The Past and the Future of the Semantic Web

A Personal Journey

Dr. Ora Lassila

Principal Technologist, Amazon Neptune

Co-chair, W3C RDF-star WG

Speaker Introduction

Current: Principal Technologist, Amazon Neptune

also: Co-chair of the W3C RDF-star Working Group

Past:

State Street, Pegasystems, Here, Nokia Ventures, …

Nokia Research, MIT, CMU, Helsinki Univ. of Technology

Education:

Ph.D CS, Helsinki Univ. of Technology

M.Sc CS + telecom, Helsinki Univ. of Technology

NCO, military communications + combat training, Finnish Armed Forces

Some items of note:

creator of the original Semantic Web vision

co-inventor of RDF

creator of KR software for NASA’s Deep Space 1

winner: Grand Prize, Usenix Obfuscated C Code Contest

founded “So Many Aircraft” (publishing + photography + illustration)

Game plan

1. The Semantic Web vision
2. A brief history of the Semantic Web
3. Where are we now? And how is any of this relevant to modern data practice?
4. The future of the Semantic Web

Disclaimer (of sort)

I am about to tell you a story

Some people will say "*that's not how it happened*"

To that, all I can say is "*I was there*"

Part 1: The Vision

The Semantic Web vision

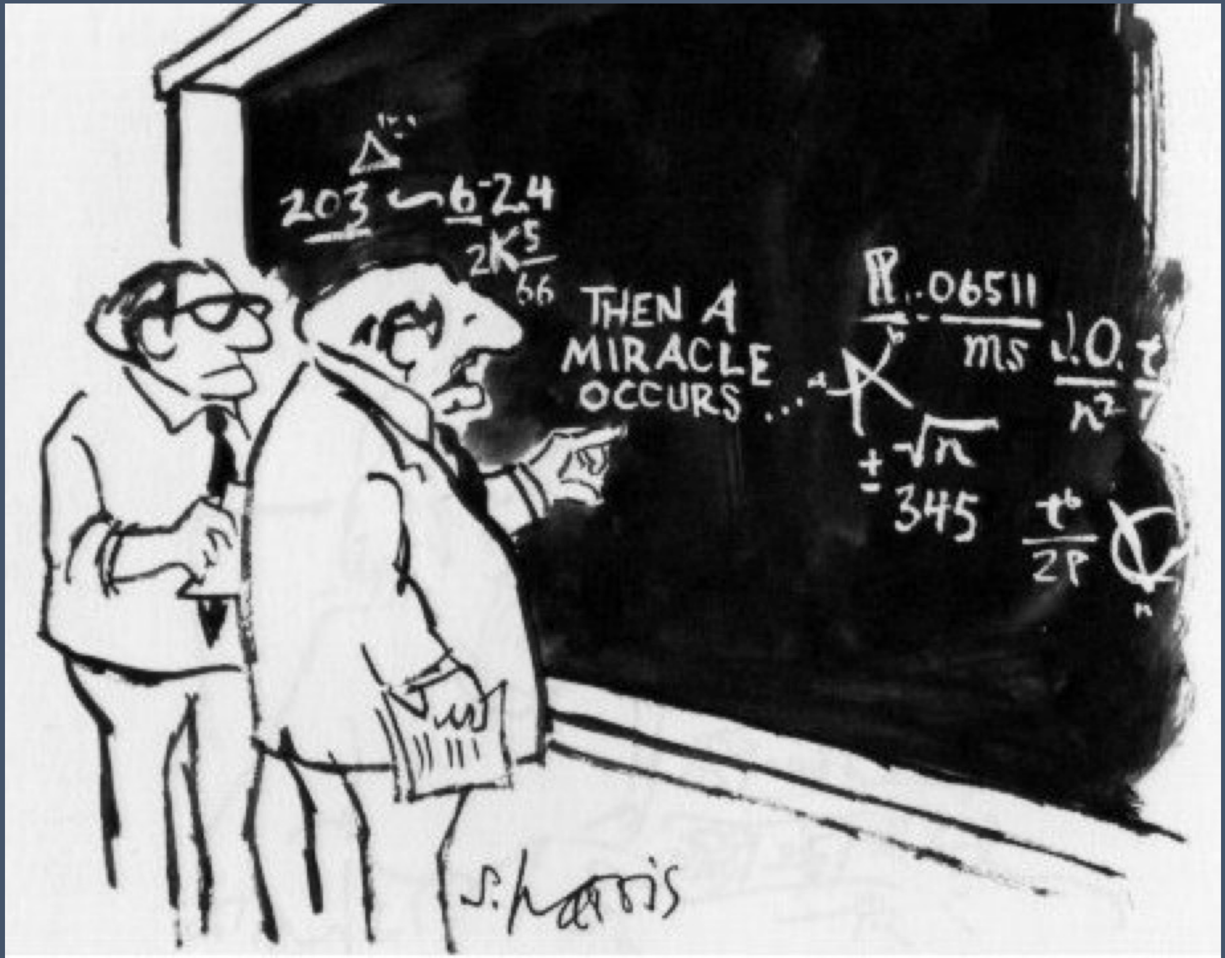
Years in the making, eventually published in 2001

Shifts emphasis from “human-interpretable” to “machine-interpretable” Web content

It would be a mischaracterization to say it is only about the Web; it is about humans interacting with digital technologies and systems (including hardware), and enabling these systems to have more autonomy

Berners-Lee, Hendler & Lassila: “The Semantic Web”, Scientific American, May 2001

Developing the grand vision



"I think you should be more explicit here in step two."

from *What's so Funny about Science?* by Sidney Harris (1977)

Backlash after the publication of the vision

"Science fiction", "unrealistic", etc.

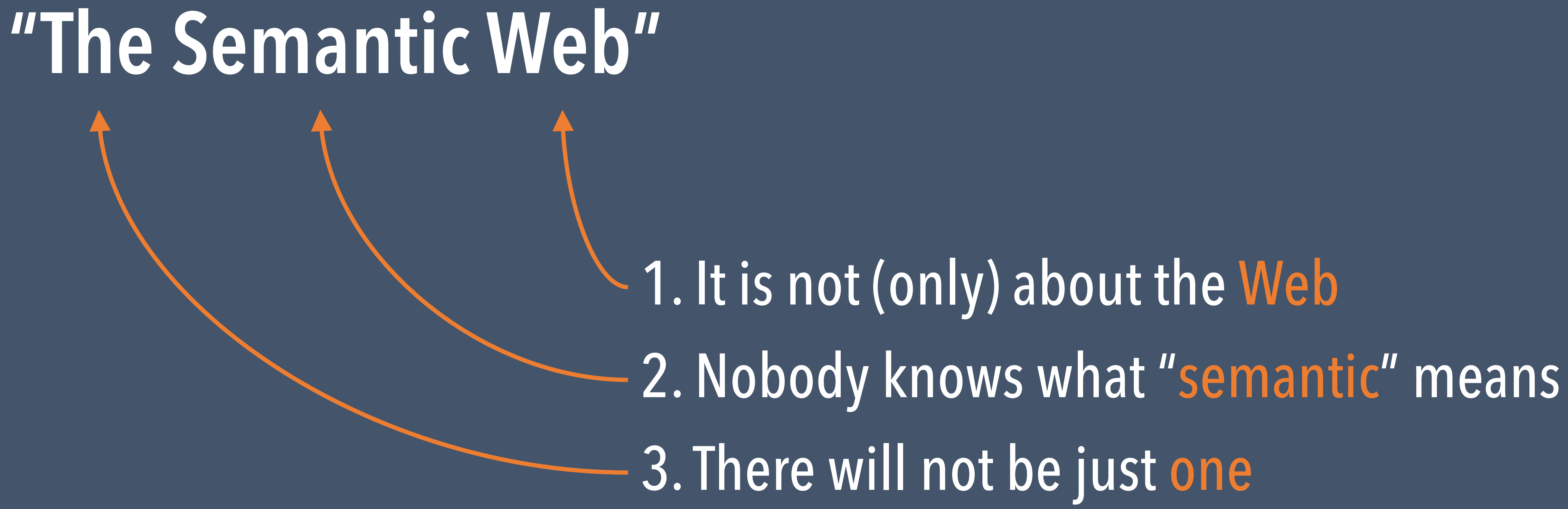
Questions about where the data would come from

Not taken seriously by KR people or DB people

Nevertheless, most of the predictions of the article have been realized

- (more about this later)

There might be three things wrong with the term



"The Semantic Web"

1. It is not (only) about the Web
2. Nobody knows what "semantic" means
3. There will not be just one

This is the essence, though

"The Semantic Web"

adjective

of, relating to, or arising from the different **meanings** of words or other symbols

(from dictionary.com)

The Semantic Web is about the meaning of information

The Semantic Web is often referred to as an effort to have "data on the Web"

This is not the key point

The key point is that is that for data, we want to have both the physical bits and the meaning of those bits, and we want to be able to share them

Ideally, when data moves, it should move with its semantics (rather than every system re-articulating semantics)

Deconstructing the vision

Traditional way of achieving interoperability is very "binary"

- things either work or they don't
- this is not how humans interact with one another; there is partial understanding and "middle ground"
- also, ask yourself this: "if my data model changes, how likely is it that I have to write new code?"

Traditionally, computers are tools

- (think of a hammer)
- Semantic Web, in my mind, was the first step in a paradigm shift where computers work on behalf of their human users

There is a difficult message…

Any specific problem (typically) has a specific solution that does not require Semantic Web technologies

Q: Why then is the Semantic Web so attractive?

A: For future-proofing

There is a difficult message…

Any specific problem (typically) has a specific solution that does not require Semantic Web technologies

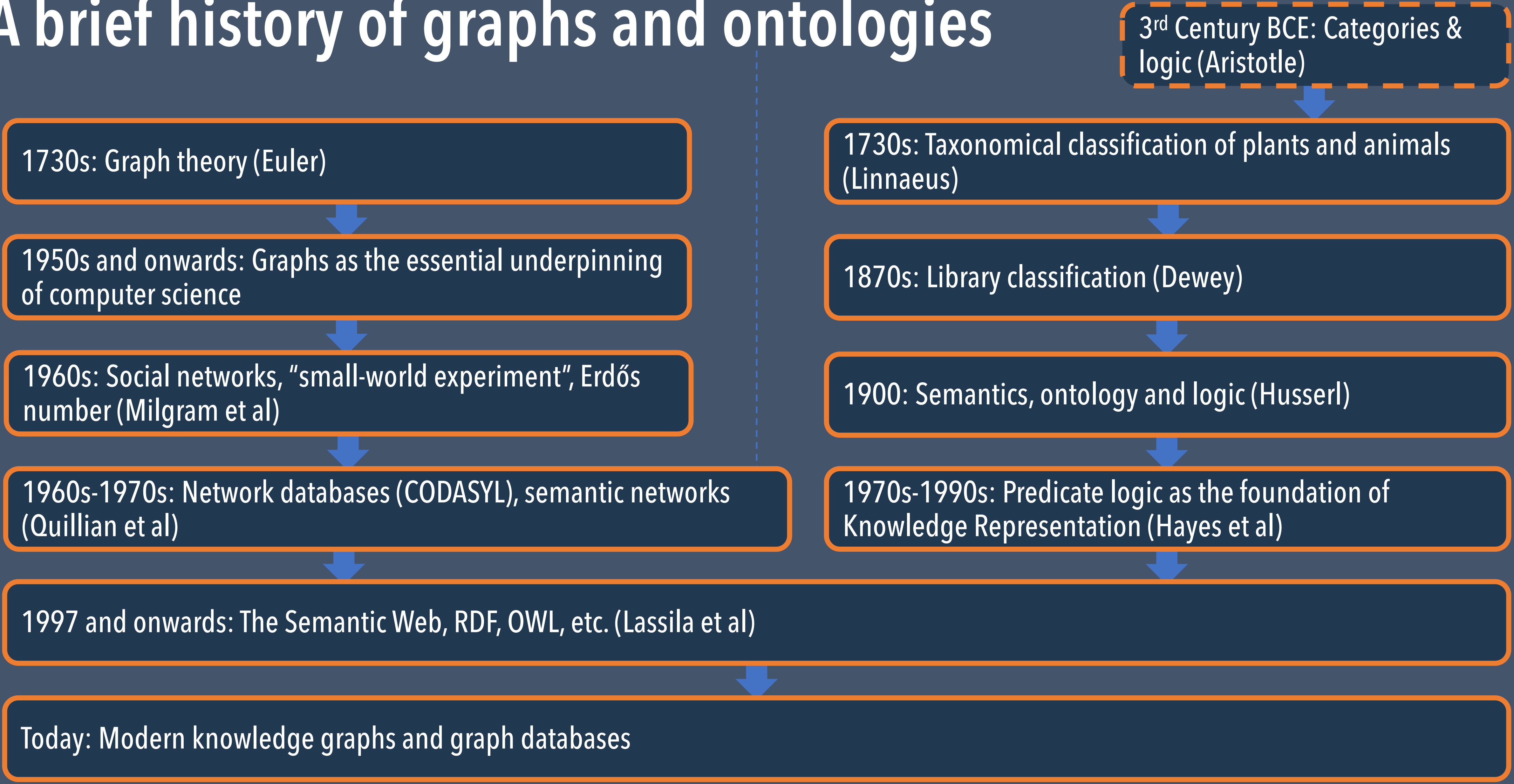
Q: Why then is the Semantic Web so attractive?
A: For future-proofing

Semantic Web can be a solution to those problems and situations that we are yet to define

(yep, seriously)

Part 2: History

A brief history of graphs and ontologies



3rd Century BCE: Categories & logic (Aristotle)

1730s: Graph theory (Euler)

1950s and onwards: Graphs as the essential underpinning of computer science

1960s: Social networks, “small-world experiment”, Erdős number (Milgram et al)

1960s-1970s: Network databases (CODASYL), semantic networks (Quillian et al)

1730s: Taxonomical classification of plants and animals (Linnaeus)

1870s: Library classification (Dewey)

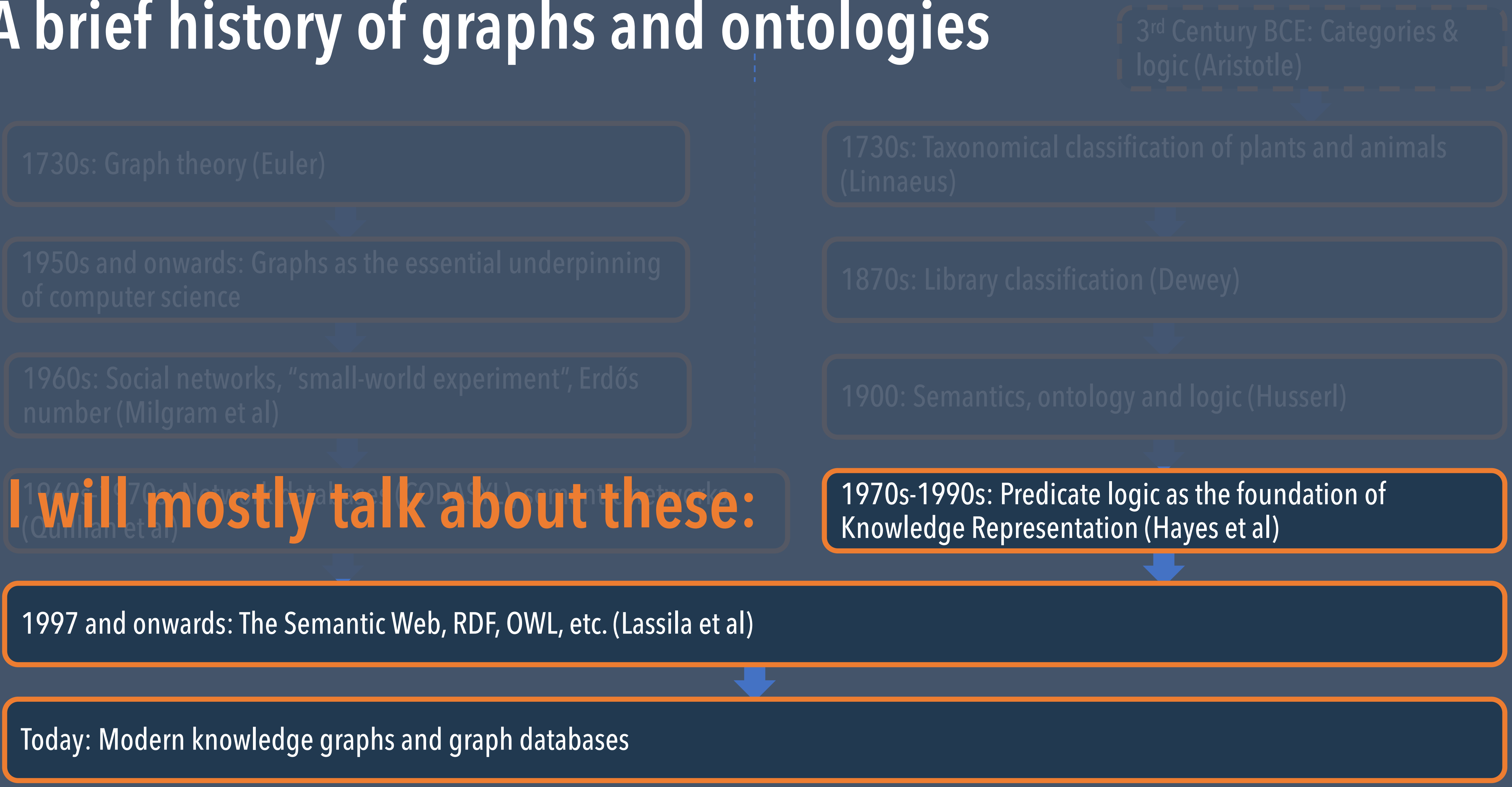
1900: Semantics, ontology and logic (Husserl)

1970s-1990s: Predicate logic as the foundation of Knowledge Representation (Hayes et al)

1997 and onwards: The Semantic Web, RDF, OWL, etc. (Lassila et al)

Today: Modern knowledge graphs and graph databases

A brief history of graphs and ontologies



3rd Century BCE: Categories & logic (Aristotle)

1730s: Graph theory (Euler)

1950s and onwards: Graphs as the essential underpinning of computer science

1960s: Social networks, "small-world experiment", Erdős number (Milgram et al)

1960s-1970s: Network databases (CODASYL), semantic networks (Quillian et al)

1730s: Taxonomical classification of plants and animals (Linnaeus)

1870s: Library classification (Dewey)

1900: Semantics, ontology and logic (Husserl)

1970s-1990s: Predicate logic as the foundation of Knowledge Representation (Hayes et al)

I will mostly talk about these:

1997 and onwards: The Semantic Web, RDF, OWL, etc. (Lassila et al)

Today: Modern knowledge graphs and graph databases

Pre-history

Knowledge Representation (KR) is a long-established subfield of AI

- I got introduced to KR as a grad student in the late 1980s
- I worked on several KR systems (some of my own design) and built ontologies
- a KR system that I designed and built flew on NASA's Deep Space 1 probe as part of planning software to support the autonomous behavior of the spacecraft

"Symbol grounding" is the fundamental problem of AI

- KR tackles this

When I was working at MIT, Tim Berners-Lee asked me a "fateful" question…

One day in 1996 @ MIT

Per my recollection, this is what happened…

TimBL: *"So Ora, what do you think is wrong with the Web?"*

me: *"Well Tim, the Web was built for humans, and that makes it hard to automate anything. I want autonomous agents."*

TimBL: *"That's it! How do we solve this?"*

me: *"Ugh… I don't know. Maybe we should see if knowledge representation could be used."*

TimBL: "Please look into this."

Pre-history

Tim’s questions to me came at a very opportune moment

- I was involved in researching “ubiquitous computing”, an idea that computation (use of personal computing) can be “pushed to the periphery of the user’s attention” and can become part of the physical environments where we live and work [Weiser 1991]
- my hypothesis was that we needed knowledge representation (to model various devices, their interconnections, and their capabilities) and intelligent agents (to act as the layer between users and the computing environment)
- consequence: this is not just “about the Web”

My subsequent experimentation led to the formation of the first RDF WG

Early days at W3C

Work on RDF was difficult and contentious

- this was in the middle of the so-called “browser wars”, and both Netscape and Microsoft were participating in the working group

We ended up with two groups: “RDF Model and Syntax” and “RDF Schema”

- there was no technical reason for this, it was just timing and organization
- this is the reason why we now have the “rdf:” and “rdfs:” namespaces
- this was a mistake and has lead to a lot of confusion

Political pressures led to the adoption of an XML-based syntax

- this was a mistake and has lead to a lot of confusion

“Why can’t I just use XML?”

I think this seemingly simple question delayed the adoption of Semantic Web technologies by several years

Many people are very focused on syntax

- but, syntax does not matter
- metamodels and semantics are much more important

I once sat through a presentation where the authors added all kinds of stuff to XML to show that “XML is enough”

- all this effectively amounted to a poorly constructed alternative to RDF

“Why can’t I just use XML?”

Unfortunately, we learned nothing

- flash forward 20 years: “Why can’t I just use JSON?”
- (well, um, for the same reason: syntax does not matter)
- and besides, it is never “just JSON”

RDF ideas and "alternatives"

There were several competing ideas and proposals:

- PICS-NG – my early experiments at MIT, formed the initial basis of RDF
- Meta-Content Framework (MCF) – Netscape's proposal, strongly influenced RDF
- Topic Maps – eventually standardized by ISO, but has now largely disappeared
- XML-Data – proposed by Microsoft, essentially re-invented RDF in a wholly XML-centric way, but missed the entire semantics part

People suggesting an alternative approach had usually misunderstood RDF…

RDF and RDF Schema

First W3C working group formed in 1997

- October 1997: First public draft
- February 1999: RDF becomes a standard (a "W3C Recommendation")

"RDF Schema" becomes a Candidate Recommendation in March 2000

- it takes until February 2004 for it to become an official Recommendation

We did not have enough usage experience, but we needed a basic standard

- Semantic Web is a "team sport", and researchers needed common rules of the game

RDF timeline

May 1997	First version of the PICS-NG proposal
October 1997	First public “working draft” of something called RDF
February 1999	RDF “Model and Syntax”
February 2004	RDF “Model and Syntax” revised (“RDF 1.0”)
January 2008	SPARQL 1.0
March 2013	SPARQL 1.1
February 2014	RDF 1.1
December 2021	RDF-star CG final report
August 2022	RDF-star WG formed

Web Ontology Language OWL

DARPA launched their Semantic Web research program (“DAML”) in 1999

The KR community viewed RDF as not expressive enough, and wanted to specify a more powerful ontology language

DARPA effort merged with similar European effort

A series of draft ontology languages emerged: DAML-ONT, DAML+OIL, OWL

Many DAML researchers also helped refine RDF into version 1.1

Proliferation of Semantic Web specifications

Languages, vocabularies, ontologies

GRDDL – extract RDF from Web pages

SHACL – validate RDF data

R2RML – map relational data to RDF

RIF – rule interchange

OWL-S – “Semantic Web Services”

Dublin Core – basic metadata

FOAF – social networks

SKOS – thesauri and classification

PROV-O – provenance and lineage

DCAT – data catalogs

Semantic Web Services?

Basic idea was to model the semantics of Web Services using ontologies

Started as DAML-S, this later became OWL-S

The problem is much more difficult than one would imagine

- describing behavior, not just data
- solution to the general problem has been attempted before DAML-S (process calculus; Milner, Hoare, et al), and people continue to work on it
- comes with all the "fun" of programming language semantics, etc.

Linked Data?

Early on, we* insisted that the Semantic Web is ultimately about AI

- and here "AI" is the classical, symbolic AI (rules, KR, etc.)

At the time, AI was not popular at all

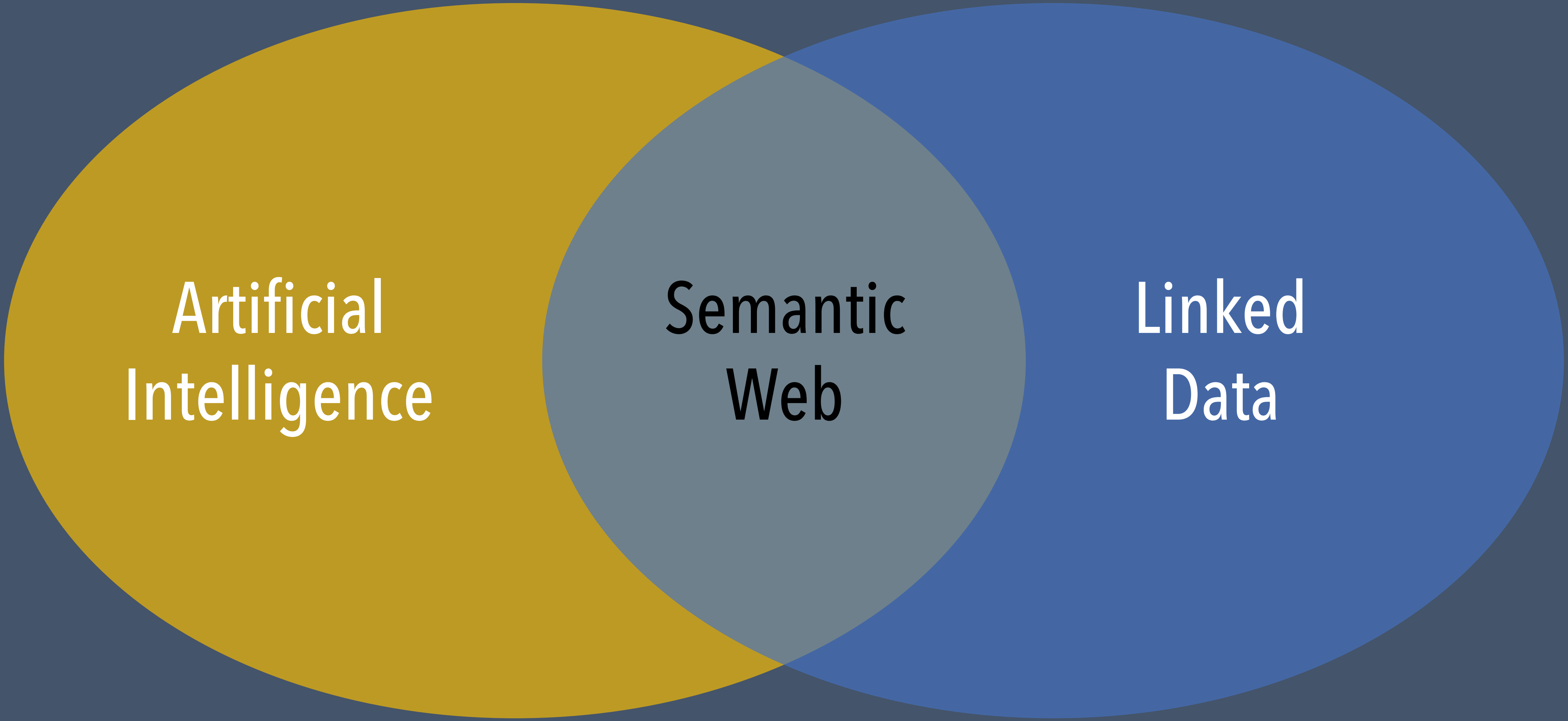
- (we were in the middle of the so-called "AI Winter")

Linked Data is essentially "the Semantic Web with AI removed"

- I view Linked Data as a useful endeavor that can supply the Semantic Web with data, but it falls short of the original vision
- there has been a lot of work to "put AI back in the Semantic Web"

** Jim Hendler and me*

I told you this was about AI, but nobody listened…



Part 3: State of the Union

Short summary

We are winning!

Let me explain…

Longer summary

Many Semantic Web technologies have moved to the mainstream

Gartner says ontologies, knowledge graphs, RDF are a “major trend”

Modern “enterprise knowledge graphs” are the mature transformation of the original vision

- reducing the scope from the whole Web to an enterprise makes several tough problems more manageable

More good news

Knowledge graphs are increasingly seen as a good means to do data integration

Challenge: Everybody wants a “semantic layer”, but nobody really knows what it is

- (I have an idea)
- this has partly to do with the fact that nobody knows what “semantics” is

Also: RDF standardization continues, we are working on RDF 1.2 (aka “RDF-star”)

What about Labeled Property Graphs?

A graph is a graph is a graph, or is it? (short answer: it is not)

RDF has its underpinnings in knowledge representation, ontologies, symbolic AI

LPGs come from the programming and database worlds

RDF-based knowledge graphs can be (and should be) built on the assumption that there is more knowledge "out there" (= "open-world assumption") and that we don't necessarily know all the use cases and applications we will build on top

LPG-based systems are typically closed and built for a single purpose (much like we have traditionally used databases)

Some observations about "closed world vs. open world"

Closed world

If a database does not contain X, we must conclude that X does not exist (aka "negation by failure")

We can choose any identifiers for our objects in our database, since the outside world "does not matter"

Open world

No such assumption can be made; X could still be out there, we just haven't seen it yet

We need strong, globally unique identifiers for our objects, since effectively the world continues outside our database

Labeled Property Graphs vs. RDF

RDF	LPGs
Highly standardized • one query language • schema language(s)	Standardization only now beginning • many query languages • no schema language
Nodes are merely identifiers, edges have no structure	Both nodes and edges are structured objects
Graph is a logical model	Graph is a data structure
Formal semantics	No semantics (or “ad hoc”)
Cumbersome to program with	Better suited for programming

Part 4: Future

Questions

Can we now fully realize the Semantic Web vision?

What is still missing?

Does generative AI play a role here?

What is the relationship between LLM and KGs?

How do we enable the adoption of these technologies and principles in a way that can happen in an incremental fashion?

Will we see alignment between RDF and LPGs?

Questions

Can we now fully realize the Semantic Web vision?

What is still missing?

Does generative AI play a role here?

What is the relationship between LLM and KGs?

These are all related

How do we enable the adoption of these technologies and principles in a way that can happen in an incremental fashion?

Will we see alignment between RDF and LPGs?

What is missing? Intelligent agents!

The “full” Semantic Web vision is predicated on the idea that we can converse with our agents and give them tasks to perform

Using LLMs, sufficiently flexible and open-ended conversational user interfaces are finally possible

Through curated and audited knowledge graphs, we can have trusted sources of information for the agents to consume (and avoid LLM hallucinations)

Full realization of intelligent agents requires reasoning and planning capabilities

- LLMs do not have these

Knowledge graphs and LLMs

Amidst all the hype around generative AI, there are beliefs that we can

- use LLMs to construct ontologies and knowledge graphs
- eliminate hallucinations using knowledge graphs to improve RAG techniques

(I do not share these beliefs)

Rather than using KGs to improve LLM-based systems, we should use LLMs to improve KG-based systems

- answers should come from trusted, curated sources (= KGs), not from LLMs
- simplistically: let LLMs write queries and translate query results, but not actually answer questions or generate answers

Questions

Can we now fully realize the Semantic Web vision?

What is still missing?

Does generative AI play a role here?

What is the relationship between LLM and KGs?

How do we enable the adoption of these technologies and principles in a way that can happen in an incremental fashion?

Will we see alignment between RDF and LPGs?

Thank you! Any questions?

Contact: ora@amazon.com