Extracting Systems Architecture from Non-Standard Sources (ArchEx)

Mike Hadjimichael
Matt Cotter
Mitch Kinney
Monica Carley-Spencer
Aleksandra Markina-Khusid

Ingest informally specified systems architectures from images, automatically extract the represented architecture, and output Systems Modeling Language (SysML) standard diagrams within the industry-standard MBSE tools.

Oct 21 2021

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

Problem

- **Model-Based Systems Engineering (MBSE) methods, processes and tools are a key enabler for rigorous, complete and cohesive system design, with many additional benefits as well.**

- **A large percentage of programs, both MITRE and sponsors, have not yet adopted MBSE technologies.**

- **Those projects that _do_ adopt MBSE technologies, always must go through a "transition period", converting some portion of existing documentation into one or more system models.**
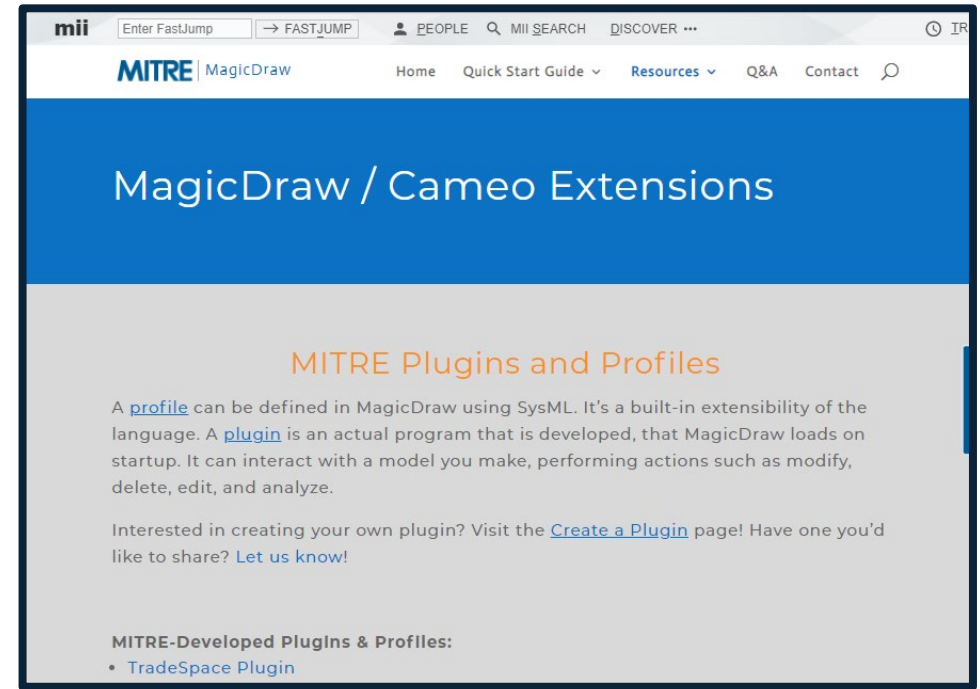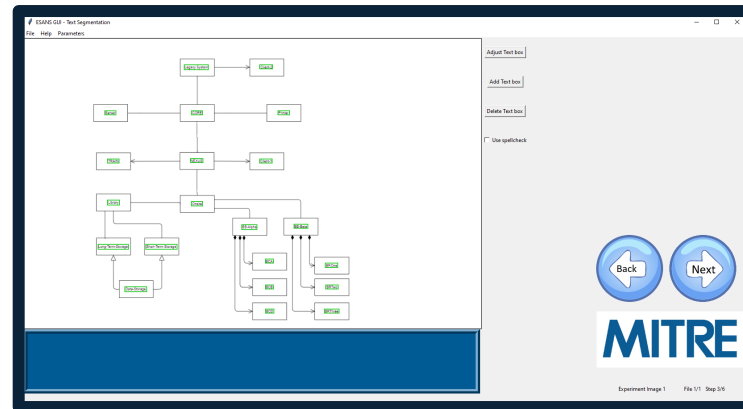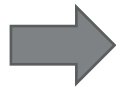

Emerging MBSE extensions and capabilities, which <u>could</u> provide value to a project, are locked behind MBSE transition!
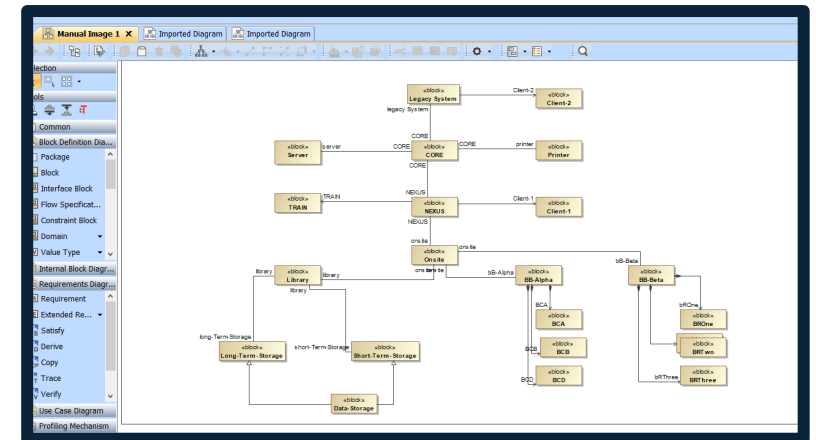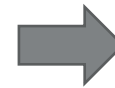
Idea

We proposed a tool to convert non-standard systems architecture artifacts into standard MBSE architecture content. Lowering the bar for MBSE implementors.

"Virtual" Paper-based system documentation
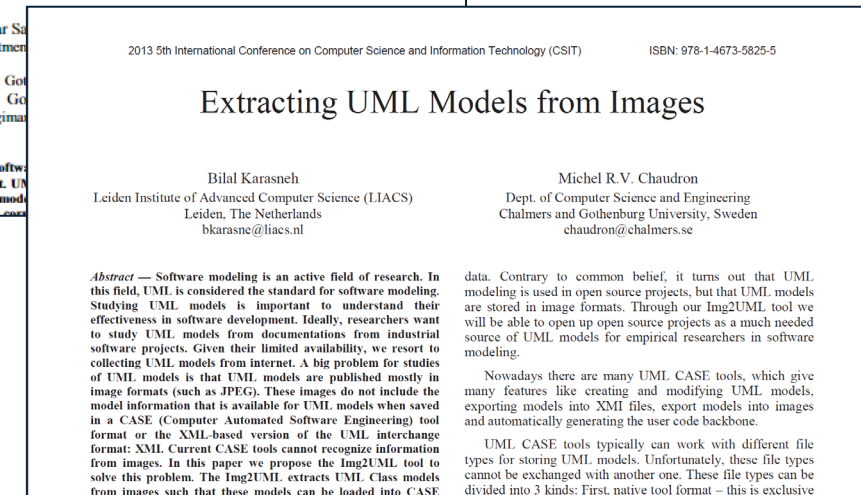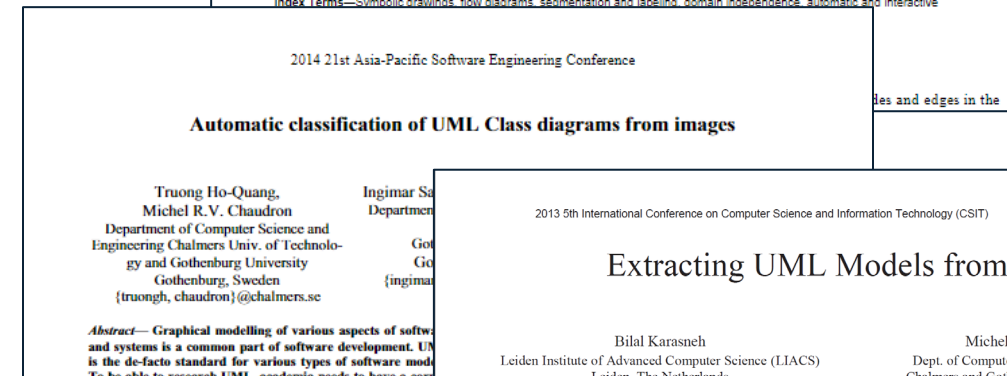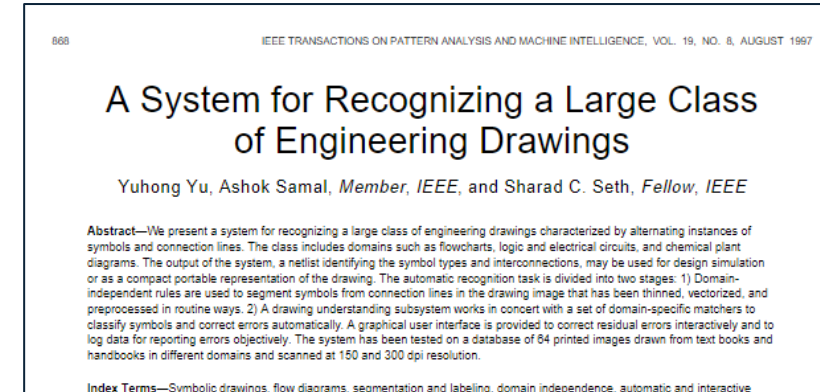
AI-Assisted Architecture Parsing

Output & Integration into a formal digital model

There **is** room for automation!

MITRE

Background / Literature Review

- **Nearly all reference and follow processes described by Yu et. al, 1997 (Top Right)**

- **One group, led by Professor Michel Chaudron, (Bottom Right) has publications that:**

  - Extract class diagram data from images

  - Classify an image as a class diagram

  - Sought to create an open-source repository of class diagrams for future experimentation

- **Yet:**

  - **No publications exist for SysML translation across diagrams other than the** <u>class</u> **diagram**

  - **Unresolved CV challenges (e.g., diagonal connections, informal diagrams)**

- **New technologies from machine learning perspective stand to dramatically impact the effectiveness of any future prototype.**

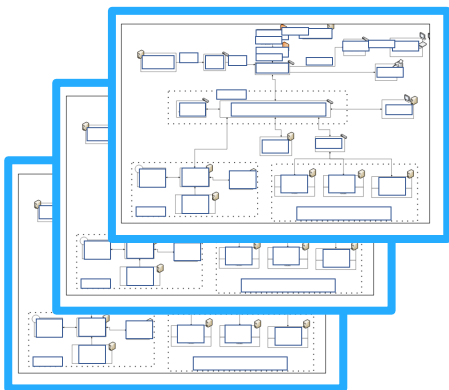868   IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 8, AUGUST 1997

A System for Recognizing a Large Class of Engineering Drawings

Yuhong Yu, Ashok Samal, *Member, IEEE*, and Sharad C. Seth, *Fellow, IEEE*

Abstract—We present a system for recognizing a large class of engineering drawings characterized by alternating instances of symbols and connection lines. The class includes domains such as flowcharts, logic and electrical circuits, and chemical plant diagrams. The output of the system, a netlist identifying the symbol types and interconnections, may be used for design simulation or as a compact portable representation of the drawing. The automatic recognition task is divided into two stages: 1) Domain-independent rules are used to segment symbols from connection lines in the drawing image that has been thinned, vectorized, and preprocessed in routine ways. 2) A drawing understanding subsystem works in concert with a set of domain-specific matchers to classify symbols and correct errors automatically. A graphical user interface is provided to correct residual errors interactively and to log data for reporting errors objectively. The system has been tested on a database of 64 printed images drawn from text books and handbooks in different domains and scanned at 150 and 300 dpi resolution.

Index Terms—Symbolic drawings, flow diagrams, segmentation and labeling, domain independence, automatic and interactive

2014 21st Asia-Pacific Software Engineering Conference

Automatic classification of UML Class diagrams from images

Truong Ho-Quang, Michel R.V. Chaudron
Department of Computer Science and Engineering Chalmers Univ. of Technology and Gothenburg University
Gothenburg, Sweden
{truongh, chaudron}@chalmers.se

Ingimar Sa
Department

Got
Go
{ingimar

Abstract— Graphical modelling of various aspects of softwa and systems is a common part of software development. U is the de-facto standard for various types of software mod

2013 5th International Conference on Computer Science and Information Technology (CSIT)   ISBN: 978-1-4673-5825-5

Extracting UML Models from Images

Bilal Karasneh
Leiden Institute of Advanced Computer Science (LIACS)
Leiden, The Netherlands
bkarasne@liacs.nl

Michel R.V. Chaudron
Dept. of Computer Science and Engineering
Chalmers and Gothenburg University, Sweden
chaudron@chalmers.se

Abstract — Software modeling is an active field of research. In this field, UML is considered the standard for software modeling. Studying UML models is important to understand their effectiveness in software development. Ideally, researchers want to study UML models from documentations from industrial software projects. Given their limited availability, we resort to collecting UML models from internet. A big problem for studies of UML models is that UML models are published mostly in image formats (such as JPEG). These images do not include the model information that is available for UML models when saved in a CASE (Computer Automated Software Engineering) tool format or the XML-based version of the UML interchange format: XML. Current CASE tools cannot recognize information from images. In this paper we propose the Img2UML tool to solve this problem. The Img2UML extracts UML Class models from images such that these models can be loaded into CASE

data. Contrary to common belief, it turns out that UML modeling is used in open source projects, but that UML models are stored in image formats. Through our Img2UML tool we will be able to open up open source projects as a much needed source of UML models for empirical researchers in software modeling.

Nowadays there are many UML CASE tools, which give many features like creating and modifying UML models, exporting models into XMI files, export models into images and automatically generating the user code backbone.

UML CASE tools typically can work with different file types for storing UML models. Unfortunately, these file types cannot be exchanged with another one. These file types can be divided into 3 kinds: First, native tool format – this is exclusive

MITRE

Technical Approach

Systems Engineer

Legend

🟦	Data File
🟧	ArchEx Parser (Python)
🟩	ArchEx Cameo Plugin

Architecture Images (.png/.jpeg/etc.)

#1. Configure + Review Architecture Parsing

Graph of Derived Content (.graphML)

#2. Import Derived Content And Review

Digital Model Content (.mdzip)

Functionality:

- Text Segmentation
- Shape Segmentation
- Line Segmentation
- Line Classification

Functionality:

- Model Content Creation
- Model Diagram Creation
- Model Diagram Organization
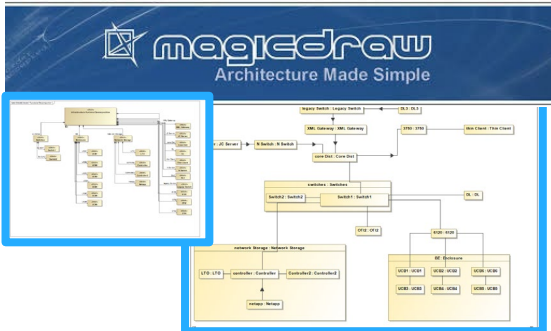
Progress to Date

Two Primary Artifacts:

1) ***Architecture Extractor (ArchEx) Parser***

   - *Python-based software application*

   - *Extracts content from static images of architectures*

2) ***ArchEx Plugin for Cameo***

   - *Java-based software application*

   - *Loosely Coupled with ArchEx*

   - *Extends commercial tool, MagicDraw*
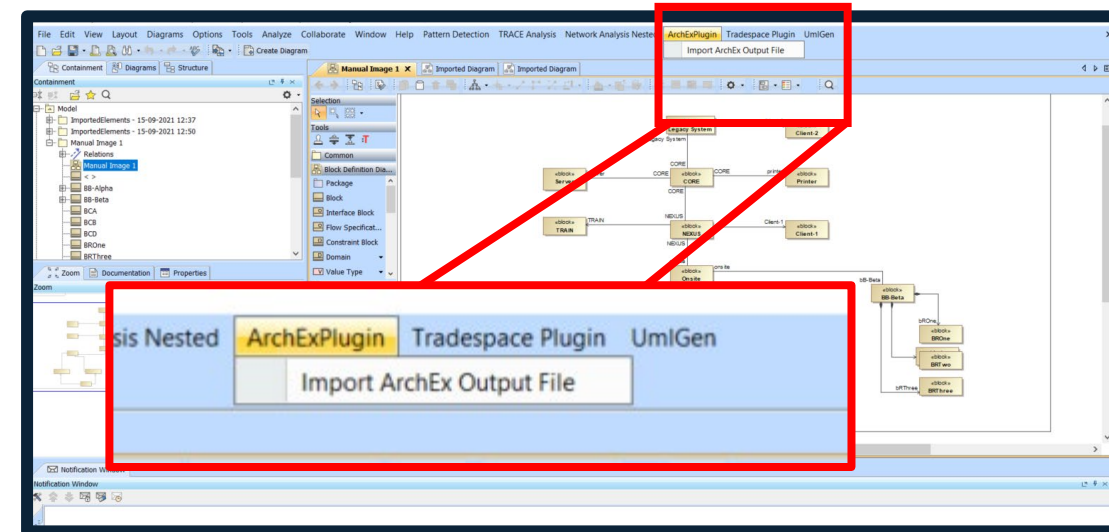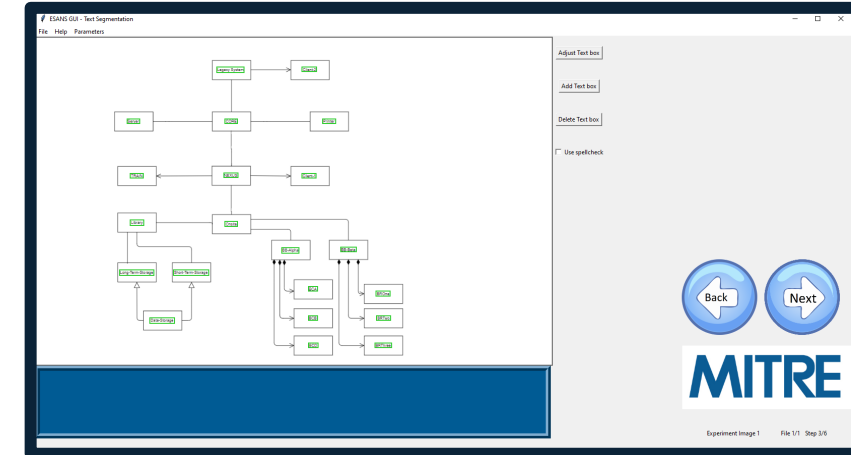
Progress To Date: Experiment: Useability, efficiency

- **Pilot experiment w/ 2 participants has been run**

- **Full experiment to follow w/ 16 participants – 4 subsets of 4 participants each for purpose of ordering assigned test images**

  - Each subset group will be assigned a unique ordering of the 4 images, while each participant within the group will have a different combination of assignments w/ respect to the method they will use to extract each diagram – manually or using the ArchEx tool

- **Participants are asked to complete a survey immediately following completion of the diagram extraction task. The survey has two sets of questions:**

  - the NASA Task Load Index rating scale, *and*

  - a set of questions formulated to elicit specific areas of improvement for ArchEx
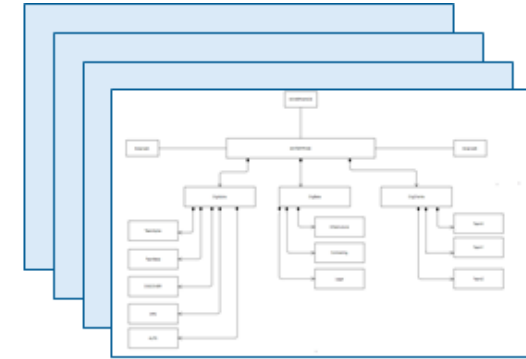
Participants asked to extract diagrams from each of 4 images

A	B	D	C
B	C	A	D
C	D	B	A
D	A	C	B

A balanced Latin Square design was used to assign the order of images to each subgroup

How successful were you in accomplishing what you were asked to do?

Very Low ... Very High

MITRE

Demo

MITRE
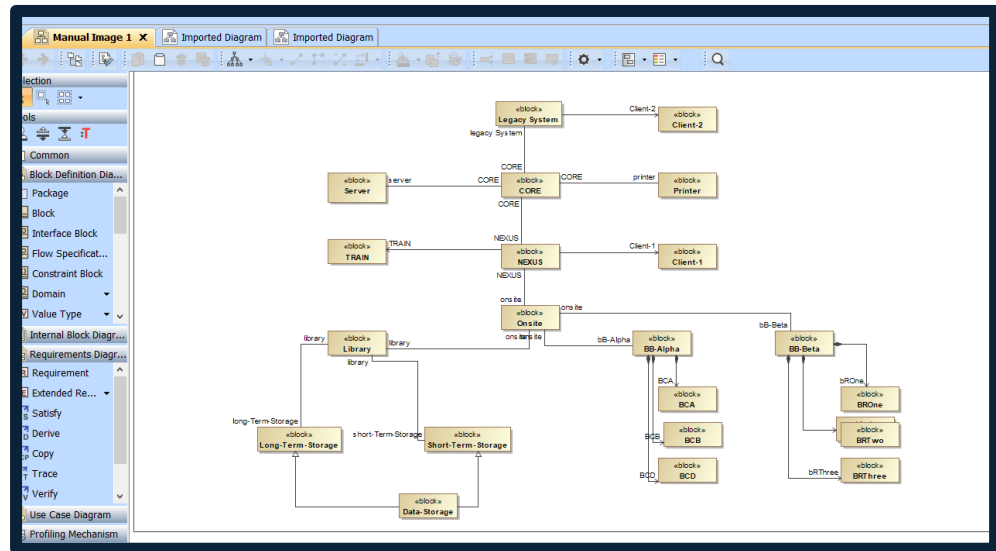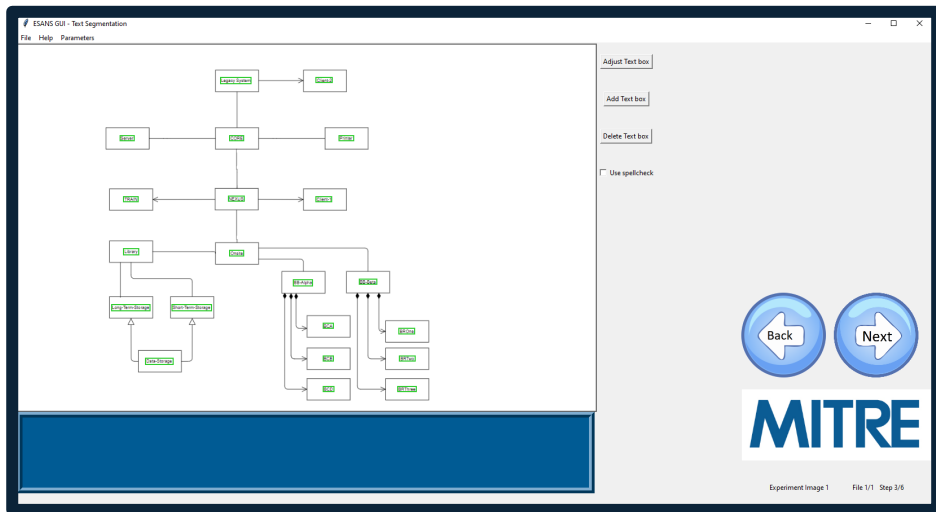
Lessons Learned

1. As the prototype matured and we continue to develop against sponsor data, we've realized an Input Image has the following <u>attributes</u> that capture the complexity of the problem and help us quantify our capability:

    - Standardization (Artistic vs. Expressed formally)

    - Quality (Resolution, content clarity)

    - Engineering Domain (Software, Hardware, Mechanical, Electrical, Networking, Database)

    - Diagram Type (Structural vs. Behavioral vs. Other)

2. A degree of human interaction is required to ensure the quality of the content as the ArchEx parser extracts image information

MITRE

Advocate Feedback and Support

"- For legacy weapon systems, the biggest obstacle to adoption of MBSE is return-on-investment. MBSE has to show it will provide more benefit to a system than it costs in terms of time, talent, and treasure. ==Your research effort is helping to tilt the balance in favor of adoption, with the promise of reductions in time to convert thousands of pages of legacy weapon system architecture products into SySML for modern systems engineering analysis.==

- I appreciate your candor that this will never replace a Human SySML modeler. However, it's heartening that ==this project could enable a modeling team to work much faster, with the tool enabling MMIII MBSE efforts. At the end of the day, it promises to let the modelers focus on architecting,== with significantly less effort in rote copying of legacy documents."

YOUNGMEYER, MATTHEW J Lt Col USAF AFMC AFNWC/NME

MITRE

Continuation (FY22) & Wrap-up

Technical

- **Complete experiment**

- **Improve user workflow**

- **Automated error checking**

- **Extended set of parse-able diagram types and domains**

- **Upgrade GUI**

Transition & Exposure

- **Investigate licensing options & internal MITRE hosting**

- **Conduct MITRE-internal "Road Show"**

- **Conference presentations (2 Accepted)**

- **Pursue journal article publication**

We are continuously looking for:

1. Architecture artifacts that you or your project would be able to share

2. Projects that are transitioning legacy data to a tool like Cameo Enterprise Architecture

3. Feedback, criticisms, and any information about YOUR pain points that would help ensure our prototype provides real value to MITRE projects and our sponsors

MITRE

Backup

MITRE

Attribute #1: Standardization

- Diagram standardization reflects how closely any given image already conforms to a _formal_ modeling language, such as SysML

- The more standardized an image is, the easier the Computer Vision (CV) task becomes (because of the known set of shapes)

- The easier the CV task, the higher the likelihood of us being able to reliably parse said image in a meaningful, accurate way.
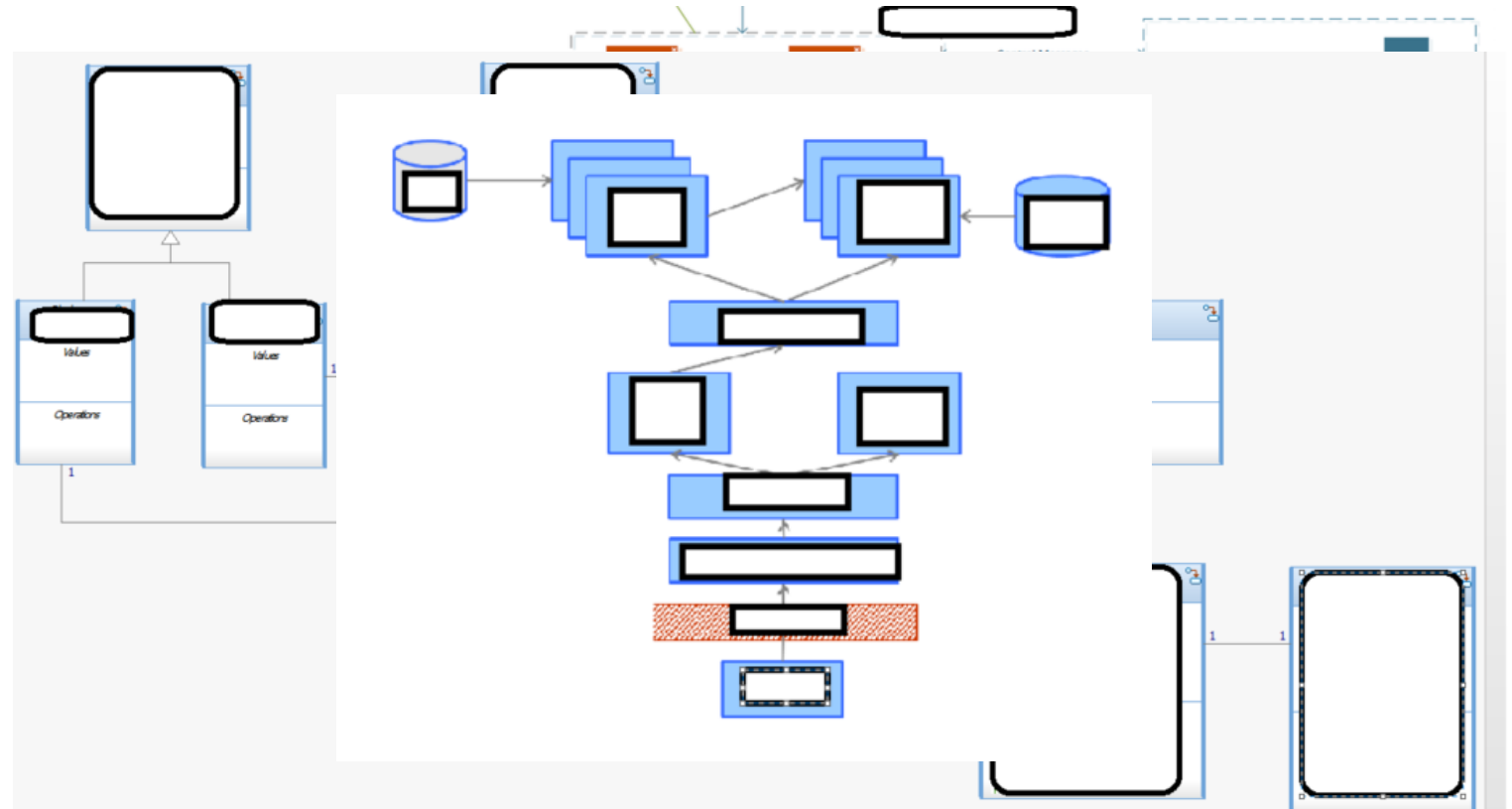
Level 1: UML or SysML Formatted (Training Purposes Only)
Level 2: Informal UML or SysML Notation / Simple Shapes
Level 3: Minimal or No UML or SysML Notation / Complex Shapes

MITRE

Attribute #2: Quality

Quality reflects how well we will be able to detect distinct elements in the image.
This attribute has two sub-measures:

1) Image Resolution: The level of detail an image holds

Low Resolution (Harder Problem) ⟵⟶ High Resolution (Easier Problem)
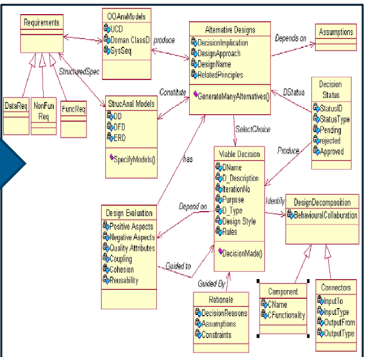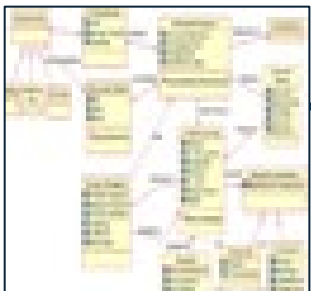
2) Content Clarity: The level of overall clear organization between elements in an image

Low Clarity (Harder Problem) ⟵⟶ High Clarity (Easier Problem)

MITRE
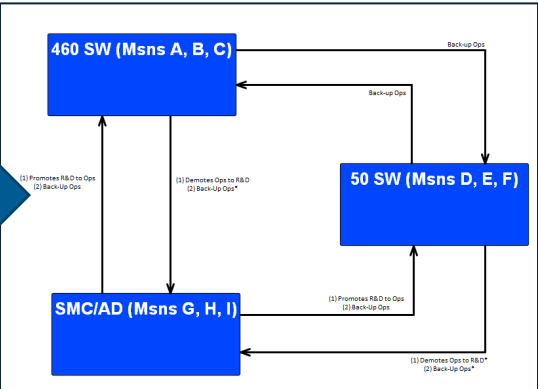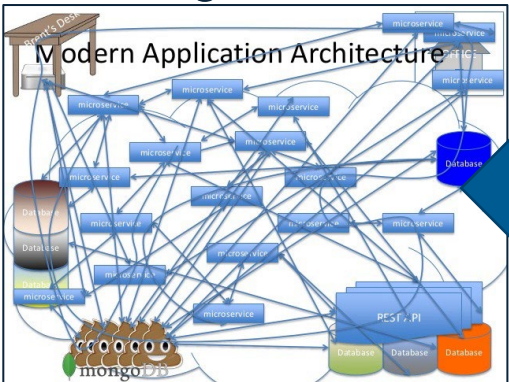
Attribute #3: Domain

An Image's Domain reflects the engineering discipline (or task) closest aligned with the intent of the image, and implicitly comes with a set of shapes and features we may (or should) expect to see. **Some examples include**:
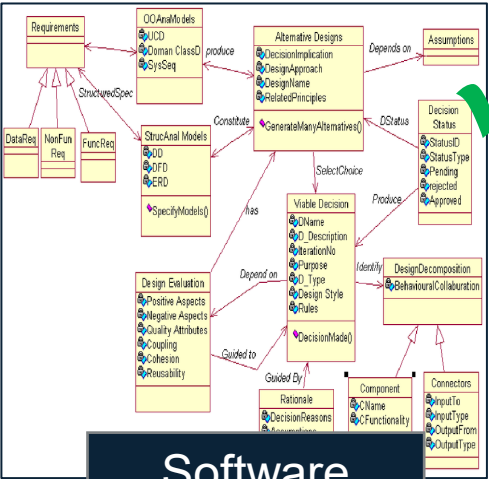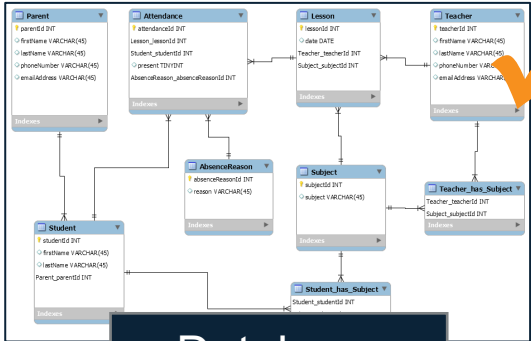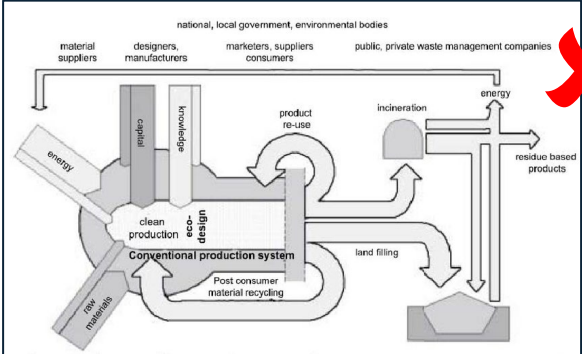

Software
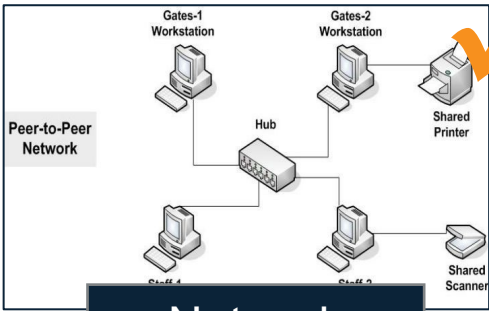
Network
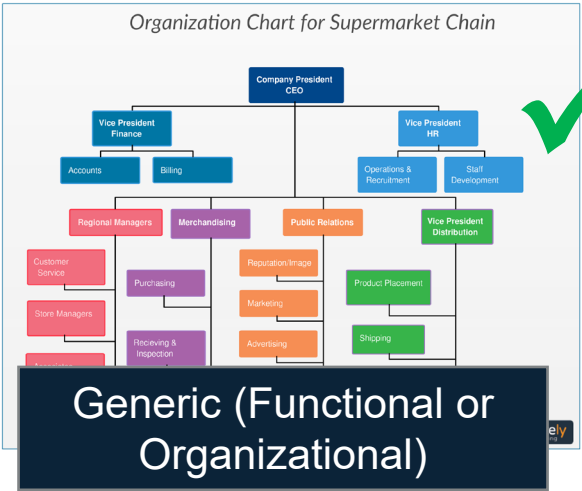
Generic (Functional or Organizational)
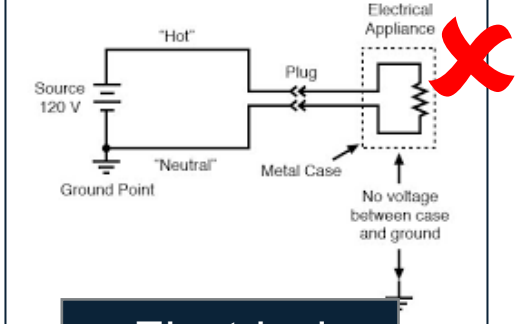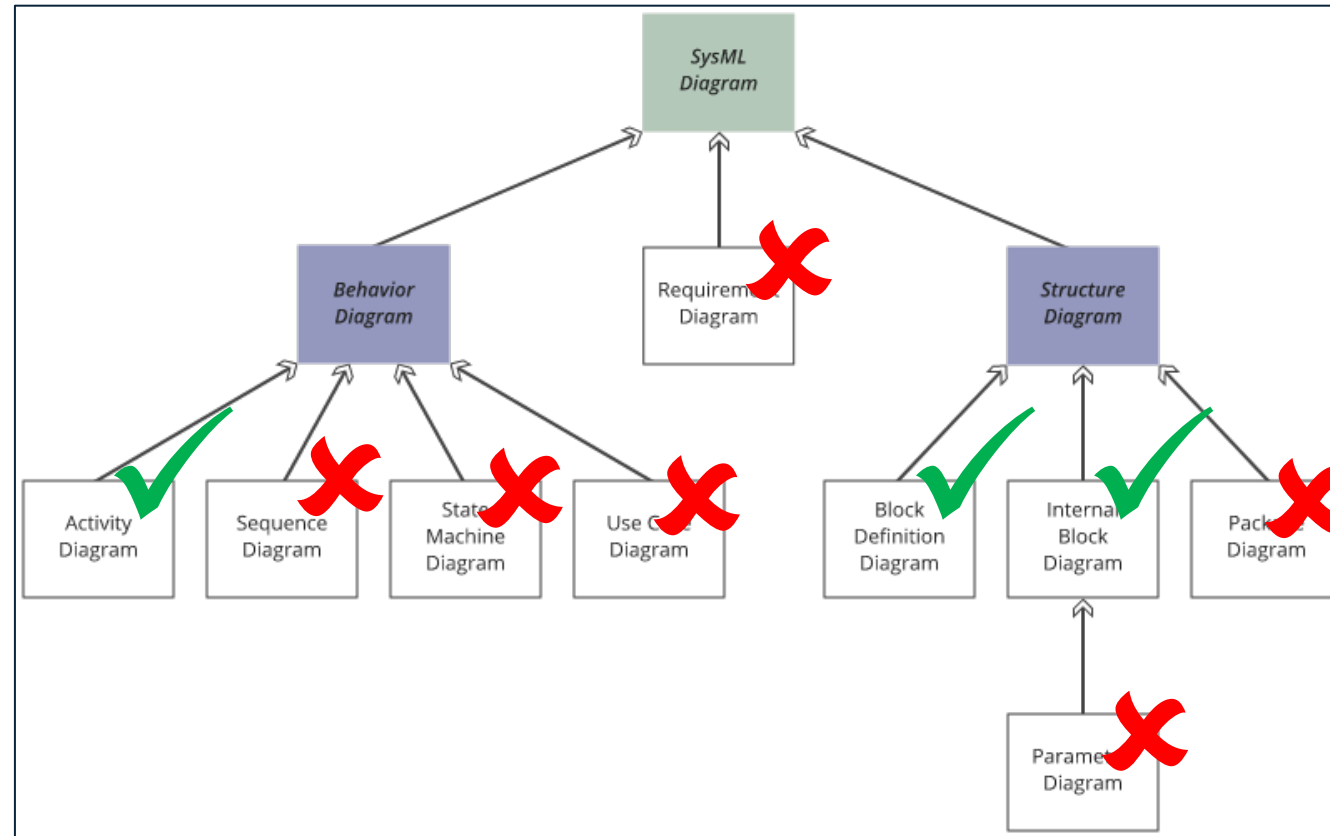
Electrical

Database

Mechanical (Physical)

✓ Able to read

✓ Partially able to read (requires more development)

✗ Unable to read (Potentially out of scope for FY21)

MITRE

Attribute #4: Type

An Image's Type reflects the sort of information it is meant to convey.

For the purposes of this research, type loosely aligns with SysML diagram types

Each Type may come with a unique set of shapes and semantics that would need to be handle, and would need to be sorted into the appropriate SysML diagram type as part of the end product as well

MITRE