LOCKHEED MARTIN

Data Dependency Network Analysis in Systems of Systems

May 1, 2018

Reggie Cole

Senior Fellow

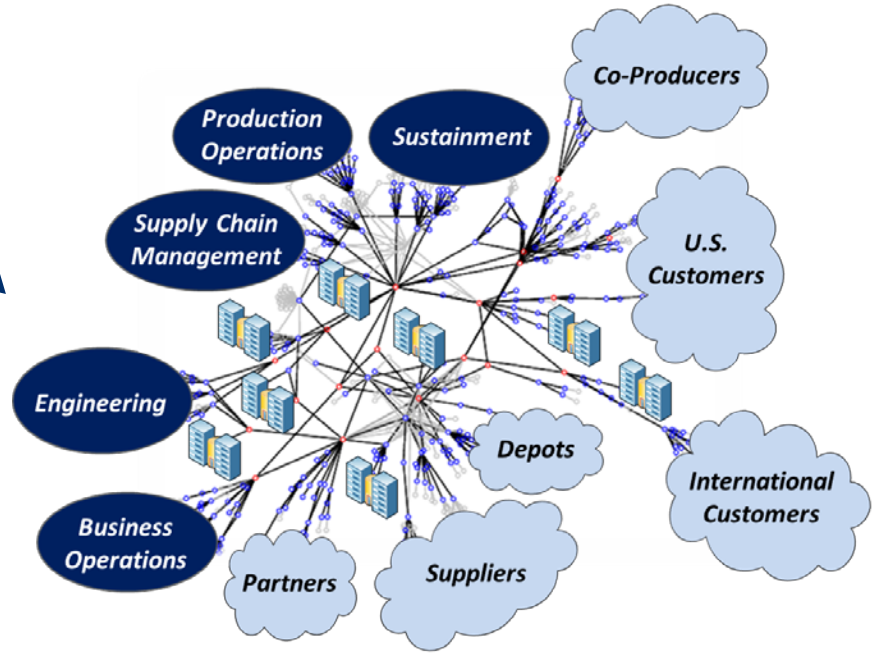
Lockheed Martin Corporation

reggie.cole@lmco.com

Messy Data Has Been Around for a Long Time – But the Problem is Particularly Acute for SoS & IoT

Messy Data
+
System-of-System
+
Internet of Things

Extreme Challenge



In 1963, Hafley and Lewis first discussed the analysis of messy data, which they defined simply as data that cannot be effectively consumed and used in the intended way.

The last 50 years has seen a continual growth in data, across an increasingly complex set of systems and enterprises.

This inherent messiness is particularly problematic for SoS & IoT, where constituent systems produce data for purposes that might be different than the intended uses of consuming systems.

So, how can we start to get our hands around this problem?

Messy Data in SoS & IoT — Effects & Mitigation

	Data Perspective	User Perspective
Context Independent	Spelling / Formatting Error Missing / Duplicated Data Incorrect / Unallowable Value Outdated Data Syntax / Constraint Violation	Inaccessible Information Insecure Information Irretrievable Information Difficult to Aggregate Transformation Errors
Context Dependent	Domain Constraint Violation Business Rule Violation Company Policy Violation Regulatory Violation Local Constraint Violation	Unreliable Information Irrelevant Information Inconsistent Information Incomplete Information Difficult to Process Difficult to Understand

Two Key Mitigation Strategies

Govern Quality of Outgoing Data

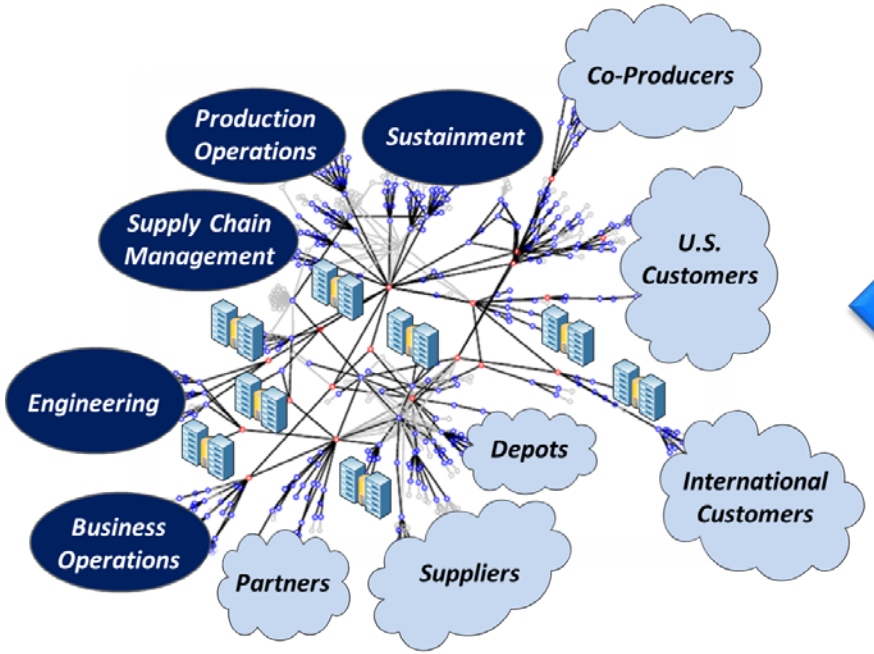
Cleanse Incoming Data

Messy Data is a Key Source of System Degradation

But… What is the right mix of cleansing and governance, and how should the strategies be applied to a particular SoS?

How Can We Apply Modeling to the Problem?

Modeling Question #1 — Given a specific SoS architecture, what are the effects of messy data on the SoS?

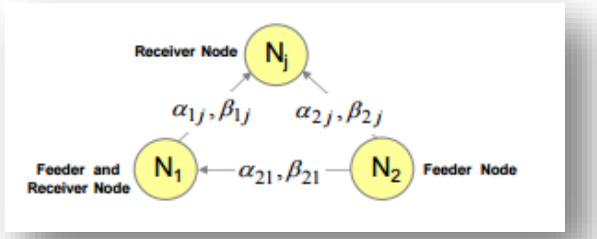


We can apply Data Dependency Network Analysis (DDNA)

Modeling Question #2 — How should we apply governance and cleansing strategies to optimize cost and performance?

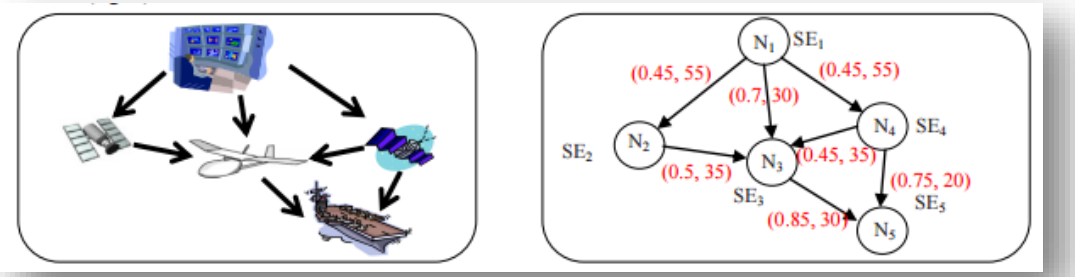
Development of the DDNA Model

Garvey and Pinto established the foundation for functional dependency network analysis in their 2009 paper.



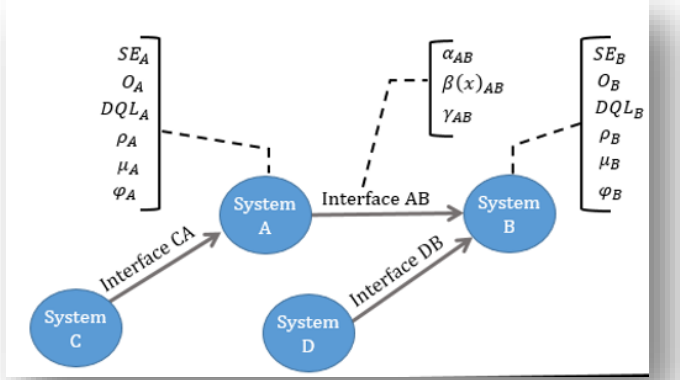
Adds the notion of self-effectiveness and applies the method to systems of systems

Guariniello and DeLaurentis applied FDNA to systems-of-systems operability in their 2013 paper.

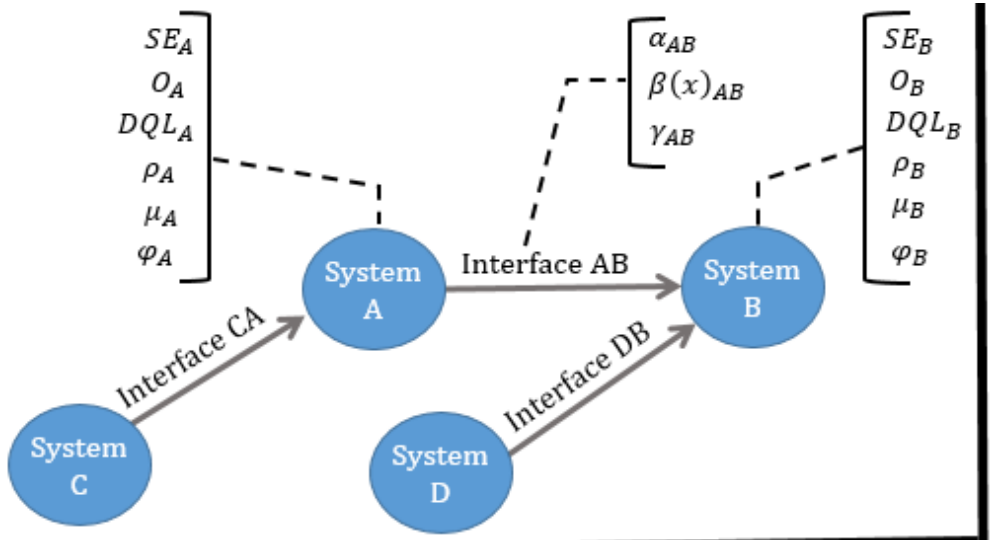


Applies the model to data dependency and data quality level of provider

This paper extends the FDNA approach to handle data dependencies and their effect on system-of-systems operability.



Model Details — Variables & State Transition



$SE = System\ Self\ Effectiveness\ (From\ 0\ to\ 100)$
$O = System\ Operability\ (From\ 0\ to\ 100)$
$DQL = Output\ Data\ Quality\ Level\ (From\ 0\ to\ 100)$
$\rho = Incoming\ Data\ Cleansing\ Effectiveness\ (From\ 0\ to\ 1.0)$
$\mu = Data\ Governance\ Effectiveness\ (From\ 0\ to\ 1.0)$
$\varphi = Operability\ Strength\ (From\ 0\ to\ 1.0)$
$\alpha = Strength\ of\ Data\ Dependency\ (From\ 0\ to\ 1.0)$
$\gamma = Contextual\ Alignment\ Factor\ (From\ 0\ to\ 1.0)$

$\beta(x) = Criticality\ of\ Dependency$

$$\beta(x) = \begin{cases} 0\ if\ Provider\ Operability < x \\ 1\ if\ Provider\ Operability \geq x \end{cases}$$

Where:
$x = Minimum\ Operability\ Threshold\ for\ Provider\ System$

For systems with no dependencies:

$$O = SE$$

Since the system has no external dependencies, the system is in complete control of its own operability. In the diagram, the operability for System C and System D are governed by this equation.

For systems with N dependencies (where N > 0):

$$O = \prod_{i=1}^{N} \beta(x)_i \cdot \left[\sum_{i=1}^{N} \left(\frac{\alpha_i \cdot [\gamma_i \cdot DQL_i + \rho \cdot (100 - \gamma_i \cdot DQL_i)]}{N} \right) + SE \cdot \left(1 - \sum_{i=1}^{N} \frac{\alpha_i}{N} \right) \right]$$

In the diagram, the operability of System B is a factor of its own self effectiveness, the quality of data being provided, the strength of dependency on the data being provided, and whether any of its predecessors have an operability below their respective operability thresholds.

For each system the quality of data being provided (DQL) is a factor of its own operability and the effectiveness of its own data governance.

$$DQL = \varphi \cdot O + \mu \cdot (100 - \varphi \cdot O)$$

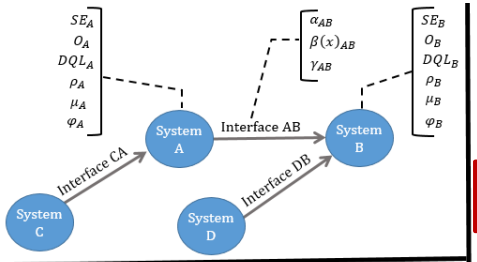
The *Total System Performance* of the system-of-systems with M constituent systems is calculated as follows:

$$Performance = \sum_{k=1}^{M} w_k \cdot O_k$$

Where:

$$w_k = Performance\ Contribution\ of\ System\ k$$

Model Details — System Operability



SE = *System Self Effectiveness* (*From* 0 *to* 100)
O = *System Operability* (*From* 0 *to* 100)
DQL = *Output Data Quality Level* (*From* 0 *to* 100)
ρ = *Incoming Data Cleansing Effectiveness* (*From* 0 *to* 1.0)
μ = *Data Governance Effectiveness* (*From* 0 *to* 1.0)
φ = *Operability Strength* (*From* 0 *to* 1.0)
α = *Strength of Data Dependency* (*From* 0 *to* 1.0)
γ = *Contextual Alignment Factor* (*From* 0 *to* 1.0)
$\beta(x)$ = *Criticality of Dependency*

$$\beta(x) = \begin{cases} 0 \text{ if Provider Operability} < x \\ 1 \text{ if Provider Operability} \geq x \end{cases}$$

Where:
x = *Minimum Operability Threshold for Provider System*

For systems with no dependencies:

$$O = SE$$

Since the system has no external dependencies, the system is in complete control of its own operability. In the diagram, the operability for System C and System D are governed by this equation.

For systems with N dependencies (where N > 0):

$$O = \prod_{i=1}^{N} \beta(x)_i \cdot \left[\sum_{i=1}^{N} \left(\frac{\alpha_i \cdot [\gamma_i \cdot DQL_i + \rho \cdot (100 - \gamma_i \cdot DQL_i)]}{N} \right) + SE \cdot \left(1 - \sum_{i=1}^{N} \frac{\alpha_i}{N} \right) \right]$$

In the diagram, the operability of System B is a factor of its own self effectiveness, the quality of data being provided, the strength of dependency on the data being provided, and whether any of its predecessors have an operability below their respective operability thresholds.

For each system the quality of data being provided (DQL) is a factor of its own operability and the effectiveness of its own data governance.

$$DQL = \varphi \cdot O + \mu \cdot (100 - \varphi \cdot O)$$

The *Total System Performance* of the system-of-systems with M constituent systems is calculated as follows:

$$Performance = \sum_{k=1}^{M} w_k \cdot O_k$$

Where:

$$w_k = Performance\ Contribution\ of\ System\ k$$

System Operability **is the key observable state. It is calculated for each node based on:**

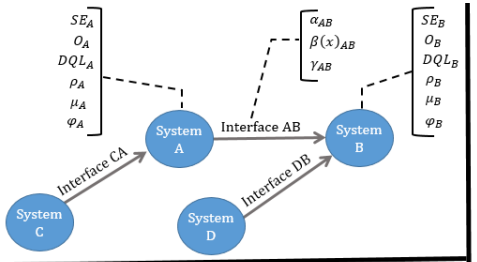
SE = *System Self Effectiveness* (*From* 0 *to* 100)
DQL = *Output Data Quality Level* (*From* 0 *to* 100)
ρ = *Incoming Data Cleansing Effectiveness* (*From* 0 *to* 1.0)
α = *Strength of Data Dependency* (*From* 0 *to* 1.0)
γ = *Contextual Alignment Factor* (*From* 0 *to* 1.0)
$\beta(x)$ = *Criticality of Dependency*

$$\beta(x) = \begin{cases} 0 \text{ if Provider Operability} < x \\ 1 \text{ if Provider Operability} \geq x \end{cases}$$

Where:
x = *Minimum Operability Threshold for Provider System*

$$O = \prod_{i=1}^{N} \beta(x)_i \cdot \left[\sum_{i=1}^{N} \left(\frac{\alpha_i \cdot [\gamma_i \cdot DQL_i + \rho \cdot (100 - \gamma_i \cdot DQL_i)]}{N} \right) + SE \cdot \left(1 - \sum_{i=1}^{N} \frac{\alpha_i}{N} \right) \right]$$

Model Details — Output Data Quality Level



$SE = System\ Self\ Effectiveness\ (From\ 0\ to\ 100)$
$O = System\ Operability\ (From\ 0\ to\ 100)$
$DQL = Output\ Data\ Quality\ Level\ (From\ 0\ to\ 100)$
$\rho = Incoming\ Data\ Cleansing\ Effectiveness\ (From\ 0\ to\ 1.0)$
$\mu = Data\ Governance\ Effectiveness\ (From\ 0\ to\ 1.0)$
$\varphi = Operability\ Strength\ (From\ 0\ to\ 1.0)$
$\alpha = Strength\ of\ Data\ Dependency\ (From\ 0\ to\ 1.0)$
$\gamma = Contextual\ Alignment\ Factor\ (From\ 0\ to\ 1.0)$
$\beta(x) = Criticality\ of\ Dependency$

$$\beta(x) = \begin{cases} 0\ if\ Provider\ Operability < x \\ 1\ if\ Provider\ Operability \geq x \end{cases}$$

Where:
$x = Minimum\ Operability\ Threshold\ for\ Provider\ System$

For systems with no dependencies:

$$O = SE$$

Since the system has no external dependencies, the system is in complete control of its own operability. In the diagram, the operability for System C and System D are governed by this equation.

For systems with N dependencies (where N > 0):

$$O = \prod_{i=1}^{N} \beta(x)_i \cdot \left[\sum_{i=1}^{N} \left(\frac{\alpha_i \cdot [\gamma_i \cdot DQL_i + \rho \cdot (100 - \gamma_i \cdot DQL_i)]}{N} \right) + SE \cdot \left(1 - \sum_{i=1}^{N} \frac{\alpha_i}{N} \right) \right]$$

In the diagram, the operability of System B is a factor of its own self effectiveness, the quality of data being provided, the strength of dependency on the data being provided, and whether any of its predecessors have an operability below their respective operability thresholds.

For each system the quality of data being provided (DQL) is a factor of its own operability and the effectiveness of its own data governance.

$$DQL = \varphi \cdot O + \mu \cdot (100 - \varphi \cdot O)$$

The *Total System Performance* of the system-of-systems with M constituent systems is calculated as follows:

$$Performance = \sum_{k=1}^{M} w_k \cdot O_k$$

Where:

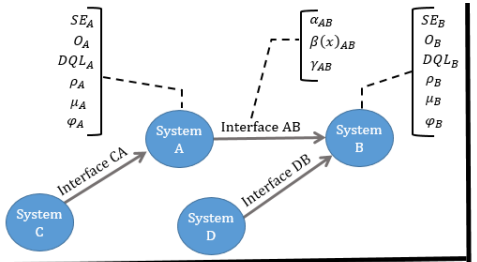
$w_k = Performance\ Contribution\ of\ System\ k$

Then the output *Data Quality Level* is calculated based on:

$O = System\ Operability\ (From\ 0\ to\ 100)$
$\mu = Data\ Governance\ Effectiveness\ (From\ 0\ to\ 1.0)$
$\varphi = Operability\ Strength\ (From\ 0\ to\ 1.0)$

$$DQL = \varphi \cdot O + \mu \cdot (100 - \varphi \cdot O)$$

Model Details — Overall SoS Performance



SE = *System Self Effectiveness* (*From* 0 *to* 100)
O = *System Operability* (*From* 0 *to* 100)
DQL = *Output Data Quality Level* (*From* 0 *to* 100)
ρ = *Incoming Data Cleansing Effectiveness* (*From* 0 *to* 1.0)
μ = *Data Governance Effectiveness* (*From* 0 *to* 1.0)
φ = *Operability Strength* (*From* 0 *to* 1.0)
α = *Strength of Data Dependency* (*From* 0 *to* 1.0)
γ = *Contextual Alignment Factor* (*From* 0 *to* 1.0)

$\beta(x)$ = *Criticality of Dependency*

$$\beta(x) = \begin{cases} 0 \text{ if Provider Operability} < x \\ 1 \text{ if Provider Operability} \geq x \end{cases}$$

Where:
x = *Minimum Operability Threshold for Provider System*

For systems with no dependencies:

$$O = SE$$

Since the system has no external dependencies, the system is in complete control of its own operability. In the diagram, the operability for System C and System D are governed by this equation.

For systems with N dependencies (where N > 0):

$$O = \prod_{i=1}^{N} \beta(x)_i \cdot \left[\sum_{i=1}^{N} \left(\frac{\alpha_i \cdot [\gamma_i \cdot DQL_i + \rho \cdot (100 - \gamma_i \cdot DQL_i)]}{N} \right) + SE \cdot \left(1 - \sum_{i=1}^{N} \frac{\alpha_i}{N} \right) \right]$$

In the diagram, the operability of System B is a factor of its own self effectiveness, the quality of data being provided, the strength of dependency on the data being provided, and whether any of its predecessors have an operability below their respective operability thresholds.

For each system the quality of data being provided (DQL) is a factor of its own operability and the effectiveness of its own data governance.

$$DQL = \varphi \cdot O + \mu \cdot (100 - \varphi \cdot O)$$

The *Total System Performance* of the system-of-systems with M constituent systems is calculated as follows:

$$Performance = \sum_{k=1}^{M} w_k \cdot O_k$$

Where:

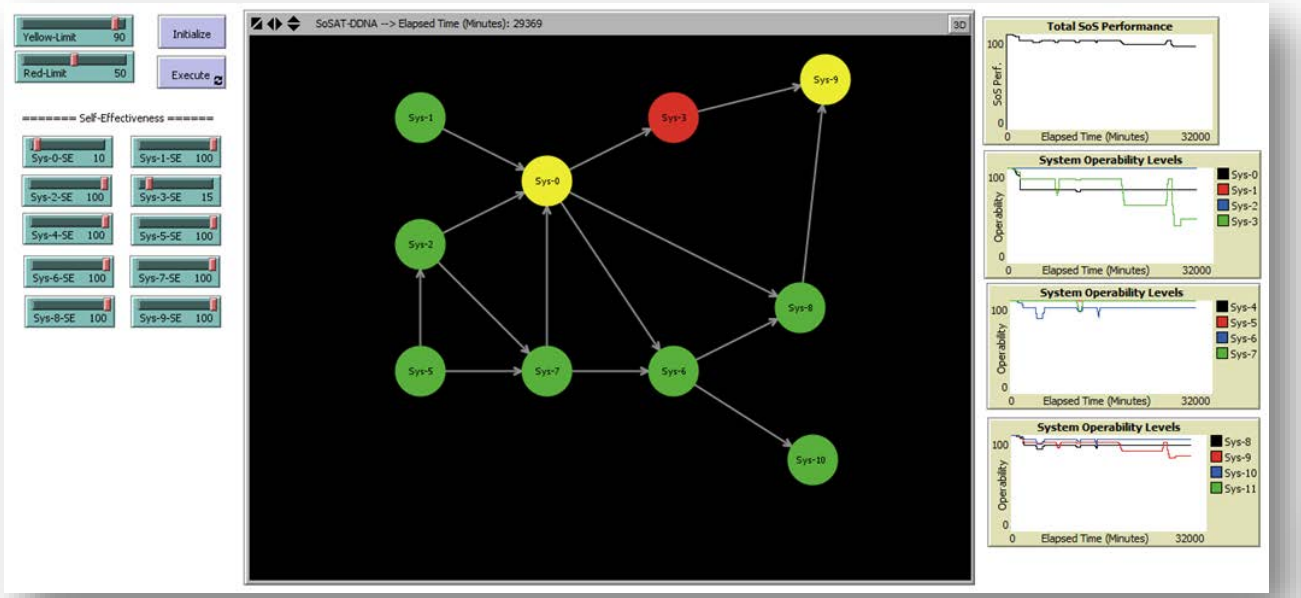
w_k = *Performance Contribution of System k*

Finally, the overall SoS performance is calculated based on the system operability of each node in the system

$$Performance = \sum_{k=1}^{M} w_k \cdot O_k$$

$$w_k = Performance\ Contribution\ of\ System\ k$$

NetLogo Implementation of the DDNA Model



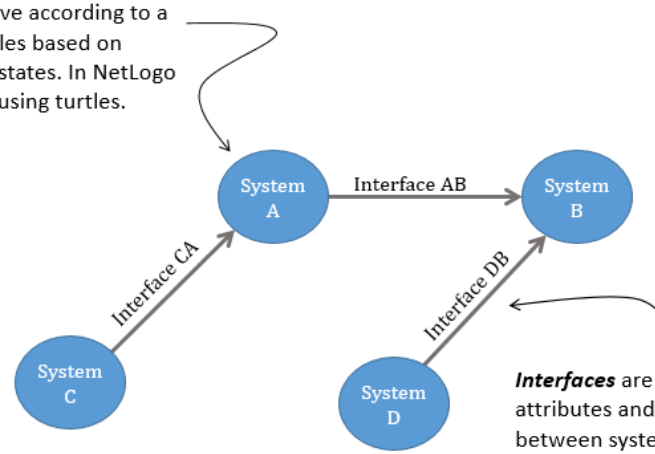
NetLogo is an open source agent-based modeling tool based on turtles, patches and links:

- **Turtles — Modeled Entities**
- **Patches — Location Entities**
- **Links — Relationship Entities**

How NetLogo Entities Are Used in DDNA:

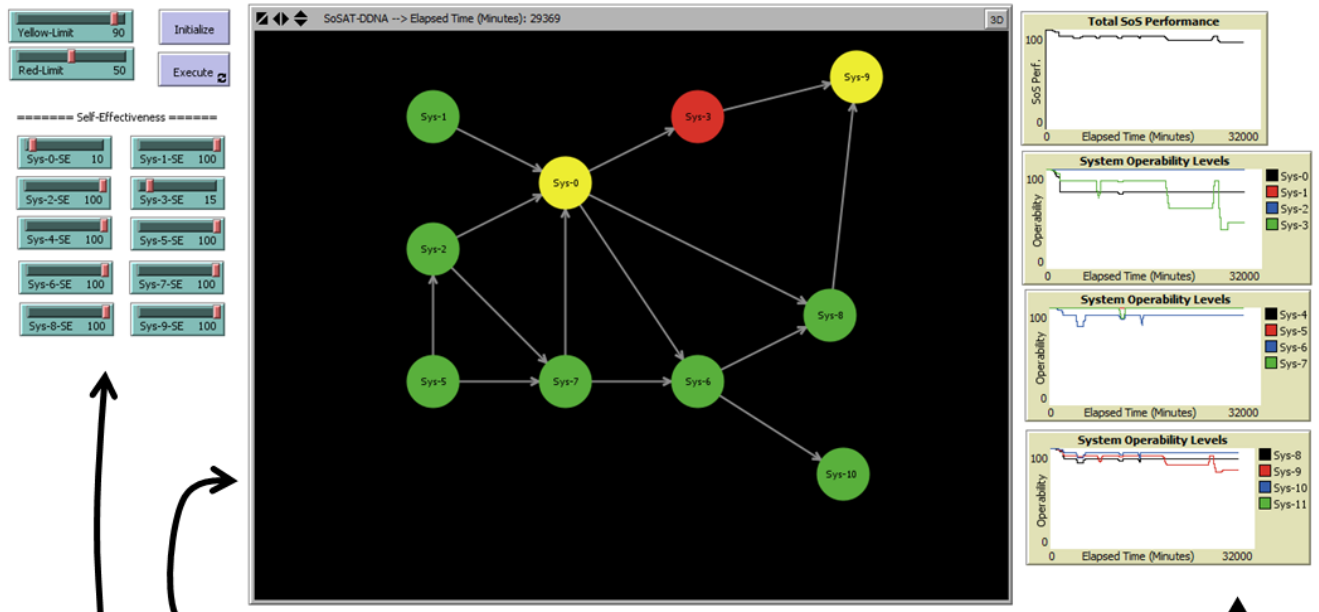
- **Turtles — Systems**
- **Patches — Not Used**
- **Links — Interfaces**

Systems are agents that have attributes and behave according to a prescribed set of rules based on inputs and internal states. In NetLogo these are modeled using turtles.



Interfaces are agents that have attributes and serve as data flows between systems. Their behaviors result from their two end points. In NetLogo they are modeled as directed links.

Putting it Together in the NetLogo Simulation

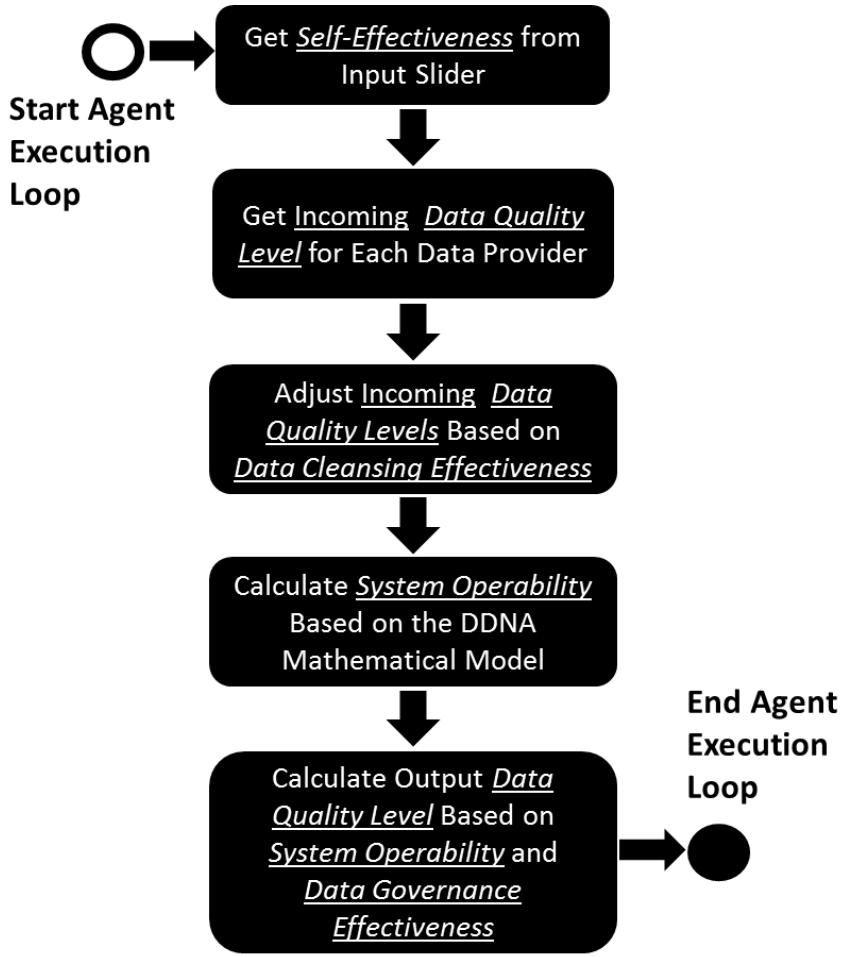


Colors indicate ***System Operability*** states (based on configurable red/yellow limits)

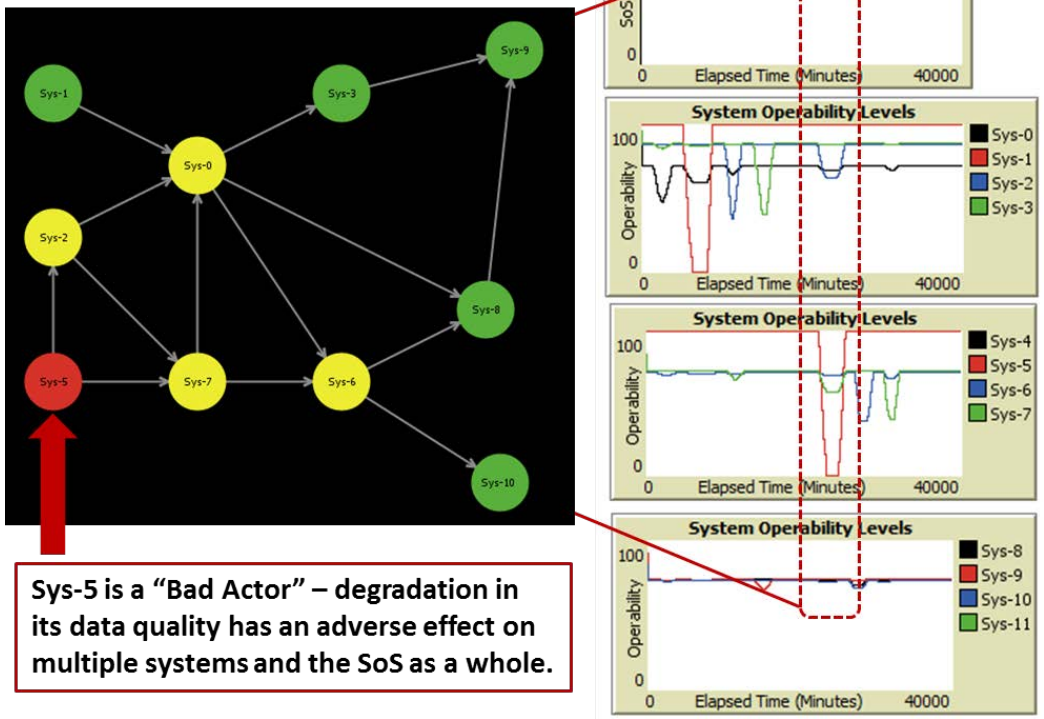
Analyst controls ***System Self-Effectiveness*** states using sliders

In addition to color indicators, the analyst monitors ***System Operability*** state histories using output collectors

Calculation of System States



Identification & Mitigation of “Bad Actors”

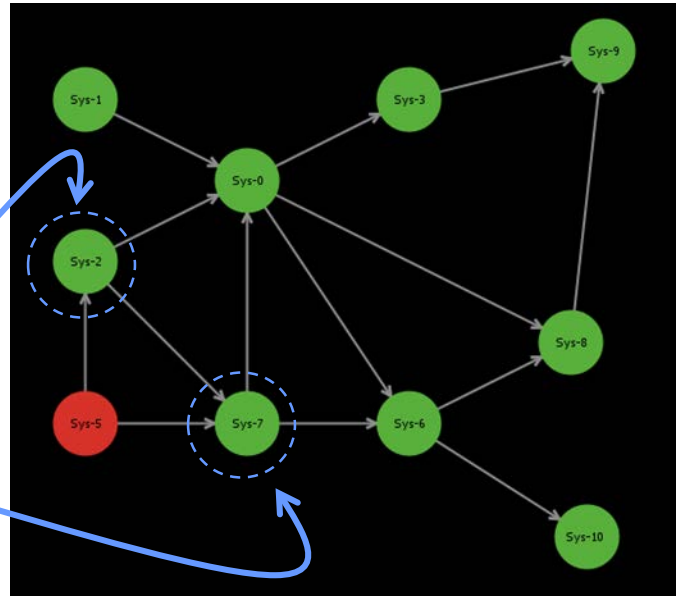


Sys-5 is a “Bad Actor” – degradation in its data quality has an adverse effect on multiple systems and the SoS as a whole.

As we adjust the ***Self Effectiveness*** of each system in the SoS, we are able to identify systems whose data quality level has the most profound effect on the SoS as a whole.

Then we can start to think about modeling mitigation strategies for addressing this SoS “architectural weakness.”

By improving the data cleansing capabilities in just two systems, we see an overall mitigation of the problem.



Summary

- **Messy data is a big problem for systems of systems**
- **There are strategies for mitigating its effects**
- **The DDNA model provides a tool for understanding the effect of messy data on systems of systems**
- **The DDNA model provides a tool for optimizing the strategy for mitigating the effect of messy data in systems of systems**