Mission Dependency Modeling for Cyber Situational Awareness

William Heinbockel, Steven Noel, and James Curbo
The MITRE Corporation
McLean, Virginia
USA

[heinbockel,snoel,jcurbo]@mitre.org

ABSTRACT

This paper describes a hierarchical graph-based model that captures mission dependencies at various levels of abstraction, showing interdependencies among mission objectives, tasks, information, and cyber assets. For this work, we employ established tools within a structured methodology for cyber resiliency analysis. Our model is focused on a strategic-level military scenario defined in a formal Request for Information (RFI) to industry and research partners by the NATO Multinational Cyber Defense Capability Development (MN CD2) Work Package 2 (WP2). We enhance this scenario with additional mission and operational context, and then build a mission dependency model for the enhanced scenario. It is anticipated that our mission dependency model will be part of an upcoming demonstration of cyber defense situational awareness capabilities in a NATO Communications and Information (NCI) Agency test environment, integrated with data sources that represent the operational military environment.

1.0 INTRODUCTION

A key aspect of maintaining situational awareness in cyberspace is understanding the interdependencies among mission elements, how mission elements depend on cyber assets, and how cyberattacks can potentially impact missions. Capturing the dependencies for realistic missions requires a structured methodology and automated tool support for dealing with complex interrelationships. Such a hierarchical mission dependency model should include high-level mission objectives, tasks that support those objectives, the information required for each mission task, down to the cyber assets that contain and process the information.

Employing a graph-based mission dependency model can help show the transitive (nth order) mission impacts of cyberattacks. For example, a graph traversal query can begin at the victim host of an attack, and traverse the graph to enumerate the mission components that depend on it, showing impact on all effected levels of the mission dependency hierarchy. A query could also traverse in the opposite direction, e.g., to show the "cyber key terrain" supported by a given mission component. Moreover, a mission dependency model must go beyond a pure mathematical graph, to include important semantics such as the underlying logical nature of dependencies (conjunctive or disjunctive), relative criticality, ownership, geographic location, etc.

A Request for Information (RFI) published by NATO in May 2015 [1] articulates requirements for cyber defense situational awareness and decision support. The RFI describes in detail the required cyber defense capabilities. However, its scenarios lack a depth of mission operational dependency tracking and potential courses of action desired for demonstrating advanced tool capabilities, especially for RFI use cases involving asset and mission dependencies. We therefore expand the scenarios with additional operational context, which forms a basis for our dependency modelling.

2.0 MODEL BUILDING PROCESS

To build our mission dependency model, we employ MITRE's Structured Cyber Resiliency Analysis Methodology (SCRAM) [2]. Illustrated in Figure 1, SCRAM defines the processes for performing varying levels of cyber resiliency analyses at different points in the lifecycle of a system, system-of-systems, or mission. SCRAM also details resources such as frameworks and models, value scales, and datasets to support these analyses.

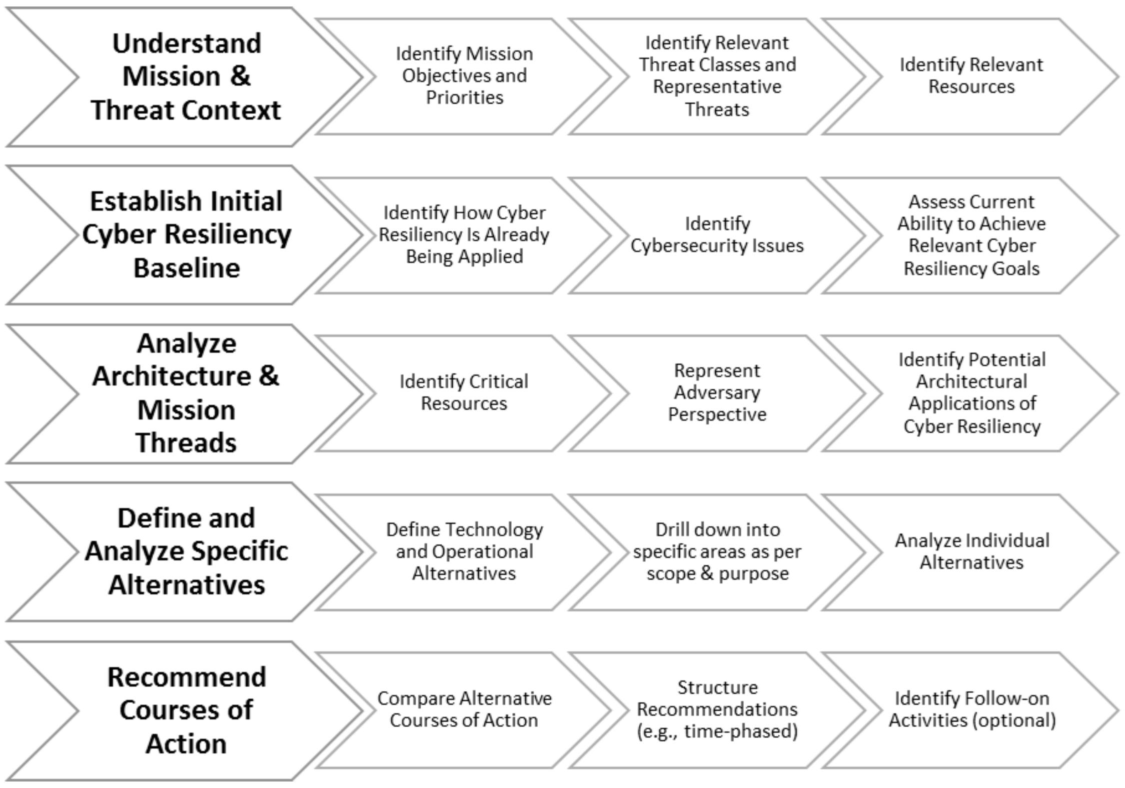
Understand Mission & Threat Context
Identify Mission Objectives and Priorities
Identify Relevant Threat Classes and Representative Threats
Identify Relevant Resources
Establish Initial Cyber Resiliency Baseline
Identify How Cyber Resiliency Is Already Being Applied
Identify Cybersecurity Issues
Assess Current Ability to Achieve Relevant Cyber Resiliency Goals
Analyze Architecture & Mission Threads
Identify Critical Resources
Represent Adversary Perspective
Identify Potential Architectural Applications of Cyber Resiliency
Define and Analyze Specific Alternatives
Define Technology and Operational Alternatives
Drill down into specific areas as per scope & purpose
Analyze Individual Alternatives
Recommend Courses of Action
Compare Alternative Courses of Action
Structure Recommendations (e.g., time-phased)
Identify Follow-on Activities (optional)

Figure 1: Overview of the SCRAM Process.

In applying SCRAM to develop the model, we leverage a number of MITRE dependency analysis tools:

- Crown Jewels Analysis (CJA) [3] is a process and corresponding toolset for "identifying those cyber assets that are most critical to the accomplishment of an organization's mission."
- CyCS (Cyber Command System) [4] is MITRE's proof-of-concept cyber situational awareness tool for addressing "mission-assurance challenges for highly distributed enterprise systems of systems through vulnerability, threat, and consequence management."
- CyGraph [5][6] is a tool for real-time cyber situational awareness that combines isolated data and events into an ongoing overall picture for decision support and situational awareness.

SCRAM typically relies on dependency maps to help understand what is most critical, beginning in system development and continuing through system deployment. The dependency map starts by identifying missions and assigning relative prioritization. From there, dependencies flow down through operational tasks and system function to cyber assets. These dependencies are expressed qualitatively in terms of impact on a parent node resulting from a failed or degraded child node, with provisions to minimize subjectivity. With a complete model, SCRAM tools can predict the impact of a cyber asset failure or degradation as the realization of each parent/child logical statement, tracing the potential impact upward to high-level mission tasks and objectives.

Analysis within SCRAM provides a dependency map to associate missions, data flows, and cyber assets, along with the methodology to "roll up" cyber asset criticality based on higher-order associations, such as mission or operational priorities. The dependency model can also be inverted (Figure 2), to identify potential mission impacts of an incident.

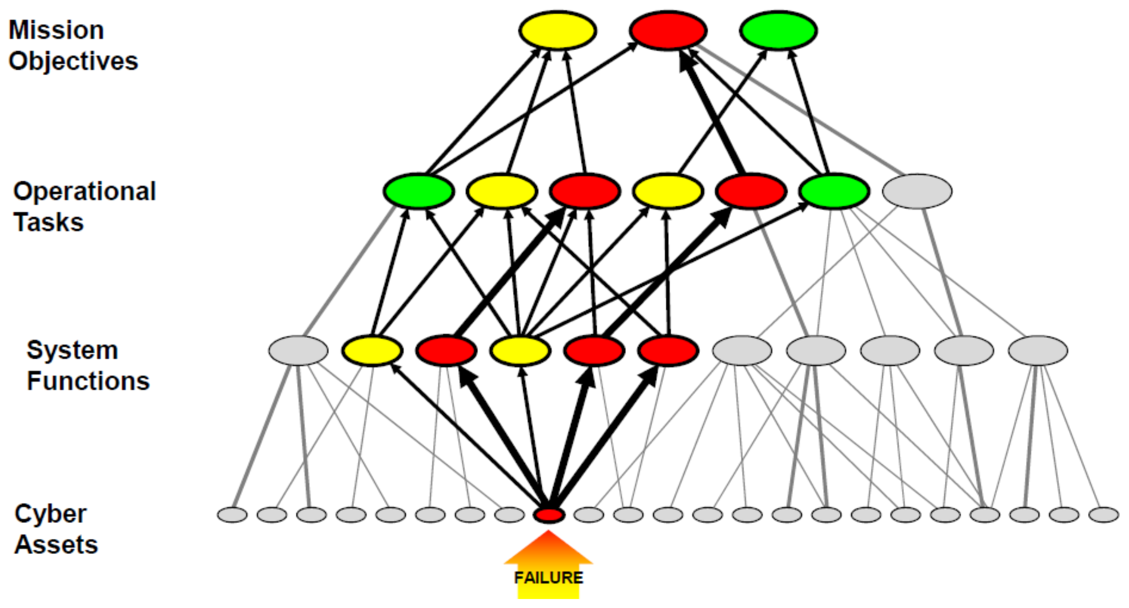

Figure 2: Assessing Failure Impact.

3.0 SCENARIO ENHANCEMENTS AND MISSION DEPENDENCY MODEL

Our effort focuses on enhancing the strategic-level "Oranjeland APT" scenario of the RFI. As appropriate for the early design and development stages, we concentrate on the first and third SCRAM steps of Figure 1: Understanding Mission & Threat Context and Analyze Architecture & Mission Threads. For the mission and threat context, the RFI documents the APT threat in the scenario, but gives limited insights into the supported operational mission of medical evacuations (Medevac).

Here is an excerpt of the RFI scenario and operational context [1]:

> "A network administrator in the RATM Network Operations Centre (NOC) notices unusual network activity, which has not been detected by the antivirus, on a server at Regional Command North (RC-N), and suspects an Advanced Persistent Threat (APT).
>
> He contacts the NATO Cyber Security Operations (CSOps), who create an incident ticket. CSOps does a series of initial investigations using the CDSAS [cyber defense situational awareness

system], and identifies this to be an Integrated Command and Control system (ICC) server. They collate all relevant information and options into a report and then they contact the Comprehensive Crisis and Operations Management Centre (CCOMC) Cybercell (CCC), recommending the course of action to disconnect the ICC server to disrupt the APT. CCC is aware of a planned mission in the area affected. Planned downtime is an important factor along with the negative impacts of the APT (e.g. exfiltration of data)…

The system can also then show the risk that if the server is turned off, CHAT would become unavailable as it is hosted on the same server. Loss of CHAT would appear with an impact of Medical Evacuations (Medevac) being hindered significantly as they are mainly done over CHAT. With a mission about to commence in the area, Medevac is likely to be an essential service. The Commander would need to judge whether the risk should be taken of running the mission without a CHAT capability, or whether it is essential to run the mission, and essential to have CHAT available, in which case the server will need to remain on. The system should allow him to compare the risks of keeping the server on, and whether that in its self will pose a threat to his mission."

The next section (Section 3.1) describes scenario enhancements through expanded courses of action for maintaining mission readiness, applying the SCRAM methodology. In Section 3.2, we describe the mission model for the RFI scenario. Section 3.3 then describes the data requirements for this model.

3.1 Expanded Courses of Action

It is important that solutions for cyber defense situational awareness incorporate the full scope of the available courses of action. Sometimes the best course of action is obvious (e.g., use a redundant server). Other times the best course of action lies within a different command. A key capability of a situational awareness solution is to understand the mission dependencies and common operating picture (COP), and to help a commander make the best decisions, leveraging the maximum amount of information available.

We recommend approaching the courses of action in order of resource intensity. Priority should be given to those courses of action that are fast, efficient, and require the least amount of coordination. In general, a solution should be able to provide courses of action within these three domains:

- Technical – redundant or spare cyber assets
- Service – redirect from other area or fall back on alternative functionality
- Operational – leverage concept of operations (CONOPS), call alternative commands for support

Within each of these domains, there are at least two categories of alternatives for courses of action:

- Technical
  - **Replace**: Can the cyber asset (e.g., system, network) be replaced with redundant components (e.g., spare servers, redundant network paths)?
  - **Reconstitute**: Can the cyber asset be reconstituted? For example, can the system replicate a server instance from a gold master virtual machine image, or dynamically reconfigure the network.

- Service
  - **Reposition**: Are there identical services, potentially in neighbouring geographic regions, that can be repositioned to cover the mission area?
  - **Repurpose**: Can the lost service functionality be (partially) replicated by repurposing other services? For example, email service may be used to provide some data transmission functionality similar to chat. Voice services (radio, VOIP) can be used as an alternative to digital communications (email, chat).
- Operational
  - **Reuse**: Can the missing functionality be fulfilled by reusing a similar service offered by another entity or organization?
  - **Retask**: Can another entity or organization be retasked to complete or support the mission?

To provide the cyber defense situational awareness solutions with more introspective to battlefield decision support, we propose expanding the “Oranjeland APT” operational RFI scenario with more detailed alternatives:

- Introduce three commands (AIR, LAND, SEA) with the Medevac being a LAND-led search and rescue mission.
- The solution should evaluate the current medical evaluation mission dependencies and compare them against the available systems, services, and operational options available in within the COP and NSIS feeds.
  - First, the solution should evaluate technical alternatives to identify any **replacement** (ICC) servers that be quickly brought online to provide the CHAT capability. Another course of action would be to determine whether the CHAT server can be **reconstituted** via backups or virtual machines.
  - Next, the solution might evaluate service oriented alternatives are usually more complex and take more resources to deploy.
    - The easiest service course of action would be to **reposition** an existing chat service. For example, the NATO-led RATM coalition may be able to reposition the nearby Applestan CHAT service to cover the Medevac area.
    - A second option is to evaluate whether the mission tasking can be altered by **repurposing** other, similar services to replicate the missing communications functionality provided by the CHAT service. For example, instead of using ICC CHAT, the Medevac may be able to use EMAIL or voice (radio, VOIP) to establish communication.

The final courses of action require operational-level coordination across commands. If the LAND-led Medevac is unable to perform the rescue mission, maybe the assets from another command (e.g., AIR or SEA) can assist. The situational awareness system should evaluate whether the SEA support forces have a CHAT service for reuse by AIR Medevac. A last option is to retask the mission to another command. For example, maybe the LAND support forces have a nearby medic team that can be caravanned in, or the SEA support forces have a helicopter that can be used for a SEA-led Medevac.

3.2 Mission Model

Our mission model maps dependencies from the high-level Rescue mission led by the NATO RATM, down to the system or asset level. Figure 3 is an overview of the model. This model focuses on expanding the Medevac scenario. It is a partial model, e.g., does not include dependencies for the top-level Medic, Rapid Deploy Medic, and SEA Medevac platforms.

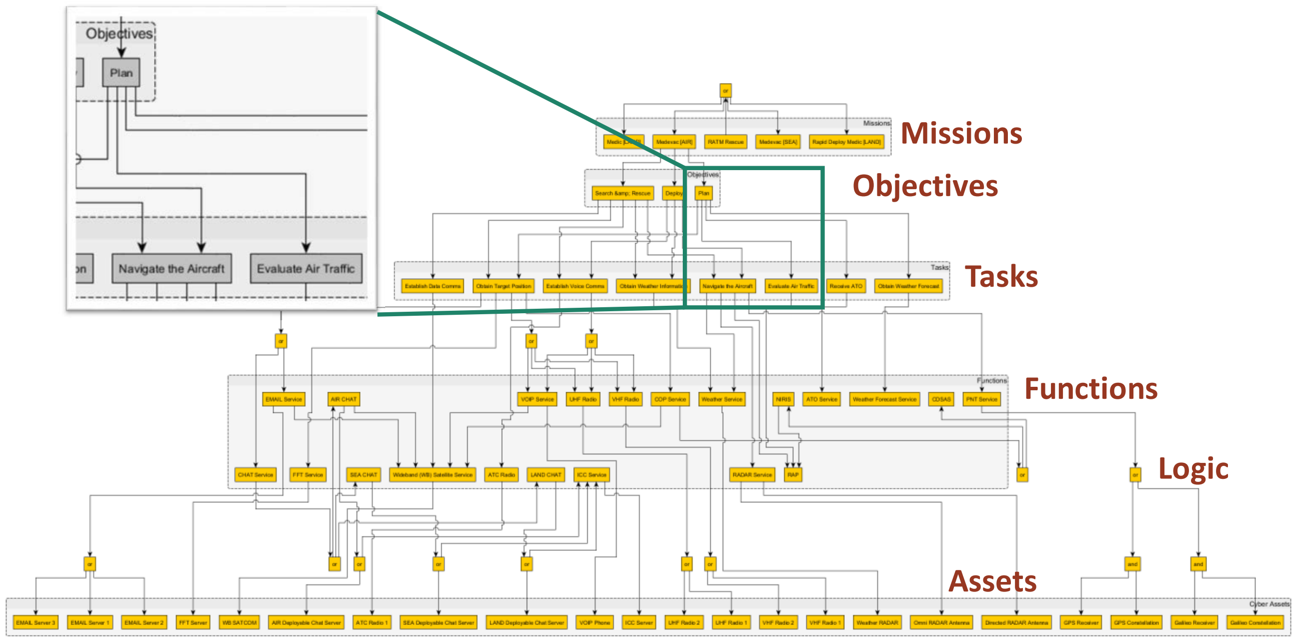

Figure 3: Mission Model Overview.

The overview in the figure omits the dependency criticality weighting and dependency logic included in the model itself. Criticality is one way of denoting the impact on the parent if the dependency is lost or unavailable. To keep discussions focused, SCRAM uses a simple four level criticality model (Table 1), which is also used across the United States Department of Defense [7][8]. Logical relations such as the fact that either the primary or backup ICC Server must be available are included in the model but not portrayed in the figure.

Table 1: SCRAM Criticality Levels.

Criticality	Description
Level I	Total Mission Failure
Level II	Significant Degradation
Level III	Partial Capability Loss
Level IV	Negligible or No Loss

3.3 Model Data Requirements

Cyber defense situational awareness solutions need to minimally track dependencies, dependency strength, geographic location, area of responsibility, area of impact, and owning unit for each mission component. We employ the XML-based GraphML format to express this mission dependency information as a graph model. Each edge is a directed dependency from the source to the destination. Features such as the dependency criticality, descriptions, locations (latitude and longitude), and command are added through node and edge keys. To support dependency logic (Boolean AND/OR) requirements, we use edge properties. Table 2 provides a list of GraphML node and edge keys used in the RFI mission model.

Table 2: Property Keys in RFI Mission Model.

Key Name	Edge or Node Key	Description
Name	Node	The name of the node
description	Node	A description of the node
Type	Node	The type of resource the node represents (Operational Mission, Platform, Objective, Task, System Function, or IT Asset)
Latitude	Node	The node's geolocation latitude
longitude	Node	The node's geolocation longitude
Radius	Node	The radius [in meters] of coverage or area of responsibility centered at [latitude, longitude]
Location	Node	The location name where the node resource resides
command	Node	The owning command: LAND, SEA, or AIR
criticality	Edge	The dependency criticality on an ordinal scale from 0-100, where 100 represents a Level I critical dependency

4.0 MISSION DEPENDENCY ANALYSIS

This section explores patterns of interdependent relationships within our mission dependency model. Given the large scale and high complexity typically expected for real missions, this kind of query-driven exploratory analysis is a crucially important capability for cyber situational awareness.

For this analysis we apply CyGraph [5][6], a MITRE a tool for capturing, analysing, and visualizing knowledge about complex relationships in cyber environments. In CyGraph, a mission model (or any other set of graph relationships) is stored in a Neo4j graph database [9][10], and the analyst formulates queries that match patterns in the stored graph. CyGraph query capabilities allow ad hoc exploration of graph models, for focusing on graph patterns of interest.

Figure 4 shows the root of our mission dependency model and its first two levels of dependency, rendered in CyGraph. This is generated by the following query (expressed in Neo4j Cypher query language [11]):

```
MATCH p = ((root {uid: 'Search & Rescue'})-[*1..2]->()) RETURN p
```

This query matches the model root node *Search & Rescue*, and all nodes that are within two child relationships from it. The default assumption is that all children of a node must be successful for the node itself to be successful, i.e., dependencies are conjunctive (Boolean AND) by default. This is indicated by triangular arrowheads pointing from a node to its children.

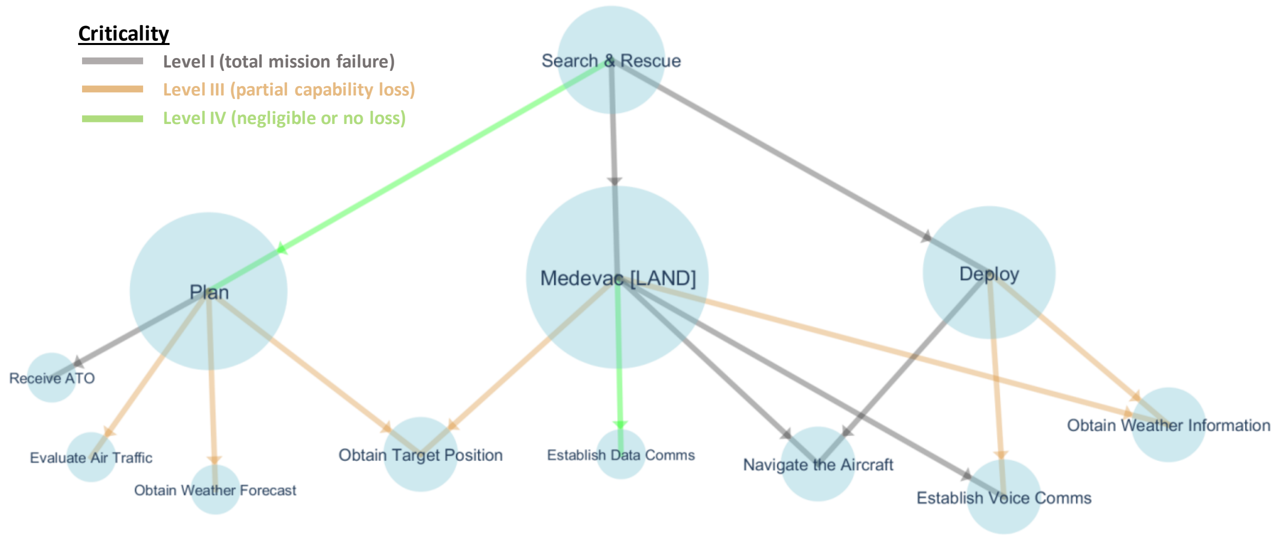

Figure 4: *Search and Rescue* Mission and Two Levels of its Dependencies.

In this high-level overview, we see that there are 4 nodes (*Obtain Target Position*, *Navigate the Aircraft*, *Establish Voice Comms*, and *Obtain Weather Information*) that are each required by two higher-level nodes (parents), thus having potentially greater mission impact if they are compromised.

Figure 4 also includes edge styling (colours) to indicate the level of mission criticality for a mission dependency relationship (graph edge), according the SCRAM Criticality levels defined in Table 1. In the figure, Level I criticality (total mission failure) is black, Level III criticality (partial capability loss) is orange, and Level IV criticality (negligible or no loss) is green. In this model, there are no dependency relationships with Level II criticality (significant degradation).

From Figure 4, there are three children of the *Search & Rescue* node: (1) *Plan*, (2) *Medevac [LAND]*, and (3) *Deploy*. Figure 5 shows the full sub-graph of dependencies for the Plan node. Here is the query:

```
MATCH p = ((root {uid: 'Plan'})-[*]->()) RETURN p
```

This query begins on the *Plan* node and traverses all outgoing (child) edges, i.e., dependencies. The query result (Figure 5) shows that the critical (Level I) dependency of *Plan* on *Receive ATO* (via *ATO Service*) has no redundancy, making it a single point of failure for mission planning of *Medevac [AIR]*.

In Figure 5, some dependencies are disjunctive (Boolean OR) rather than conjunctive, indicated by round (versus triangular) arrowheads. This indicates that the success of any child (among the conjunctive set) is sufficient for the parent to be successful, given that all disjunctive (child) dependencies are met.

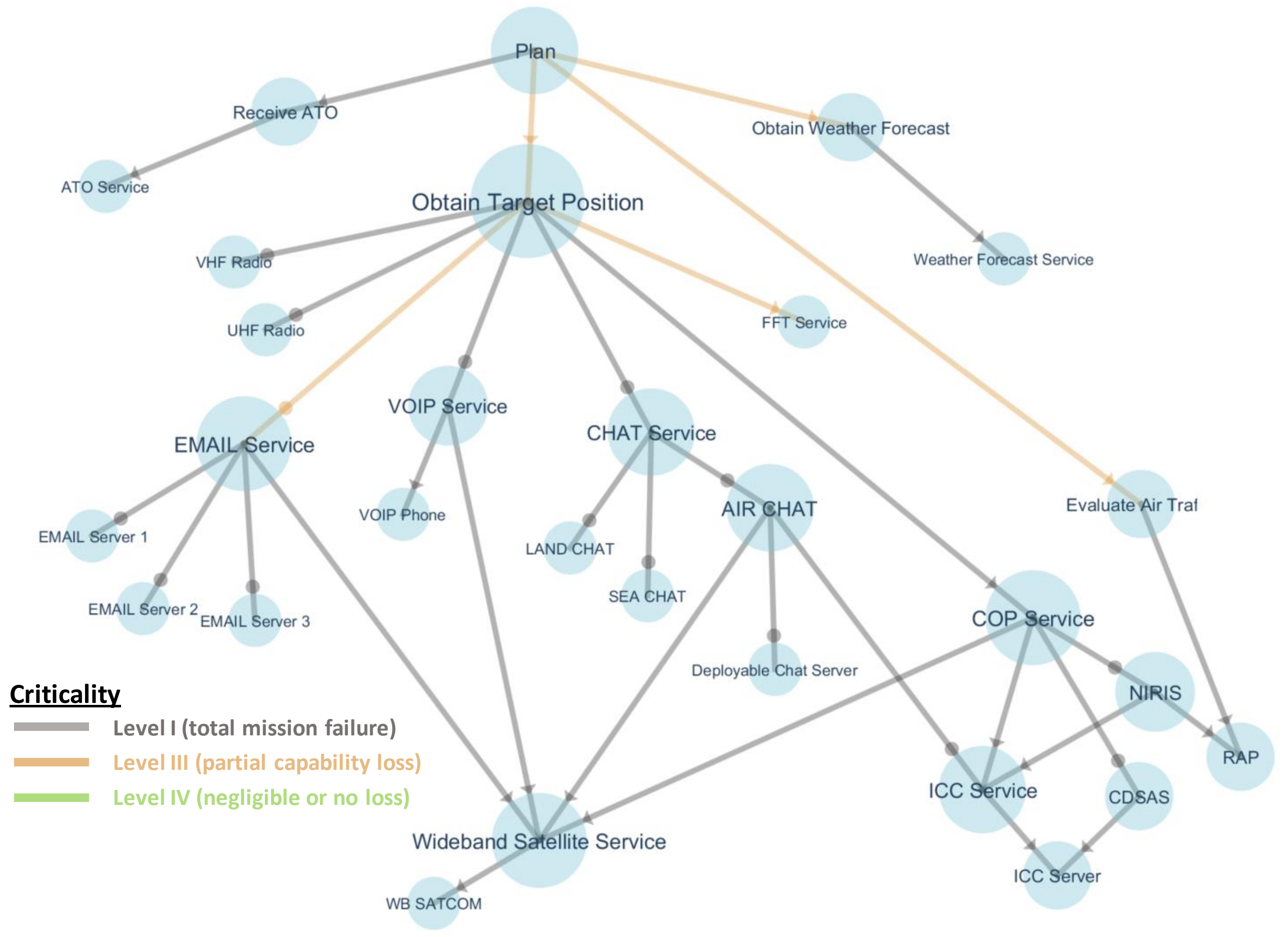

Figure 5: Dependencies for Mission Objective *Plan*.

Figure 6 shows two levels of dependencies for the *Medevac [LAND]* node. Here is the query:

```
MATCH p = ((root {uid: 'Medevac [LAND]'})-[*1..2]->()) RETURN p
```

This query begins at the *Medevac [LAND]* node, and traverses through two levels of children dependencies. The query result in Figure 6 shows that there are two critical (Level I) dependencies for *Medevac [LAND]*, i.e., *Establish Voice Comms* and *Navigate the Aircraft*. Of those, *Establish Voice Comms* depends on redundant nodes and one with only partial loss of capability (Level III). The other (*Navigate the Aircraft*) also has two points of failure of critical (Level I) nodes, i.e., *PNT Service* and *Radar Service*.

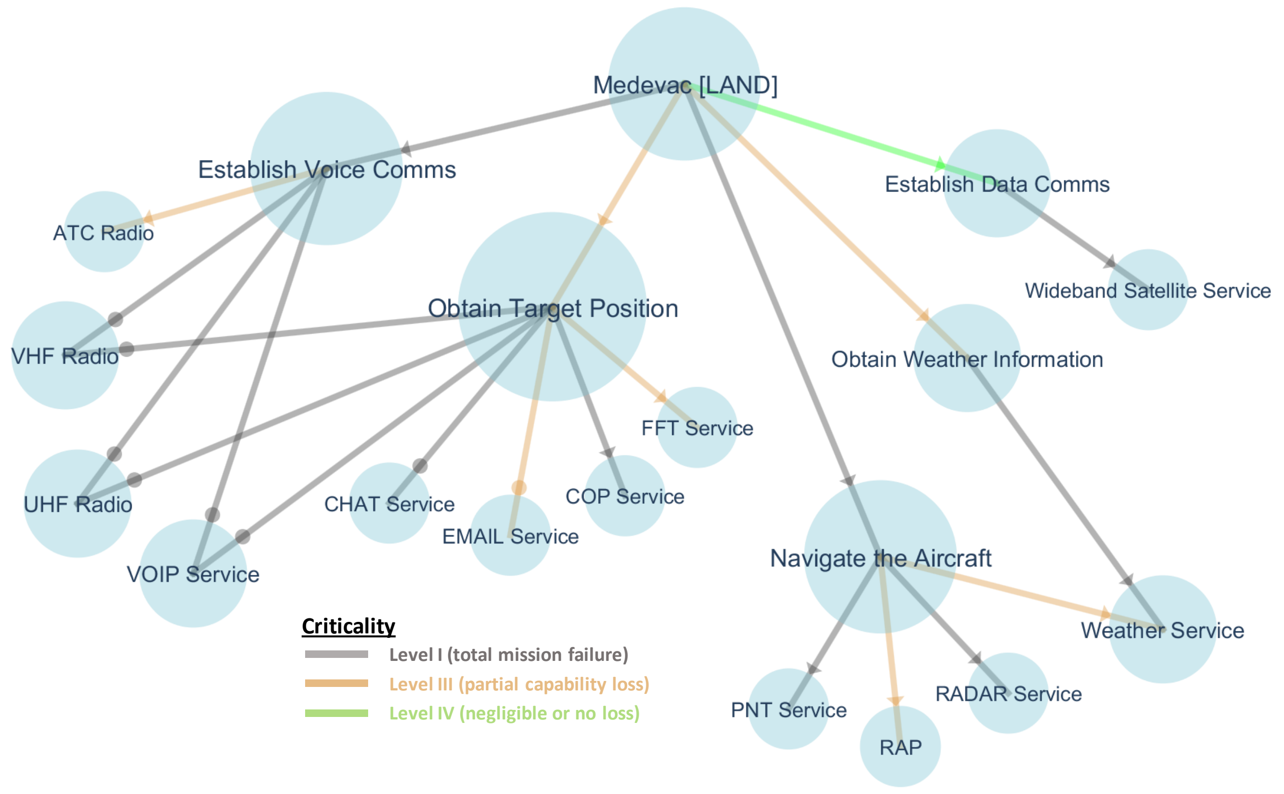

Figure 6: Two Levels of Dependencies for Objective *Medevac [LAND]*.

Figure 7 shows the full sub-graph of dependencies for the Deploy node. Here is the query:

```
MATCH p = ((root {uid: Deploy'})-[*]->()) RETURN p
```

This shows that the single point of failure of a critical node (*Omni Radar Antenna*) leads to failure of the *Deploy* objective, because of the non-redundant critical dependencies on *RADAR Service* and *Navigate the Aircraft*.

As a kind of what-if analysis, we apply a query that analyses mission impact of the loss of the wideband satellite. This matches all dependencies (transitively) on the WB SATCOM node, i.e., everything that depends on it:

```
MATCH p = (()-[*]->(leaf {uid: 'WB SATCOM'})) RETURN p
```

Figure 8 is the query result. This shows that the loss of *WB SATCOM* causes the loss of *Obtain Target Position* (via *COP Service*). This in turn causes partial loss of *Plan*, although that has negligible impact on *Search & Rescue*. It also causes at least partial loss of *Medevac [LAND]*, which also depends critically on *Establish Voice Comms* (which has redundancy protection from the loss of *WB SATCOM*), which in turn causes partial loss of *Deploy*.

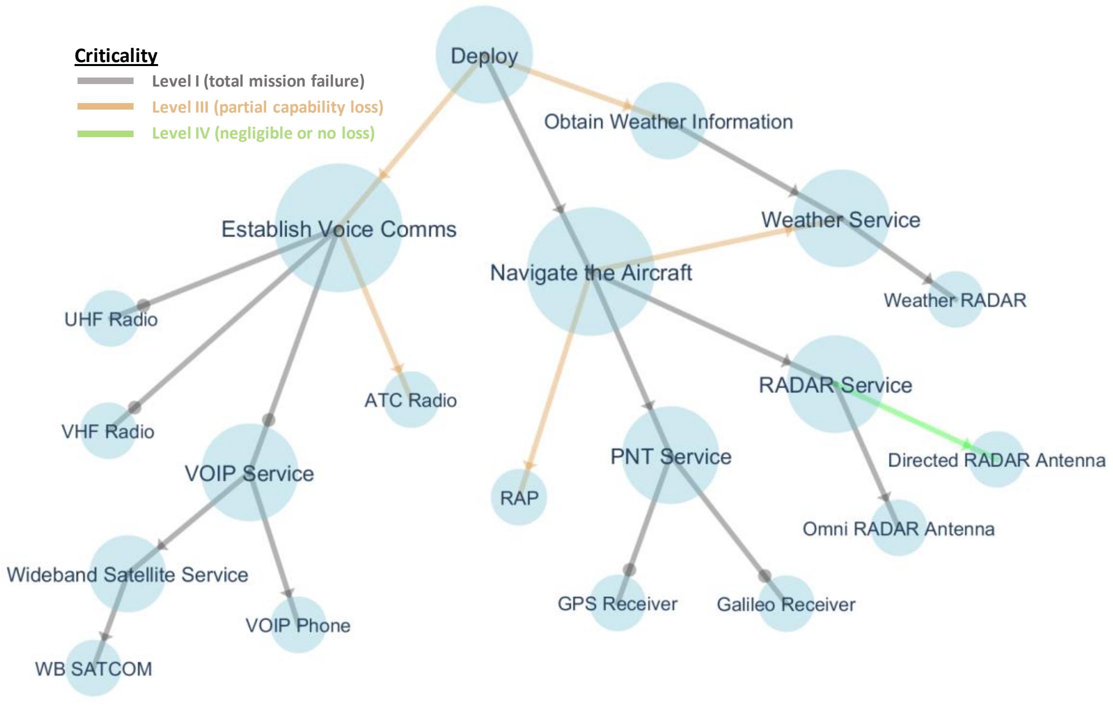

Figure 7: Dependencies for Objective *Deploy*.

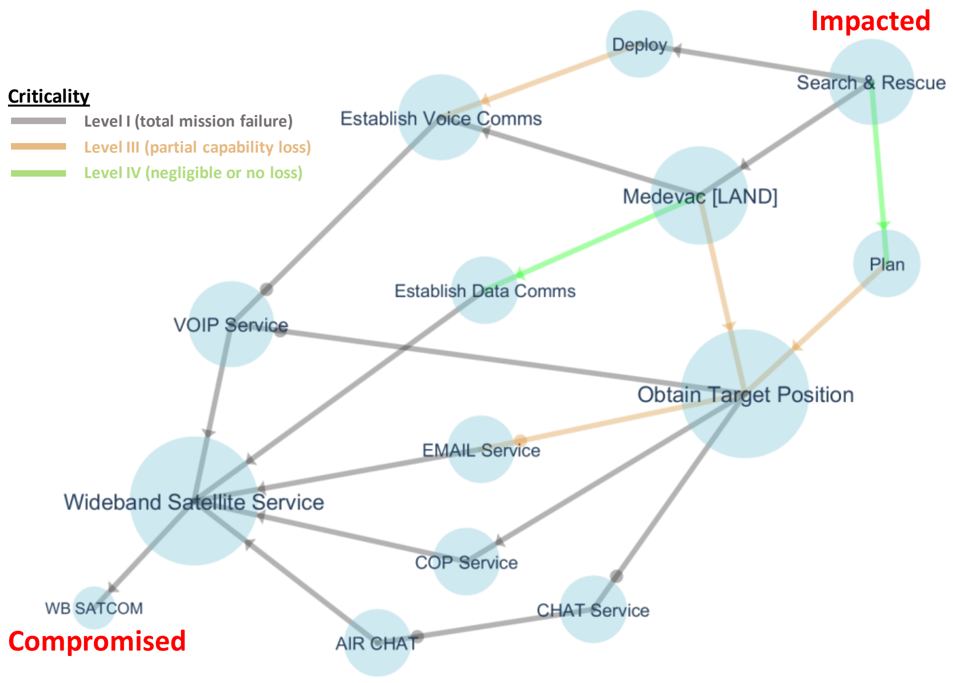

Figure 8: Mission Impact of the Loss of *WB SATCOM*.

A key question in the NATO RFI scenario is to assess the mission impact of the loss *ICC Server*. Here is the query for that:

```
MATCH p = (()-[*]->(leaf {uid: 'ICC Server'})) RETURN p
```

Figure 9 is the query result. There is a non-redundant chain of critical dependencies from *ICC Server* to *ICC Service* to *COP Service* to *Obtain Target Position*, i.e., those node fail. This results in partial loss of objectives *Medevac [LAND]* and *Plan*. This in turn is a critical failure for the *Search & Rescue* mission (via its critical dependency on *Medevac [LAND]*).

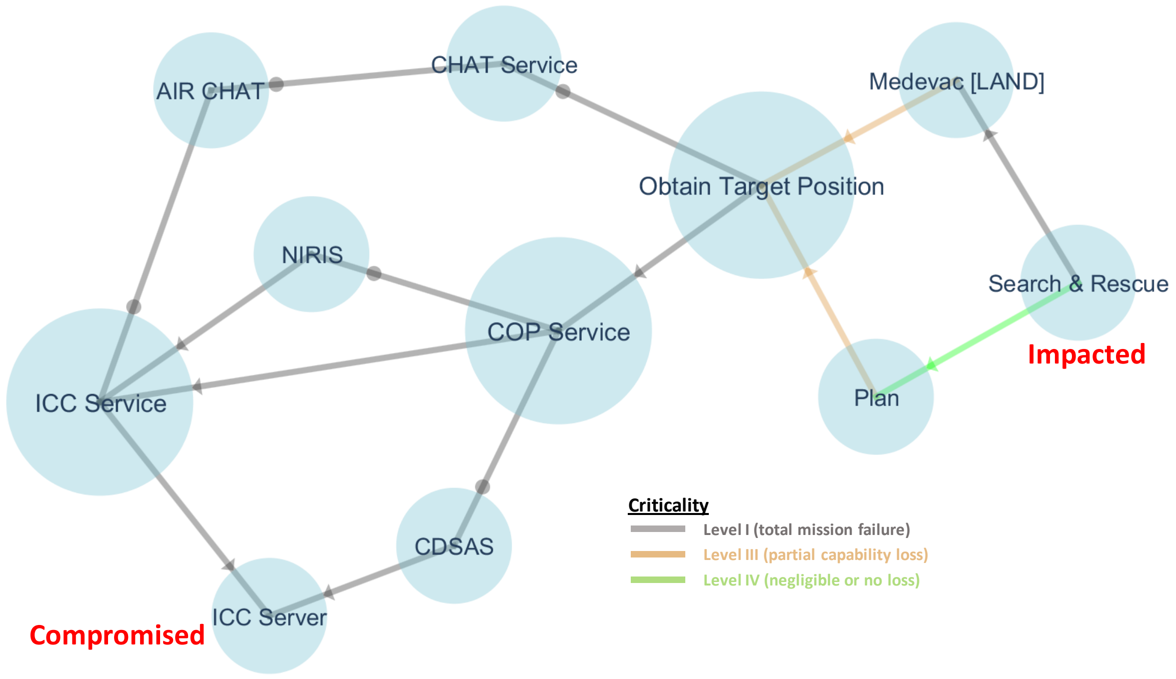

Figure 9: Mission Impact of the Loss of *ICC Server*.

5.0 SUMMARY AND CONCLUSIONS

In this paper, we describe a graph-based hierarchical model for dependency relationships among the elements of a military mission. This model captures interdependencies among mission objectives, tasks, information, and cyber assets. In building and analysing this model, we employ established MITRE tools and a structured methodology for cyber resiliency analysis. We express our model in the popular XML-based GraphML format, which we import into MITRE's CyGraph tool. This supports query-driven analytics and visualization, which we apply for analysing our model in terms of resilience and potential courses of action for response to cyberattack. Our model framework extends basic graph-theoretic properties, to include logical relationships and attributes such as ownership and criticality, which have significant influence on response options. We apply this modelling framework to a strategic-level scenario defined in a NATO Request for Information (RFI) in support of cyber defense situational awareness. This model will in turn be part of an upcoming NATO demonstration of situational awareness capabilities in a realistic military environment.

6.0 ACKNOWLEDGEMENTS

This work was funded in part by the MITRE Innovation Program (project number EPF-14-00341), with George Roelke as Cybersecurity Innovation Area Leader.

7.0 REFERENCES

[1] T. Moye, R. Sawilla, R. Sullivan, P. Lagadec, *Cyber Defence Situational Awareness Demonstration / Request for Information (RFI) from Industry and Government (CO-14068-MNCD2)*, NATO NCI Agency Acquisition, 2015.

[2] D. Bodeau, R. Graubart, *Structured Cyber Resiliency Analysis Methodology (SCRAM)*, 2016. https://www.mitre.org/sites/default/files/publications/pr-16-0777-structured-cyber-resiliency-analysis-methodology-overview.pdf.

[3] The MITRE Corporation, *Crown Jewels Analysis*, September 2013. http://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/crown-jewels-analysis.

[4] The MITRE Corporation, *Cyber Command System (CyCS)*, http://www.mitre.org/research/technology-transfer/technology-licensing/cyber-command-system-cycs.

[5] S. Noel, E. Harley, K. H. Tam, G. Gyor, "Big-Data Architecture for Cyber Attack Graphs: Representing Security Relationships in NoSQL Graph Databases," IEEE Symposium on Technologies for Homeland Security, Boston, Massachusetts, 2015.

[6] S. Noel, E. Harley, K. H. Tam, M. Limiero, M. Share, "CyGraph: Graph-Based Analytics and Visualization for Cybersecurity," in *Cognitive Computing: Theory and Applications* (Volume 35 of Handbook of Statistics), Elsevier, 2016.

[7] Defense Acquisition University, *Defense Acquistions Guidebook (DAG)*, https://dag.dau.mil/Pages/Default.aspx.

[8] Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)) and Department of Defense Chief Information Officer (DoD CIO), *Trusted Systems and Networks (TSN) Analysis*, June 2014. http://www.acq.osd.mil/se/docs/Trusted-Systems-and-Networks-TSN-Analysis.pdf.

[9] *Neo4j: The World's Leading Graph Database*, 2016. https://neo4j.com/.

[10] I. Robinson, J. Webber, E. Eifrem, *Graph Databases*, second edition, O'Reilly, 2015.

[11] O. Panzarino, *Learning Cypher*, Packt Publishing, 2014.