What is the Future of Systems Engineering?

STEVEN H. DAM, PH.D., ESEP

WARREN K. VANEMAN, PH.D. CAPT, USN (RET)

OCTOBER 25, 2018

SPEC INNOVATIONS

Agenda

- Why this Question?
- History of Systems Engineering
- What Do We Need that We Are Missing?
- Why Isn’t SysML Enough?
- How Will Technology Improvements Enable Better SE?
- Why Do We Have to Wait 10 Years?

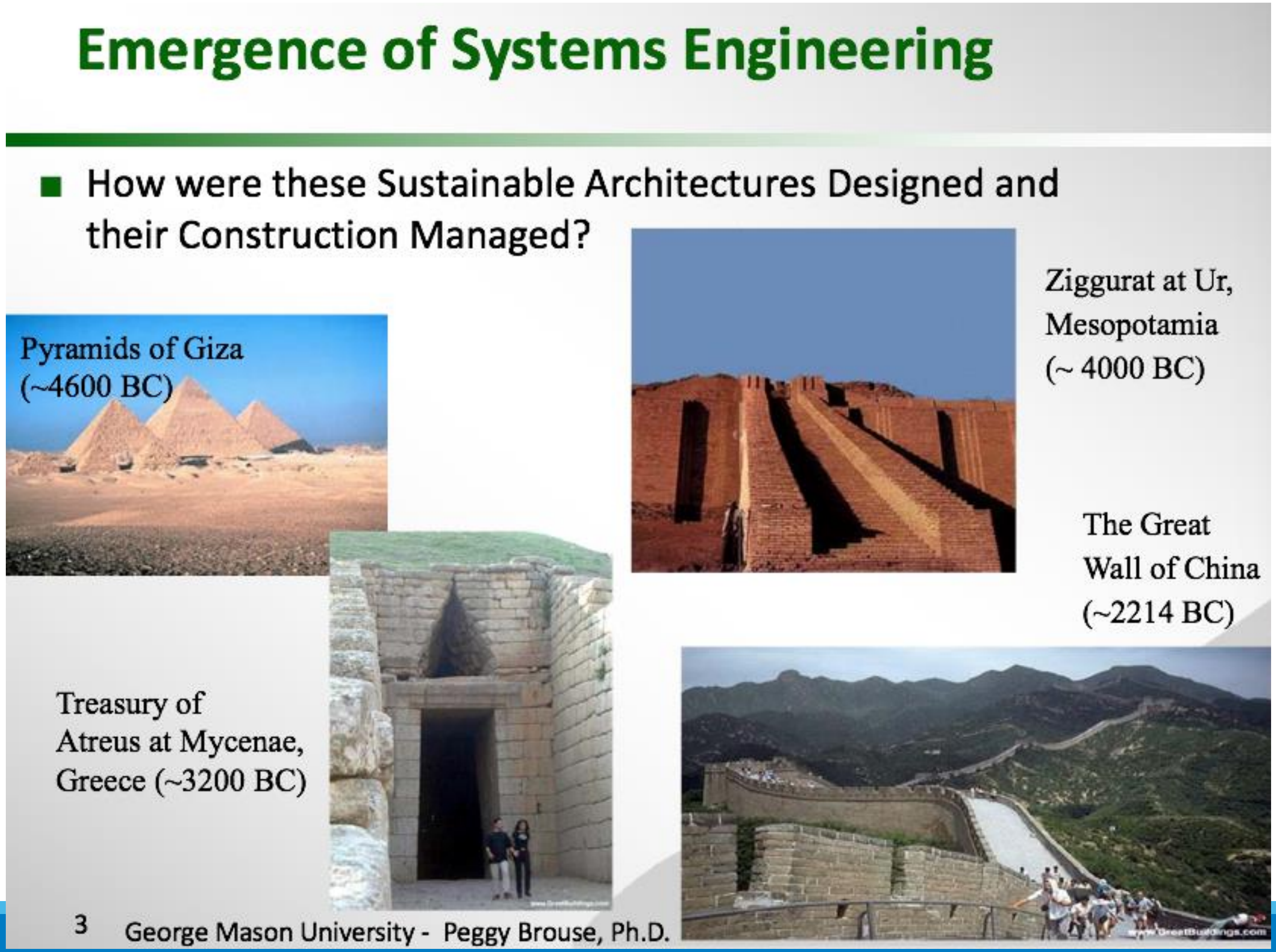
Why this Question?

- In a Spring 2018 Systems Engineering Forum, held by The Aerospace Corporation, one questioner stated,
  - “SysML is the current systems engineering, but 10 years from now it’s likely to be something else.”
- That begs the question, “What is the future of systems engineering?” and others …
  - What do we need that we are missing?
    - Why isn’t SysML enough?
  - How will technology improvements enable better SE?
  - Why do we have to wait 10 years?
  - How can we predict the future?

“It’s easy to predict the future when we have lived it so many times in the past!” S.H.Dam, 2018

History of Systems Engineering

- Some believe systems engineering can be traced back thousands of years
  - Wonders of the world could only have been designed and built systematically



Emergence of Systems Engineering

- How were these Sustainable Architectures Designed and their Construction Managed?

Pyramids of Giza (~4600 BC)

Ziggurat at Ur, Mesopotamia (~ 4000 BC)

The Great Wall of China (~2214 BC)

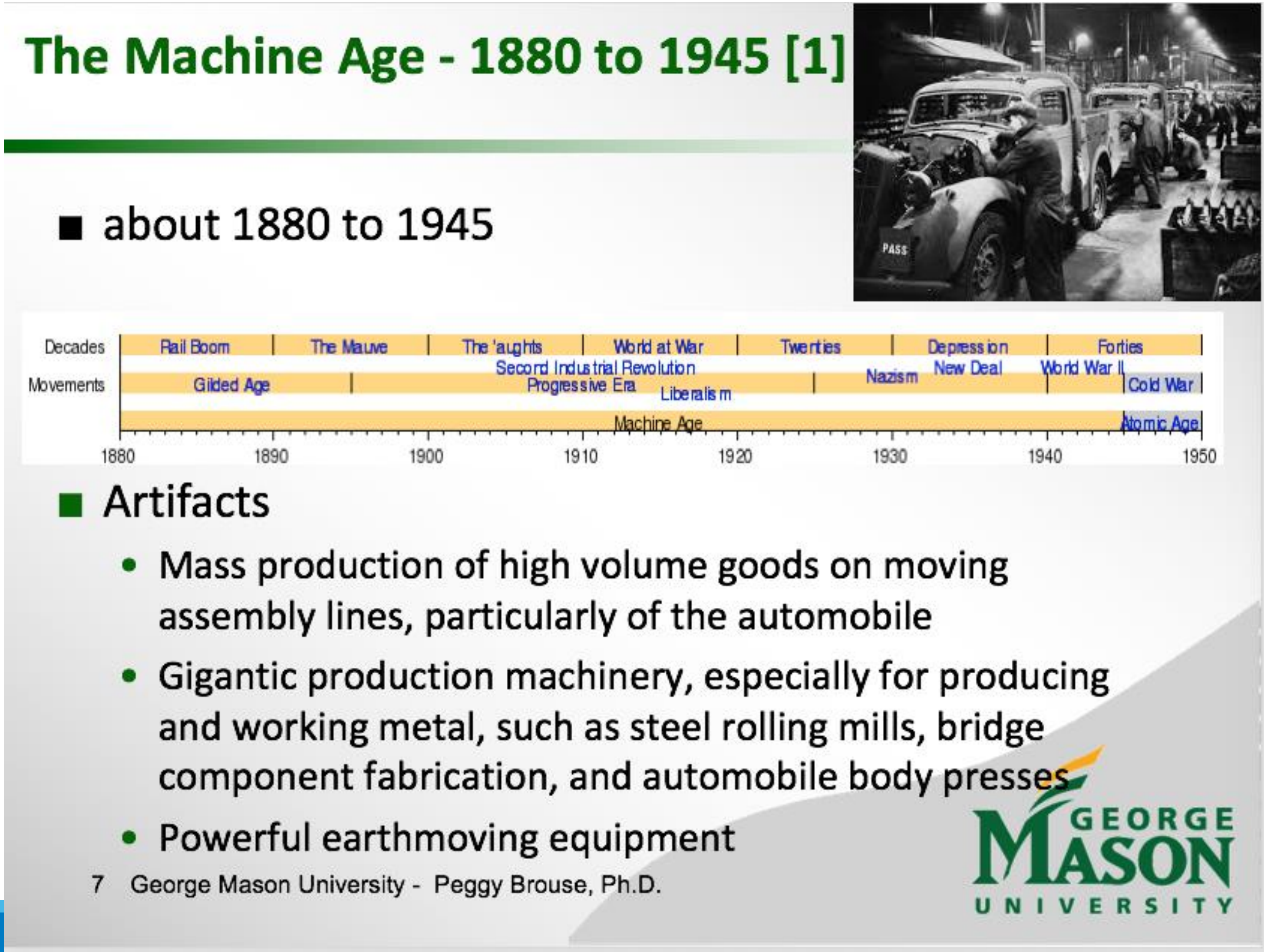
Treasury of Atreus at Mycenae, Greece (~3200 BC)

3 George Mason University - Peggy Brouse, Ph.D.

Slide from SYST 505 Course at GMU by Peggy Brouse, Ph.D.

History of SE *(continued)*

- Others trace it back to the "Machine Age"
  - Clearly the industrial revolution and the assembly lines require systems thinking



The Machine Age - 1880 to 1945 [1]

- about 1880 to 1945

Decades
Rail Boom
The Mauve
The 'aughts
World at War
Twenties
Depression
Forties
Movements
Gilded Age
Second Industrial Revolution
Progressive Era
Liberalism
Nazism
New Deal
World War II
Cold War
Machine Age
Atomic Age
1880
1890
1900
1910
1920
1930
1940
1950
PASS

- Artifacts
  - Mass production of high volume goods on moving assembly lines, particularly of the automobile
  - Gigantic production machinery, especially for producing and working metal, such as steel rolling mills, bridge component fabrication, and automobile body presses
  - Powerful earthmoving equipment

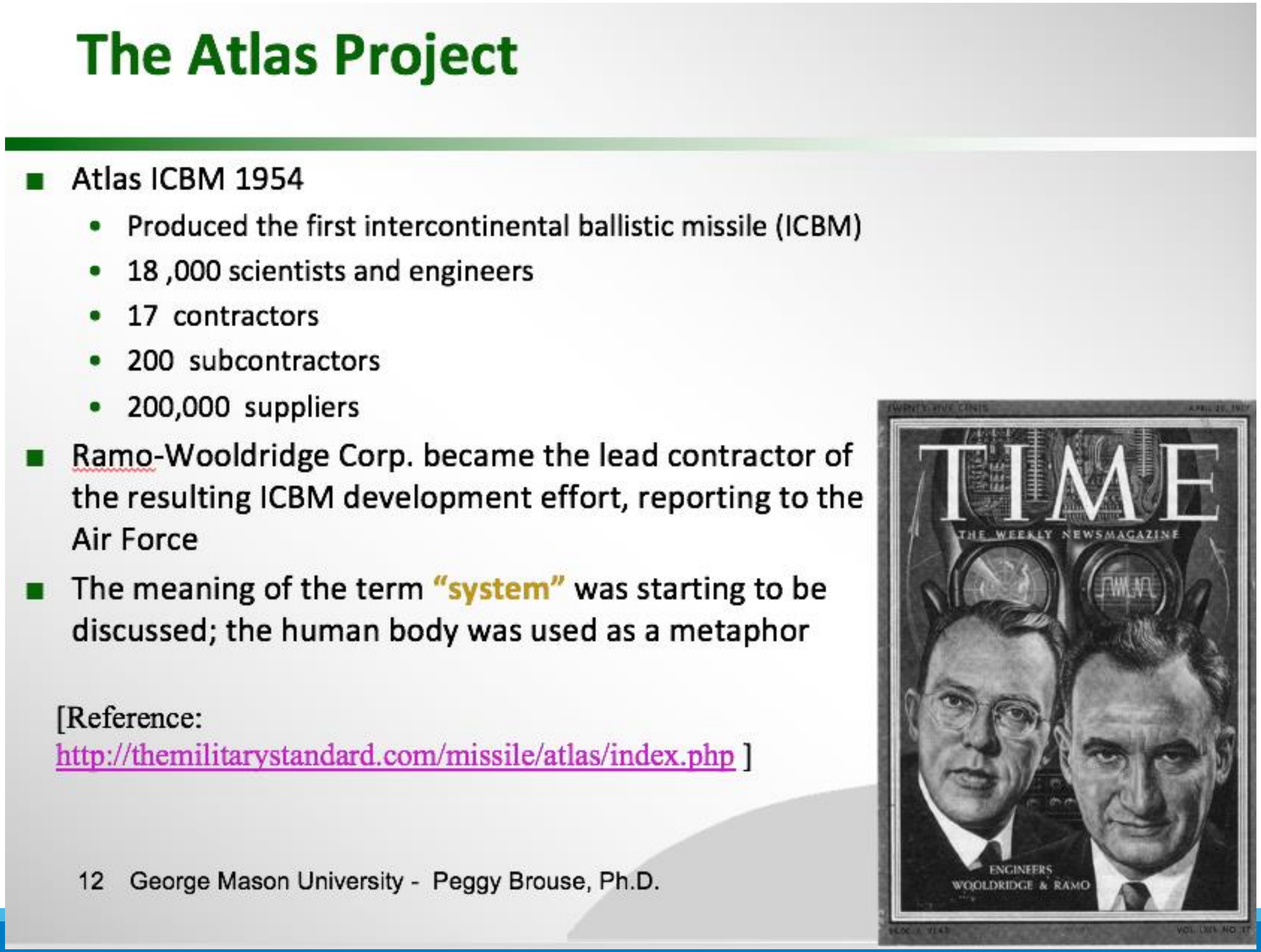
7 George Mason University - Peggy Brouse, Ph.D.

GEORGE MASON UNIVERSITY

Slide from SYST 505 Course at GMU by Peggy Brouse, Ph.D.

History of SE *(continued)*

- But by the "Space Age" systems engineering as we know it was clearly born
  - Millions of parts, clearly "systems of systems" thinking was required by this time



The Atlas Project

- Atlas ICBM 1954
  - Produced the first intercontinental ballistic missile (ICBM)
  - 18 ,000 scientists and engineers
  - 17 contractors
  - 200 subcontractors
  - 200,000 suppliers
- Ramo-Wooldridge Corp. became the lead contractor of the resulting ICBM development effort, reporting to the Air Force
- The meaning of the term **"system"** was starting to be discussed; the human body was used as a metaphor

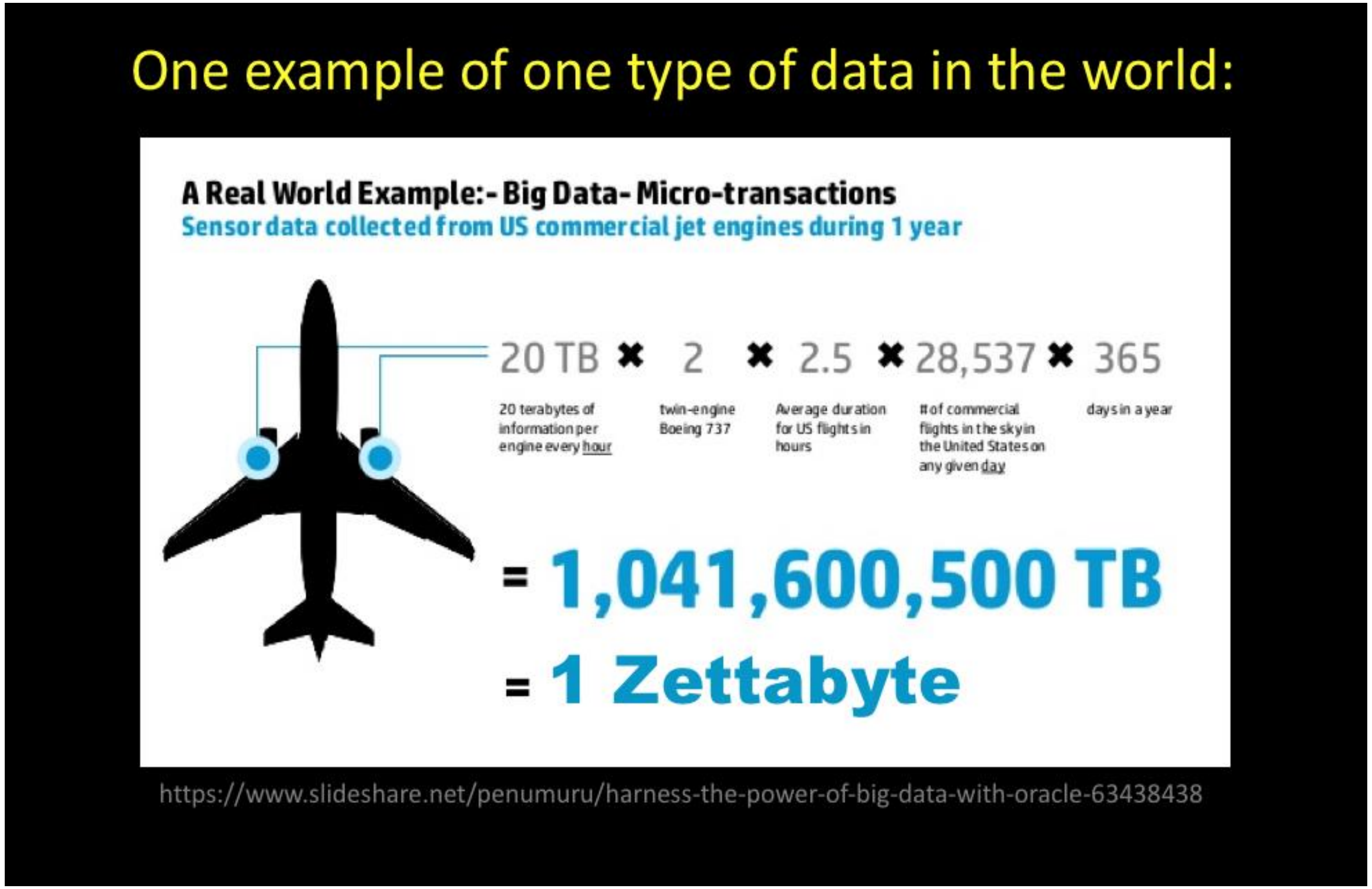
[Reference: http://themilitarystandard.com/missile/atlas/index.php]

12 George Mason University - Peggy Brouse, Ph.D.

Slide from SYST 505 Course at GMU by Peggy Brouse, Ph.D.

History of SE *(continued)*

- Today, complexity is going out of sight
- We no longer talk about Gigabytes of information, its now Zettabytes (1 x 10^{21} bytes)
- How can we deal with this much data?



One example of one type of data in the world:

A Real World Example:- Big Data- Micro-transactions

Sensor data collected from US commercial jet engines during 1 year

20 TB ✖ 2 ✖ 2.5 ✖ 28,537 ✖ 365

20 TB	2	2.5	28,537	365
20 terabytes of information per engine every hour	twin-engine Boeing 737	Average duration for US flights in hours	# of commercial flights in the sky in the United States on any given day	days in a year

= 1,041,600,500 TB

= 1 Zettabyte

https://www.slideshare.net/penumuru/harness-the-power-of-big-data-with-oracle-63438438

Slide from "Using Analytics to Predict and to Change the Future" presentation by Dr. Kirk Borne, BAH

How Have Our Languages Evolved Over the Last 60 Years?

- 1960s – used flow charting techniques derived from software (SREM created for software and systems engineering)
- 1970 –Data Flow Diagramming – heavily influenced by software development
- 1980s - IDEF, State Machine modeling and Computer-Aided Systems Engineering tools (e.g., RDD-100)
- 1990s – eFFBDs and Object-Oriented Analysis and Design/UML - derived from software techniques
- 2000s – SysML: a profile on UML
- 2010s – Still SysML, but LML emerged derived from systems engineering techniques; LML version 1.1 included an ontology so that systems engineering can be performed at the system entity level, instead of at the diagram level

History of SE *(continued)*

Why do we always seem to be 10 years behind the software world?

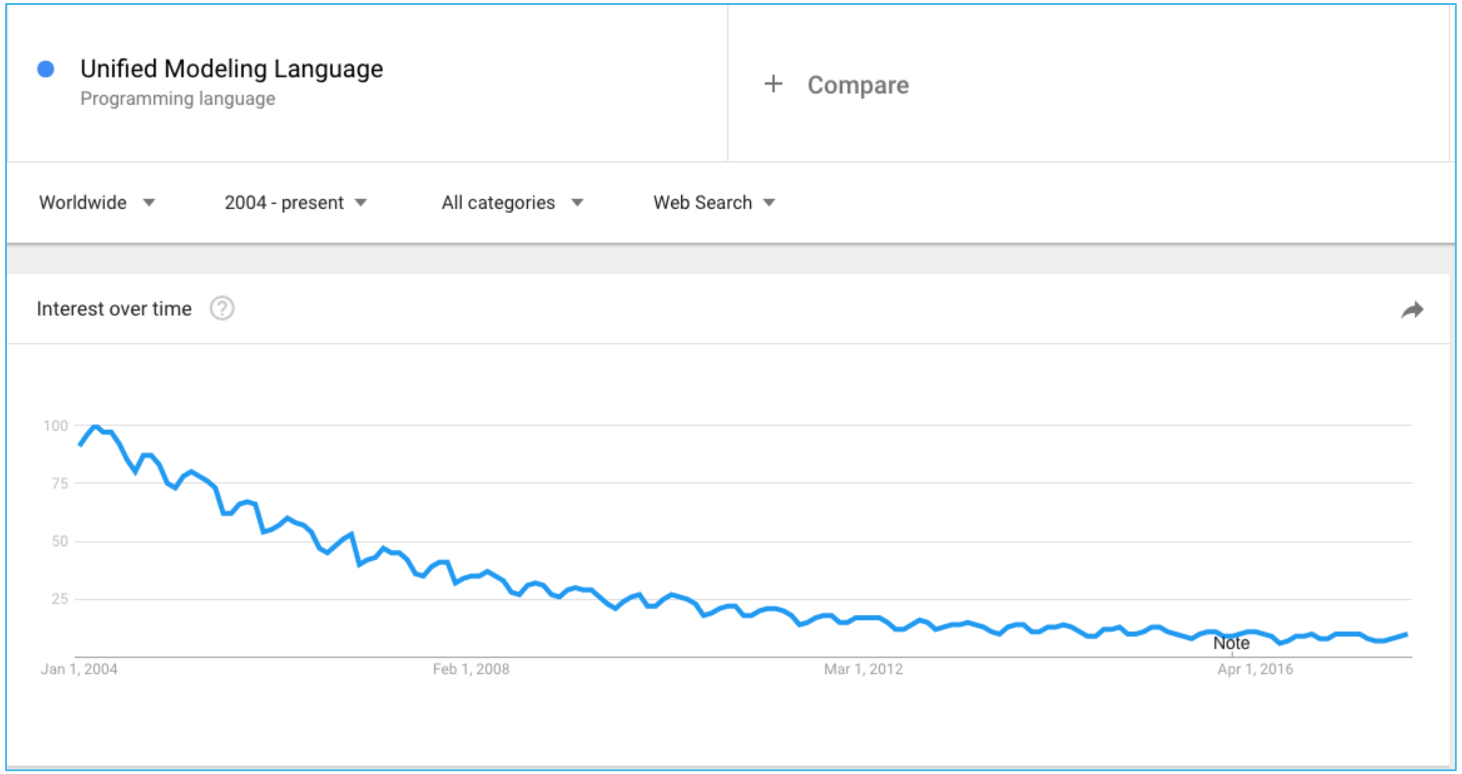
What Do We Need that We Are Missing?

- Need methods to capture and visualize tremendous amounts of information
- Massive storage and retrieval of information
- Need not only all the technical readouts, but also the programmatic information
- Capability to move data around easily, between applications
- A language that enables decomposition and abstraction
  - A systems engineering language, not a software engineering language
  - A language that is simple so that systems engineering can easily use it

But I know you are saying SysML does all this right?

Why Isn't SysML Enough?

- Systems Modeling Language was developed to extend the software focused Unified Modeling Language (UML) to systems
- Interest in UML peaked in 2004
- Software developers have moved on to Agile, which requires *functional requirements*
- Both SysML and UML require experts to create and interpret
- Systems Engineering requires communications with *all* stakeholders



● Unified Modeling Language
Programming language

+ Compare

Worldwide ▾ 2004 - present ▾ All categories ▾ Web Search ▾

Interest over time

From Google Trends retrieved 11/17/2017

If you have to be an expert in SysML's lexicon and diagram specifications, who are you communicating with?

Why Isn’t SysML Enough?

- But it’s worse than just not being easy to understand
- SysML is lacking many of the programmatic pieces of information: risk, issues, decisions, schedule, cost, … as explicit diagrams or entities
- The lack of an ontology has been noted and is in the process of being developed
- But what if there was already a language that provided an ontology for SysML and filled in the missing pieces?

There is: Lifecycle Modeling Language (LML). See it at www.lifecyclemodeling.org.

How Will Technology Improvements Enable Better SE?

- Some emerging/available technologies of the future:
  - Cloud computing (already here!)
  - Artificial Intelligence (Natural Language Process is already here!)
  - Graph Databases (already here!)
  - Optical Computing (coming soon)
- How can they help us?
  - Cloud computing provides a means to collaborate worldwide today … SE tools need to take advantage of this capability
  - Artificial Intelligence can help us find design problems or potential problems early
  - Graph Databases enable greater storage capacity
  - Optical Computing will enable create speed of computations, thus allowing for higher fidelity modeling and simulation

Why Do We Have to Wait 10 Years?

- We don't!
- As noted, many of these technologies exist today
- The Lifecycle Modeling Language (LML) provides a starting point for your language
  - It's an open standard, free for use
  - It's designed to be the "80%" solution
  - It's a simple language that can be extended it to meet your particular needs
- Innoslate® already uses cloud computing and AI (NLP) technologies and was designed to scale
- Other tools are beginning to realize these capabilities and are migrating to the cloud