISO-80000::ISO80000-7
                        Light::Quantities::luminance::luminance[candela per square metre]' referentType='
                        DataType'/>
                    </xmi:Extension>
                </importedElement>
            </elementImport>
        </packagedElement>
    </packagedElement>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_19_0_1_abb0269_1558018196295_223689_42150' name='Client_'>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_19_0_1_abb0269_1558018241743_884604_42335' name='
    itsITarget_' visibility='public' type='_19_0_1_abb0269_1558018197998_663761_42195' association='
    _19_0_1_abb0269_1558018241743_633087_42334'/>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_19_0_1_abb0269_1558018197998_663761_42195' name='ITarget_'/>
<packagedElement xmi:type='uml:Class' xmi:id='_19_0_1_abb0269_1558018200582_374751_42240' name='Adapter_'>
    <generalization xmi:type='uml:Generalization' xmi:id='_19_0_1_abb0269_1558018234443_702087_42329'
    general='_19_0_1_abb0269_1558018197998_663761_42195'/>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_19_0_1_abb0269_1558018249543_50917_42351' name='
    itsAdaptee_' visibility='public' type='_19_0_1_abb0269_1558018202213_222588_42285' association='
    _19_0_1_abb0269_1558018249543_802625_42350'/>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_19_0_1_abb0269_1558018202213_222588_42285' name='Adaptee_'/>
<packagedElement xmi:type='uml:Association' xmi:id='_19_0_1_abb0269_1558018241743_633087_42334'>
    <memberEnd xmi:idref='_19_0_1_abb0269_1558018241743_884604_42335'/>
    <memberEnd xmi:idref='_19_0_1_abb0269_1558018241744_329128_42336'/>
    <ownedEnd xmi:type='uml:Property' xmi:id='_19_0_1_abb0269_1558018241744_329128_42336' visibility='
    public' type='_19_0_1_abb0269_1558018196295_223689_42150' association='
    _19_0_1_abb0269_1558018241743_633087_42334'/>
</packagedElement>
<packagedElement xmi:type='uml:Association' xmi:id='_19_0_1_abb0269_1558018249543_802625_42350'>
    <memberEnd xmi:idref='_19_0_1_abb0269_1558018249543_50917_42351'/>
    <memberEnd xmi:idref='_19_0_1_abb0269_1558018249543_913213_42352'/>
    <ownedEnd xmi:type='uml:Property' xmi:id='_19_0_1_abb0269_1558018249543_913213_42352' visibility='
    public' type='_19_0_1_abb0269_1558018200582_374751_42240' association='
    _19_0_1_abb0269_1558018249543_802625_42350'/>
</packagedElement>
<profileApplication xmi:type='uml:ProfileApplication' xmi:id='_19_0_1_abb0269_1558017996004_847413_13273'>
    <appliedProfile href='http://www.omg.org/spec/SysML/20150709/SysML.xmi#_SysML__0'>
        <xmi:Extension extender='MagicDraw UML 19.0'>
            <referenceExtension referentPath='SysML' referentType='Profile' originalID='
            _11_5EAPbeta_be00301_1147434586638_637562_1900'/>
        </xmi:Extension>
    </appliedProfile>
</profileApplication>
```

XMI Version of Adapter Pattern In MagicDraw



Graph Representation

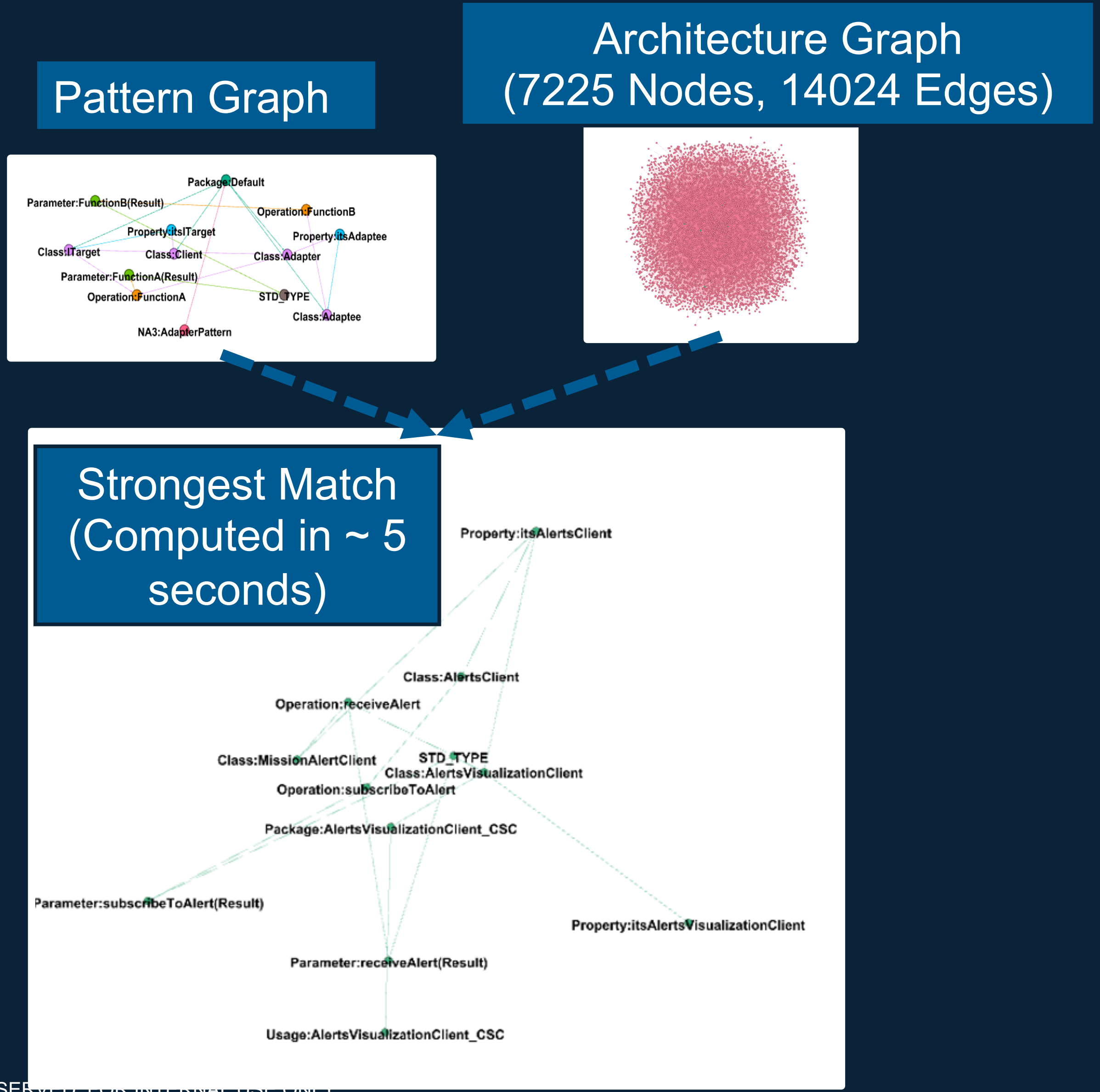
Detecting Patterns

- **The pattern detection module generates similarities scores comparing nodes and edges in the pattern graph (P) to one or more elements of the architecture graph (A)**

- **Similarity scores are generated by looking at both:**
  1) Similarities in natural language
  2) Similarly typed structures
     - Based on UML2 Metamodel

- **Returns a list of nodes in A such that there is a potential match to a node in P.**
  – Highlights what is there, **and what is missing**

- **Additional notes:**
  – Can return N strongest results, i.e. instances of patterns
  – Each pattern is associated with a set of “synonyms” that are used to help filter the search and dramatically reduce computational cost
  – All nodes in A begin with a similarity score of 0.
  – All nodes in A are never <u>deducted</u> points for missing a mapping to a pattern, they simply receive a worse score.



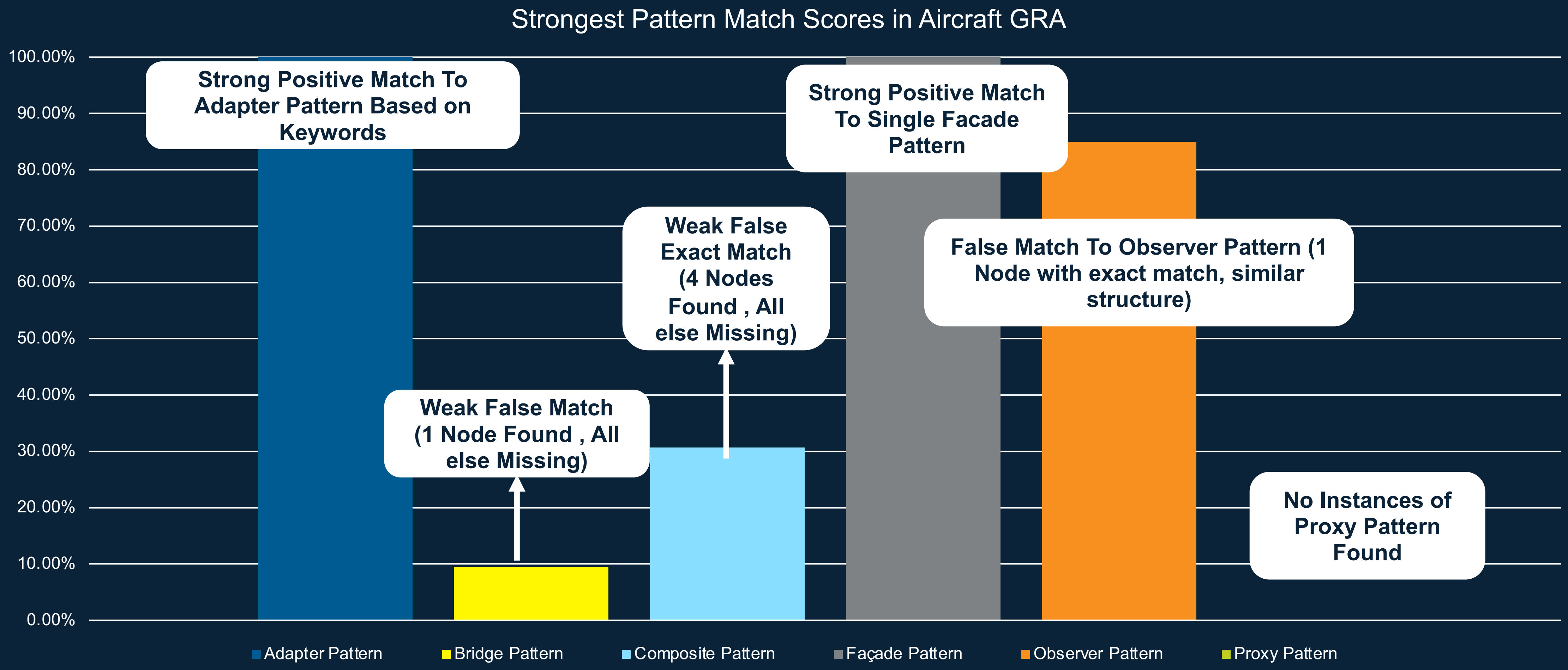
Pattern Graph

Architecture Graph (7225 Nodes, 14024 Edges)

Strongest Match (Computed in ~ 5 seconds)

Package:Default
Parameter:FunctionB(Result)
Operation:FunctionB
Property:itsITarget
Property:itsAdaptee
Class:ITarget
Class:Client
Class:Adapter
Parameter:FunctionA(Result)
Operation:FunctionA
STD_TYPE
Class:Adaptee
NA3:AdapterPattern
Property:itsAlertsClient
Class:AlertsClient
Operation:receiveAlert
Class:MissionAlertClient
STD_TYPE
Class:AlertsVisualizationClient
Operation:subscribeToAlert
Package:AlertsVisualizationClient_CSC
Parameter:subscribeToAlert(Result)
Property:itsAlertsVisualizationClient
Parameter:receiveAlert(Result)
Usage:AlertsVisualizationClient_CSC

Results From Application to Mature Aircraft Government Reference Architecture (GRA)



Strong Positive Match To Adapter Pattern Based on Keywords

Weak False Match (1 Node Found , All else Missing)

Weak False Exact Match (4 Nodes Found , All else Missing)

Strong Positive Match To Single Facade Pattern

False Match To Observer Pattern (1 Node with exact match, similar structure)

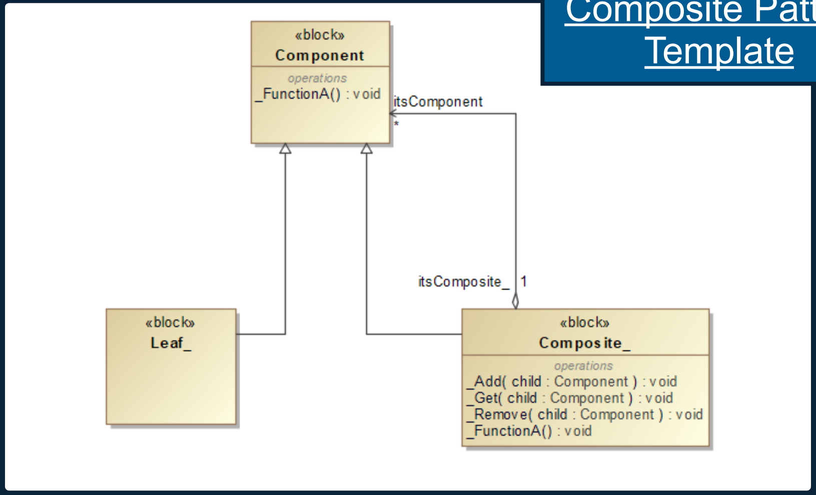
No Instances of Proxy Pattern Found

More Applications and Discussion

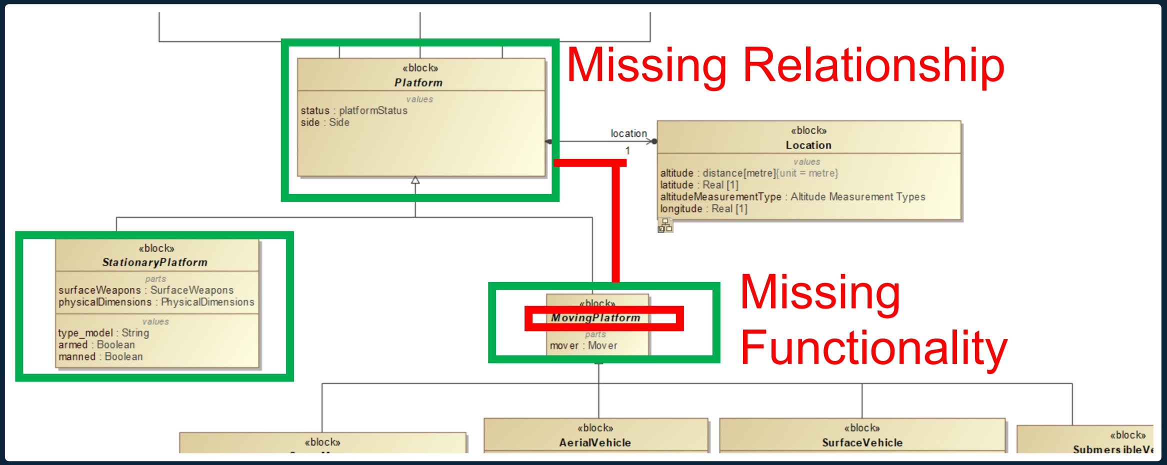
- Pattern detection results when applied to MITRE developed architecture models
  1. Generic Command and Control (C2)
     - Partial matches to composite pattern, recommend architecture modification
  2. Search and Rescue
     - Partial matches to adapter pattern, did not recommend architecture modification
  3. Joint Close Air Support (JCAS) System of Systems
     - No strong matches to any pattern in existing library (and that is OK!)

- General Lessons Learned Takeaways:
  - Two knobs to play with: similarity sensitivity and N returned results
  - Visual Inspection still required to interpret the results and make decision on whether or not to recommend a change
  - Not all system models leverage diagram types we look use in detection… future research may want to look at more of UML / SysML diagram types beyond the BDD.

Composite Pattern Template



Generic C2 Architecture (Highlighted Results)



Missing Relationship

Missing Functionality

Observable Benefits

Natural-Language based similarity scoring dramatically increases computational efficiency and widens the net for results

As long as you can describe a pattern, (or anti-pattern) in UML / SysML structural elements, this method is readily applicable.

Ability to handle complex, heterogenous directed graphs, with attributes. State-of-Practice Graph Classification and Link Prediction publications often with homogenous graphs and no attributes.

No Tool Plugins Required, but available for users that are comfortable

Working in a format that is easy to ingest and visualize for broader analysis capabilities.

Observable Challenges

Model size can strain commercial tool XMI generation

Full GRA Export to XMI ~ 20 min

Other large models cause application to CRASH.

Pattern detection is limited by:

1. The modeling language’s ability to describe multiplicities or optional content for elements of a pattern
   - “zero or more” and “one or more of” needed to be annotated manually on pattern templates
2. Our sensitivity to labels in an architecture

Architecture Patterns and Anti-Patterns are not always:

Clearly or consistently defined in literature

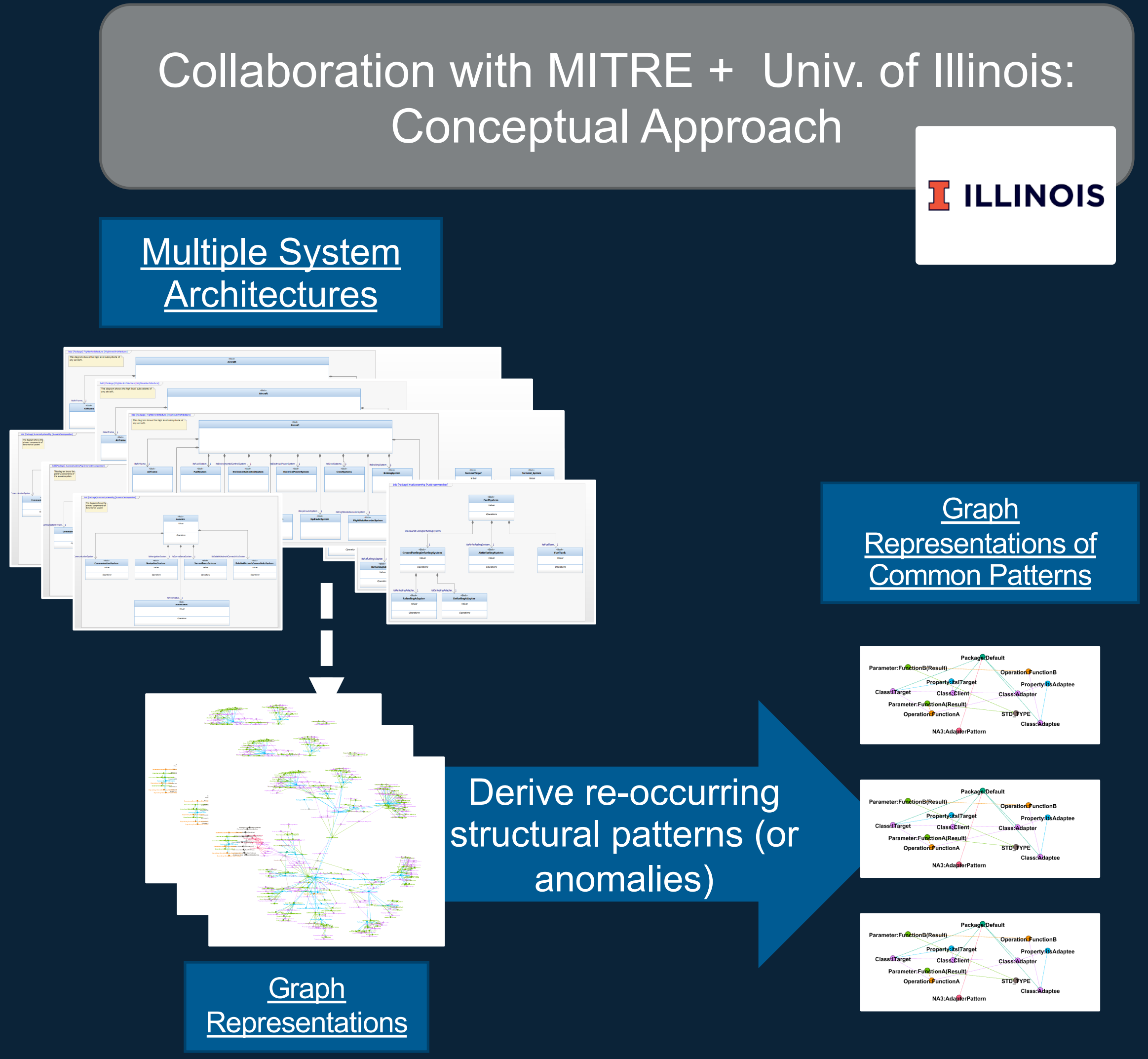
Well-understood or documented for a given domain or context

Conclusions

Able to successfully demonstrate the detection of patterns-of-interest in system architecture models built in several commercially available MBSE tools

Python-based pattern detection and architecture parsing being prepared for open-source release outside of MITRE

MagicDraw Plugin version of the method will continue to be developed and applied



Collaboration with MITRE + Univ. of Illinois: Conceptual Approach

Multiple System Architectures

Graph Representations

Derive re-occurring structural patterns (or anomalies)

Graph Representations of Common Patterns

Thank-You!