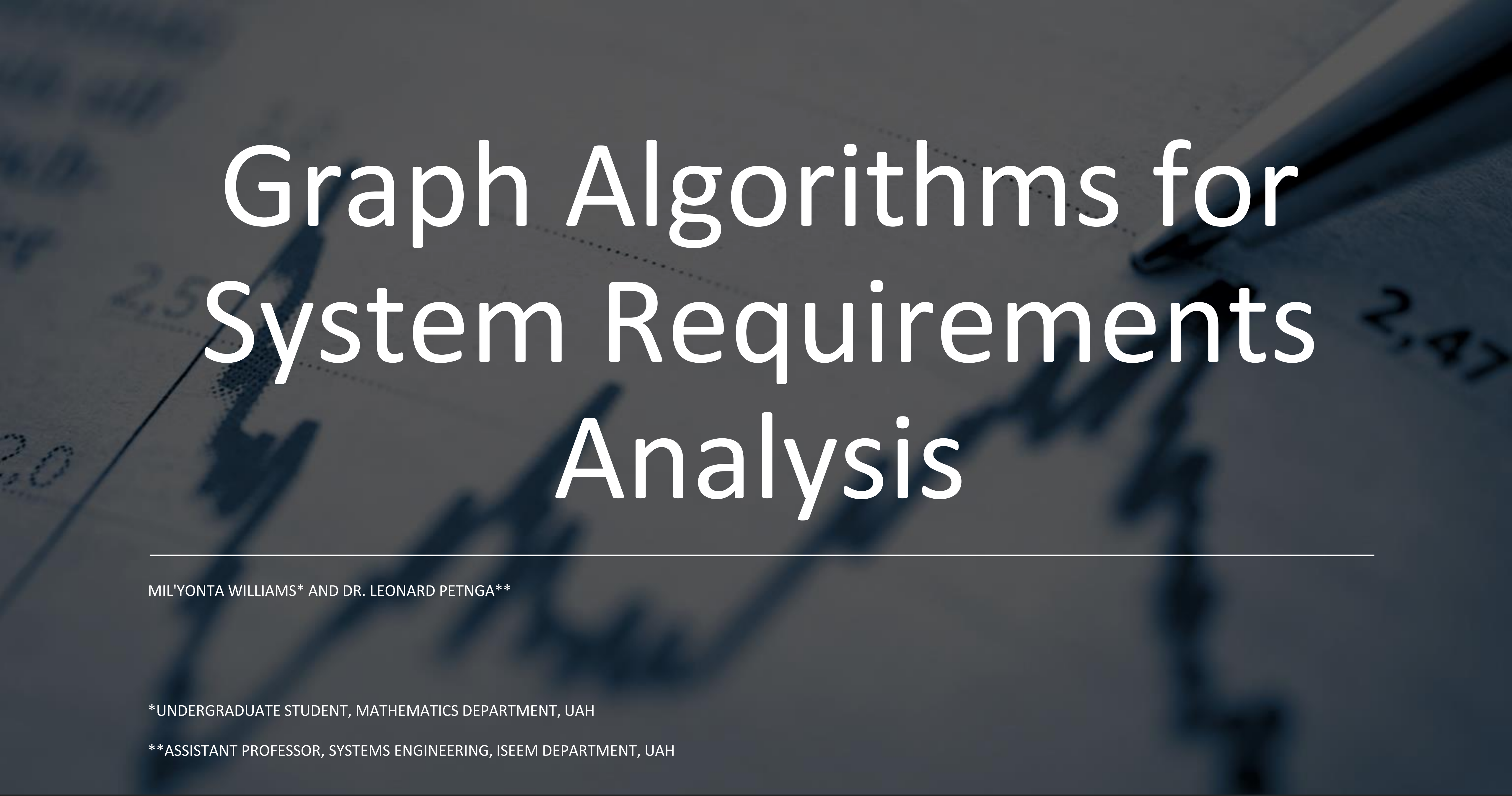


Graph Algorithms for System Requirements Analysis

MIL'YONTA WILLIAMS* AND DR. LEONARD PETNGA**

*UNDERGRADUATE STUDENT, MATHEMATICS DEPARTMENT, UAH

**ASSISTANT PROFESSOR, SYSTEMS ENGINEERING, ISEEM DEPARTMENT, UAH

Outline

- Reliability Requirements
  - Complications
  - Quantitative Reliability Requirements
- Problem
- Graph foundations:
  - Definitions & Types
  - Algorithms
- Labeled Property Graph (LPG)
- The System Modeling Language (SysML)
- Graph Algorithms for Analysis:
  - Betweenness and PageRank Centrality
  - Closeness and Harmonic Centrality
- Graph Based requirements analysis (GRA)
  - A Spacecraft requirements graph (LPG) model
  - Betweenness centrality scores for identification of most impactful.
- Future Work.

Reliability Requirements

- Typically part of a technical specifications document.
- Can be requirements that a company sets for its product and it's own engineers or what it reports as it's reliability to its customers
- Can also be set for suppliers or subcontractors.
- May be specified in either two ways:
    - As a nominal or design value with which the customer would be satisfied.
    - As a minimum acceptable value below which the customer would find the system totally unacceptable and could not be tolerated in the operational environment.

Reliability Requirement Complications

- Can be difficult to specify
- Essentials to a (reliability) requirement model should include:
  - Measurable (by test or analysis)
  - Customer usage and operating environment
  - Time
  - Failure definition
  - Confidence

Quantitative Reliability Requirement

To be meaningful, a **reliability requirement** must be specified quantitatively. There are four basic ways in which a **reliability requirement** may be defined:

- As a "mean life" or mean time between failure (MTBF):
  - Useful for long life systems in which the form of the reliability distribution is not too critical.
- As a probability of survival for a specified period of time, t:
  - Useful for defining reliability when a high reliability is required during the mission but mean time to failure beyond the mission period is of little tactical consequence except as it influences availability.

Quantitative Reliability Requirement (cont.)

- As a probability of success, independent of time:
  - Useful for specifying the reliability of one-shot devices such as the flight reliability of missiles, the detonation reliability of warheads, etc.
- As a "failure rate" over a specified period of time.
  - Useful for specifying the reliability of parts, units, and assemblies whose mean lives are too long to be meaningful or whose reliability for the time period of interest approaches unity.

Problem statement

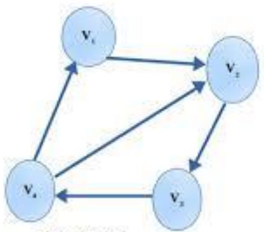
- The issues that exist within requirements is, as the system goes through its lifecycle and more diverse viewpoints and concerns are voiced and frequent changes are made across board, the system model grows extremely large. Requirements formulation, tracking and management becomes extremely difficult. Consequences can be dire:
  - At best: inefficient system design
  - At worst: project or system failure.

- The limitations of the current approach are that current models offer limited capabilities to being defect and ambiguity- free while also lacking mechanisms for probing the requirement model for deep insights that can inform the design at all phases of the system development lifecycle .

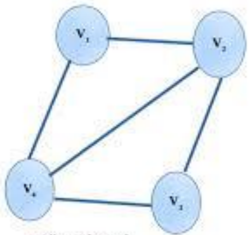
- Goal: Develop and implement prototypes of requirement domain compliant directed multigraph algorithms to support the investigation and analysis of SysML requirement models.

Graph Foundations: Definitions & Types

- Graphs are modeling tools used for representing, finding, analyzing, and optimizing relationships between elements.
- Different types of graphs:
  - Directed / Undirected
  - Weighted/ Unweighted
  - Sparse/ Dense

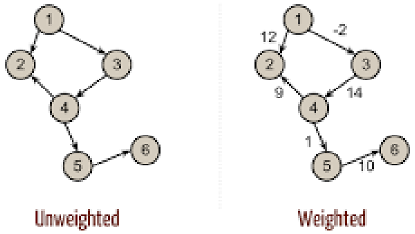


Directed Graph

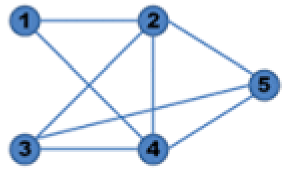


Undirected Graph

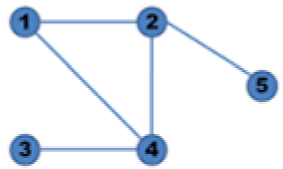
Weighted Graph



Dense Graph



Sparse Graph



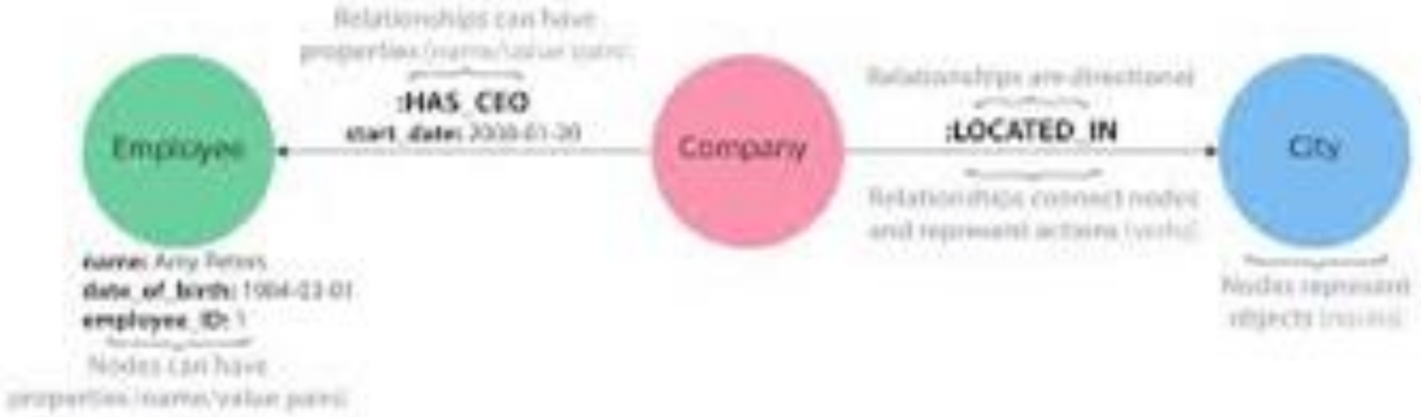
- Graphs consist of vertices (or nodes) being the objects in the system and edges being the relationships and how the objects are connected.

Graph Foundations: Algorithms

- The different types of operations that can be performed on graphs range from testing for adjacency, adding/ removing vertices or edges, as well as obtaining values for specific edges or vertices.
- Different kinds of algorithms that can be developed and applied on graph data are:
  - Connected Components
  - Shortest Path
  - Minimum Spanning Tree
  - Centrality (Betweenness, PageRank, Closeness, and Harmonic)

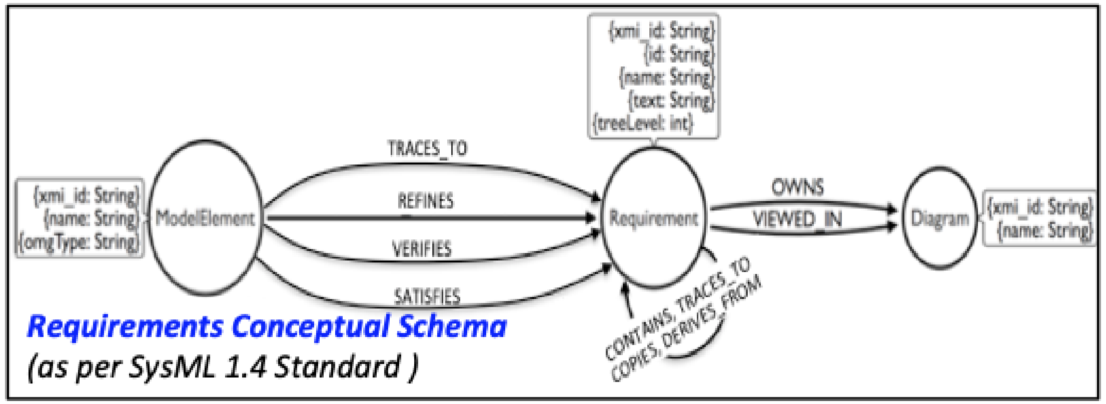
Label Property Graphs (LPG)

- A **Labeled Property Graph** is the property graph model, where data is organized as nodes, relationships, and properties (data stored on the nodes or relationships).
- LPG differ from traditional graph infrastructures because it gives you the ability to clarify the distinction from systems and subsystems.
- They are in almost all industries, including financial services, government, energy, technology, retail, and manufacturing.


Relationships can have properties (name/value pairs)
:HAS_CEO
start_date: 2008-01-20
Employee
Company
Relationships are directional
:LOCATED_IN
Relationships connect nodes and represent actions (verbs)
City
name: Amy Peters
date_of_birth: 1984-03-01
employee_ID: 1
Nodes can have properties (name/value pairs)
Nodes represent objects (nouns)

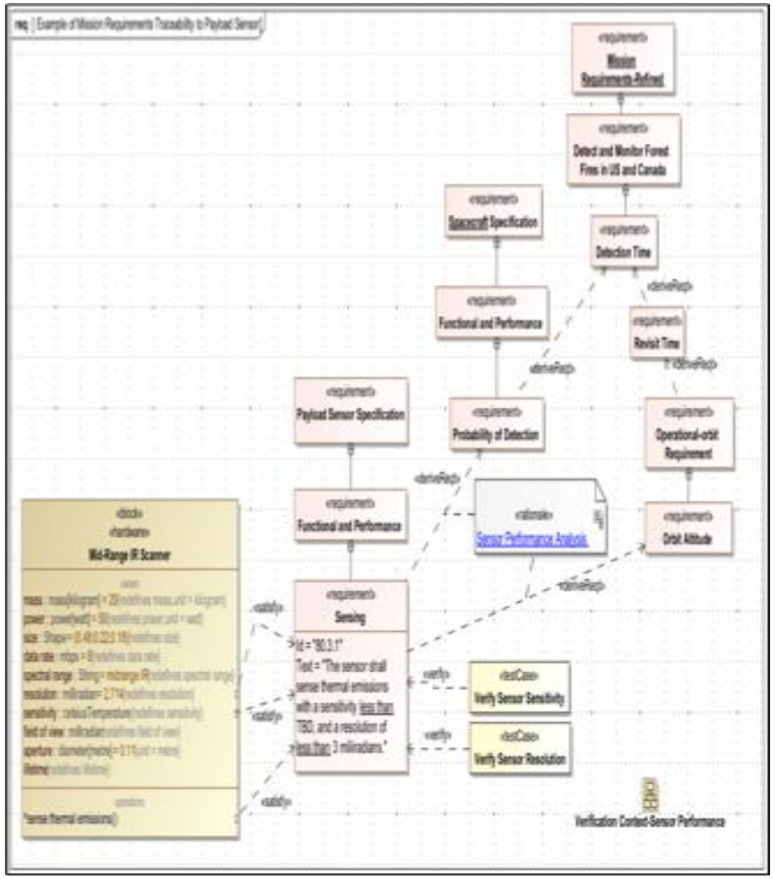
The System Modeling Language (SysML)

- The **System Modeling Language (SysML)** is a general-purpose modeling language for systems engineering applications.
- SysML incorporates the requirement diagram



LPG graph schema

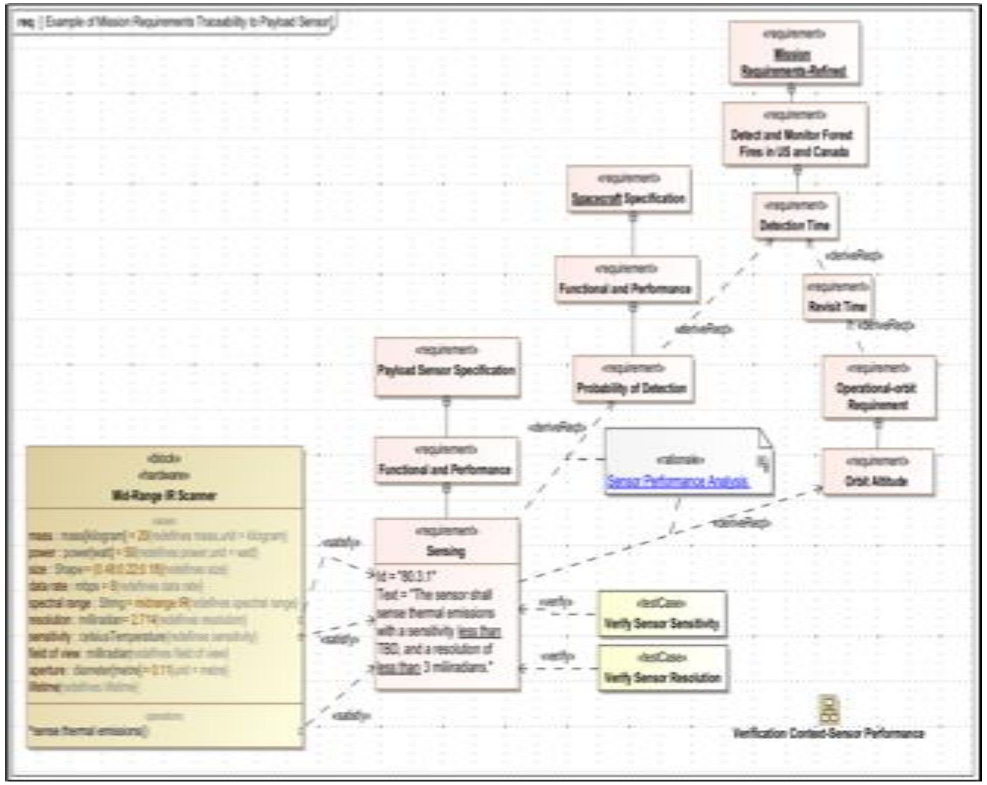
Encoded as LPG using


req [Example of Mission Requirements Traceability to Payload Sensor]
«requirement»
Mission Requirements-Refined
«requirement»
Detect and Monitor Forest Fires in US and Canada
«requirement»
Spacecraft Specification
«requirement»
Detection Time
«deriveReqt»
«requirement»
Functional and Performance
«requirement»
Revisit Time
«deriveReqt»
«deriveReqt»
«requirement»
Payload Sensor Specification
«requirement»
Probability of Detection
«requirement»
Operational-orbit Requirement
«deriveReqt»
«rationale»
Sensor Performance Analysis
«requirement»
Orbit Altitude
«block»
«hardware»
Mid-Range IR Scanner
«requirement»
Functional and Performance
«deriveReqt»
«satisfy»
«requirement»
Sensing
Id = "80.3.1"
Text = "The sensor shall sense thermal emissions with a sensitivity less than TBD, and a resolution of less than 3 milliradians."
«verify»
«testCase»
Verify Sensor Sensitivity
«satisfy»
«verify»
«testCase»
Verify Sensor Resolution
«satisfy»
sense thermal emissions()
Verification Context-Sensor Performance

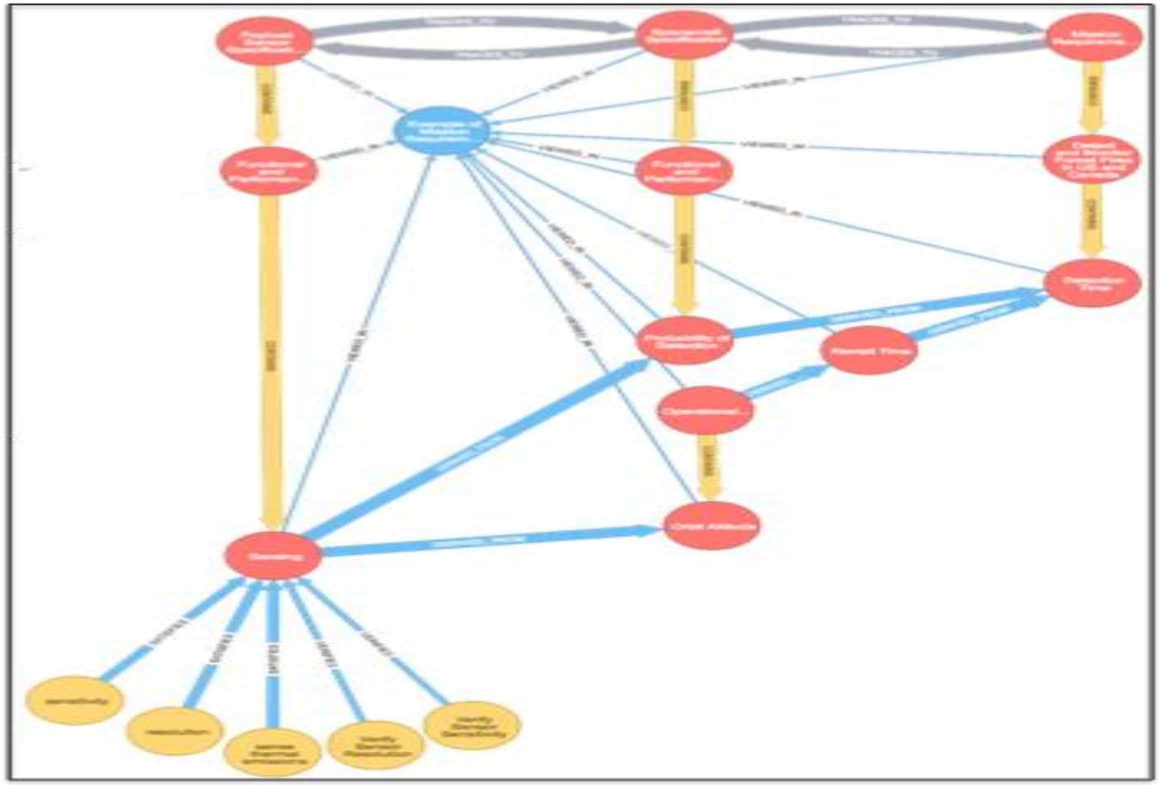
SysMLRequirement Diagram

Equivalent requirement models: SysML vs. LPG

SYSTEM MODELING LANGUAGE


req [Example of Mission Requirements Traceability to Payload Sensor]
«requirement»
Mission Requirements-Refined
«requirement»
Detect and Monitor Forest Fires in US and Canada
«requirement»
Spacecraft Specification
«requirement»
Detection Time
«deriveReqt»
«requirement»
Functional and Performance
«requirement»
Revisit Time
«deriveReqt»
«deriveReqt»
«requirement»
Payload Sensor Specification
«requirement»
Probability of Detection
«requirement»
Operational-orbit Requirement
«deriveReqt»
«rationale»
Sensor Performance Analysis
«requirement»
Functional and Performance
«requirement»
Orbit Altitude
«block»
«hardware»
Mid-Range IR Scanner
mass : mass[kilogram] = 20
power : power[watt] = 50
size : Shape
data rate : mbps = 8
spectral range : String = midrange IR
resolution : milliradian = 2.714
sensitivity : celsiusTemperature
field of view : milliradian
aperture : diameter[metre] = 0.11
lifetime
"sense thermal emissions()
«satisfy»
«requirement»
Sensing
Id = "80.3.1"
Text = "The sensor shall sense thermal emissions with a sensitivity less than TBD, and a resolution of less than 3 milliradians."
«deriveReqt»
«satisfy»
«satisfy»
«verify»
«testCase»
Verify Sensor Sensitivity
«verify»
«testCase»
Verify Sensor Resolution
Verification Context-Sensor Performance

LABEL PROPERTY GRAPH



Graph Algorithms for Requirements Analysis: Betweenness and PageRank Centrality

BETWEENNESS CENTRALITY

Betweenness centrality: measure of the amount of influence a node has over the flow of information in a graph.

$$BC(A) \sum_{I \neq J \neq A} \frac{\sigma_{IJ}(A)}{\sigma_{IJ}}$$

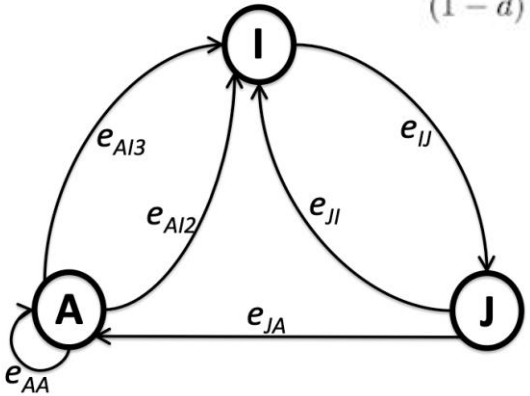
$\sigma_{IJ} =$ Total number of the shortest paths from node I to node J
$\sigma_{IJ}(A) =$ Number of those paths that pass through A

PAGERANK CENTRALITY

PageRank centrality: A “vote” by all the other pages on the web about how important a page is.

$$PR(A) = (1-d) + d(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

$PR(Tn) =$ Self importance for page n
$C(Tn) =$ The count of outgoing links for page n
$d =$ Dampener which is usually 0.85
$(1-d) =$ To ensure the sum of all the pages will be one.



Graph Algorithms for Requirements Analysis: Closeness and Harmonic Centrality

CLOSENESS CENTRALITY

Closeness centrality: measures how short the shortest paths are from node *i* to all nodes

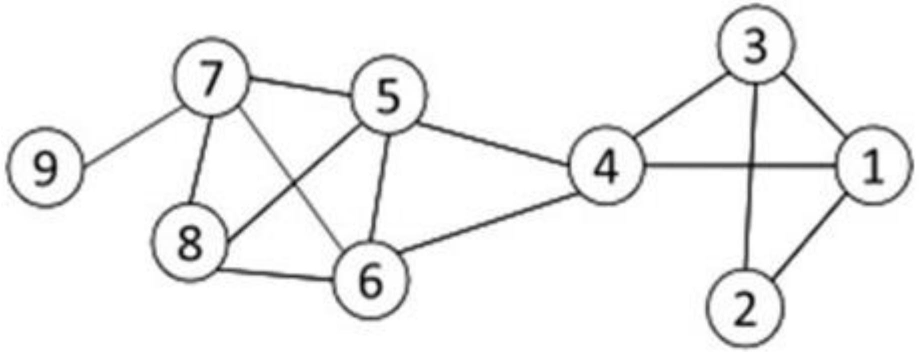
$$CC(i) = \frac{N-1}{\sum_{j} d(i,j)}$$

$i \neq j$
$d =$ length of the shortest path between nodes i and j in the network
$N =$number of nodes

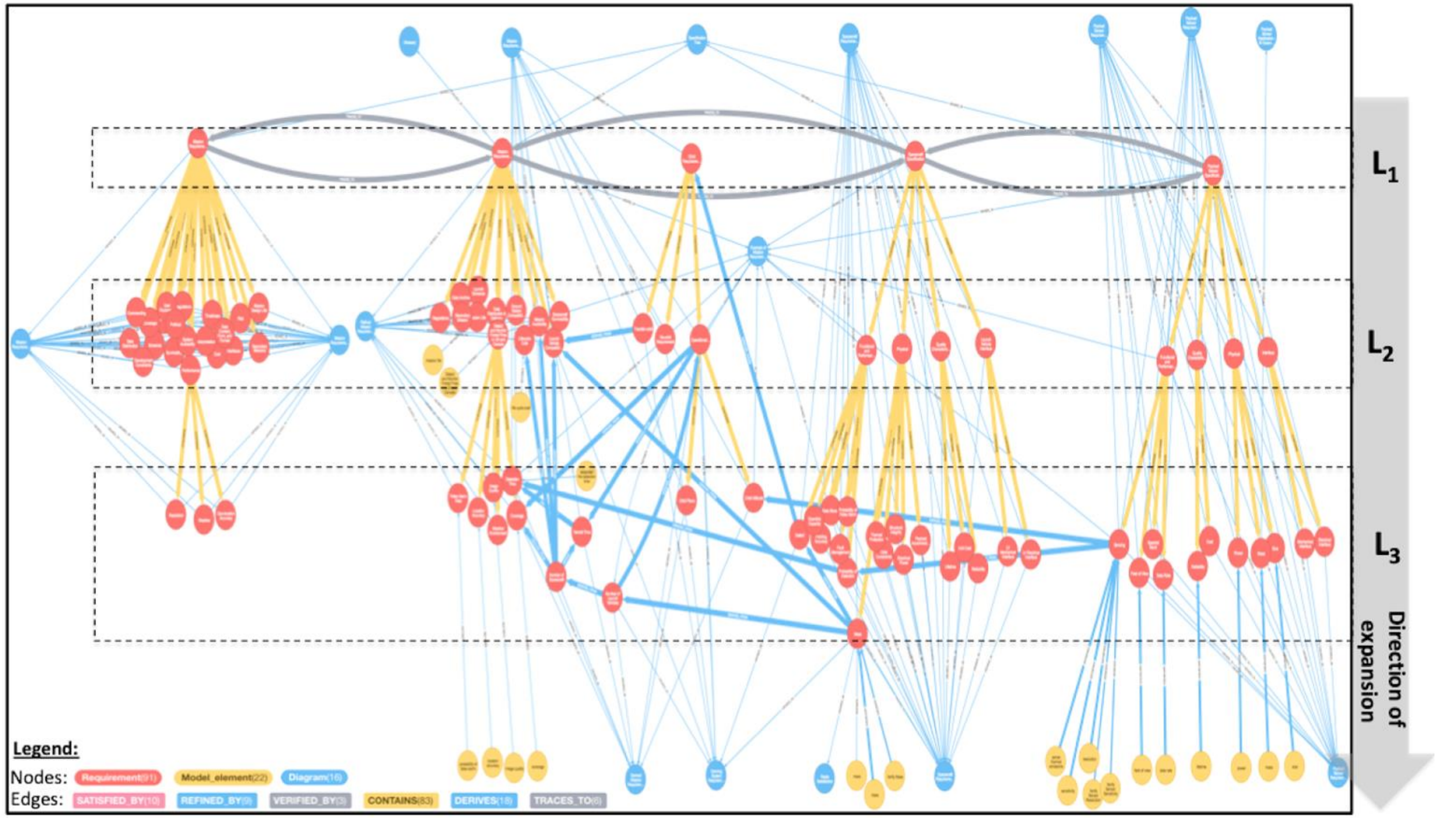
HARMONIC CENTRALITY

Harmonic centrality: reverses the the sum and reciprocal operations in the definition of closeness centrality.

$$HC(i) = \frac{N-1}{\sum_{i \neq j} d(i,j)}$$

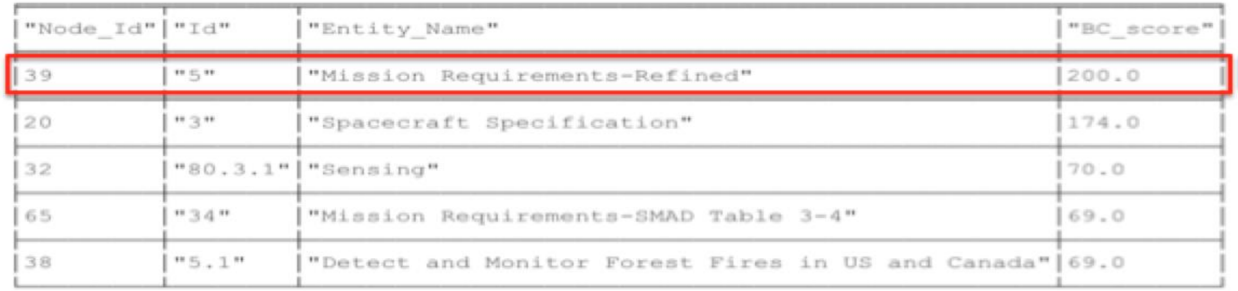


Graph-Based Requirements Analysis (GRA): A Spacecraft requirements graph (a LPG) model

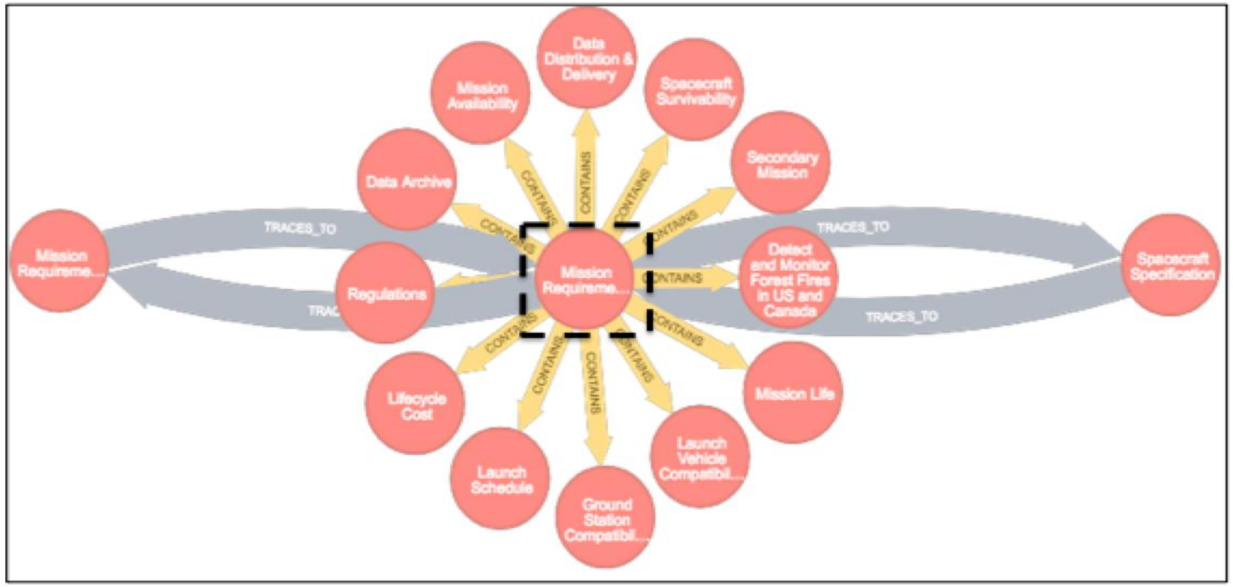

L_1
L_2
L_3
Direction of expansion
Legend:
Nodes: Requirement(91) Model_element(22) Diagram(16)
Edges: SATISFIED_BY(10) REFINED_BY(9) VERIFIED_BY(3) CONTAINS(83) DERIVES(18) TRACES_TO(6)

Spacecraft requirements graph (a LPG) model

GRA: Betweenness centrality scores for identification of most impactful requirements



"Node_Id"	"Id"	"Entity_Name"	"BC_score"
39	"5"	"Mission Requirements-Refined"	200.0
20	"3"	"Spacecraft Specification"	174.0
32	"80.3.1"	"Sensing"	70.0
65	"34"	"Mission Requirements-SMAD Table 3-4"	69.0
38	"5.1"	"Detect and Monitor Forest Fires in US and Canada"	69.0



Betweenness centrality scores for identification of most impactful requirements

Future Work

- A semi-automated procedure to analyze architecture models represented as labeled property graphs (LPG) subsequently transformed and queried for defects and insights in hidden complex patterns
- Need to scale up framework to multiple, interrelated and complex pillars of the architecture languages (such as the SysML), full automation of the procedure as well as architecture-specific graph algorithms

THANK YOU !

Mil’Yonta Williams
mtw0017@uah.edu