Semantic Technologies Seminar: Using the W3C Standard OWL for Semantic Interoperability

Updated: APR-2004

Today's Agenda

- Core Semantic Interoperability Use Cases:

  - **Semantic Web Services**

  - **Business Inferencing**

  - **Semantic Data Integration**

- What is OWL?

- Top 5 Reasons Why OWL Matters:

  - **5 – OWL semantics are model-driven**

  - **4 – OWL semantics are machine-actionable**

  - **3 – OWL semantics are more expressive**

  - **2 – OWL semantics are more precise**

  - **1 – …drum roll please!**

Use Case: Semantic Web Services

- **Why is "meaning" important in web services, SOA, and grid computing?**

  - Avoid transformation code between data sets

  - Unambiguously capture service profiles

  - Enable dynamic discovery of services

    - Use reasoners to locate services in "yellow pages"

  - Enable dynamic collaboration of services

    - Use reasoners to infer service descriptions and capabilities

  - Enable rich, automatic, service orchestration

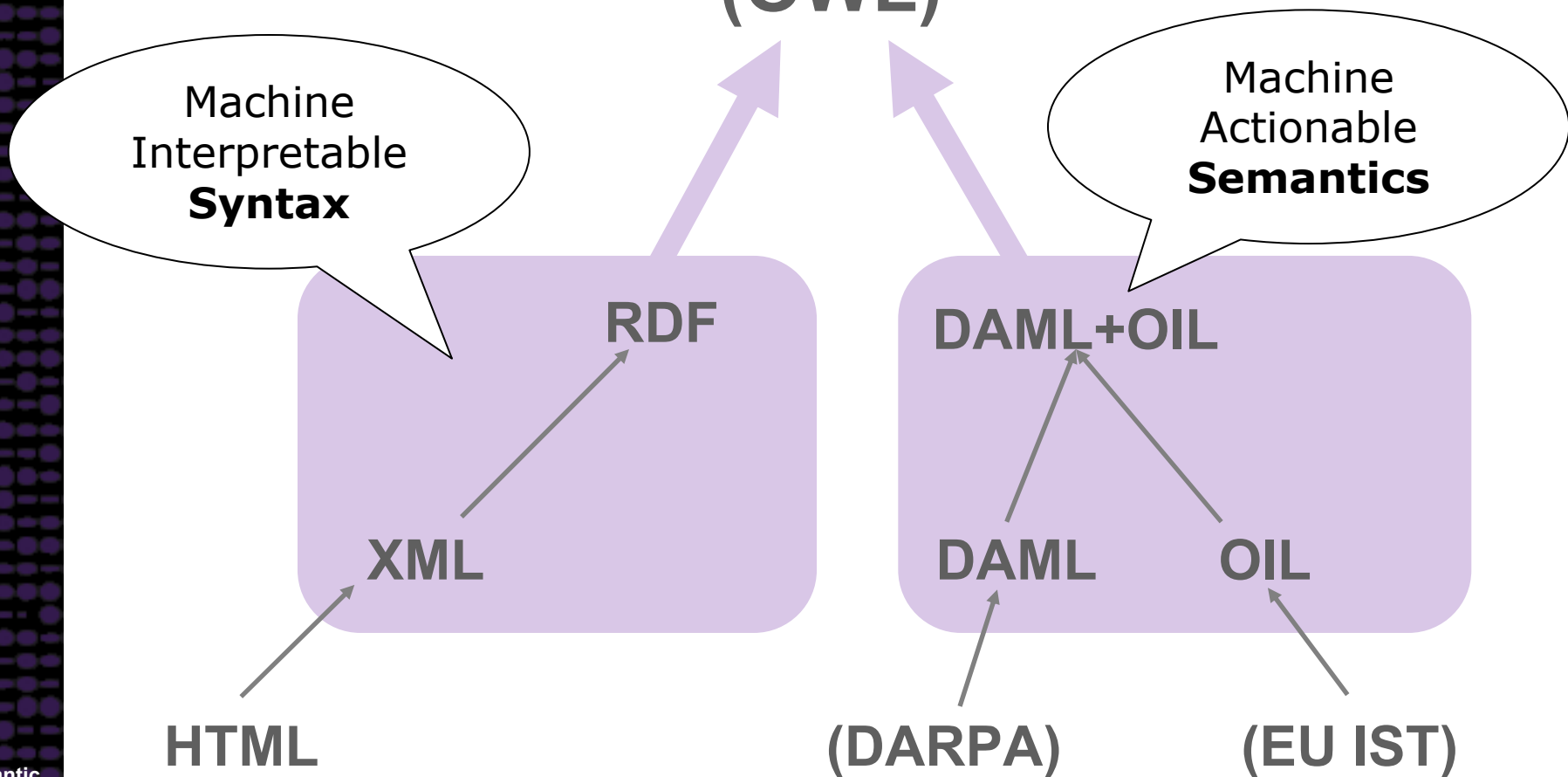   - Process layer will be **far more automated** with semantics

Use Case: Business Inferencing

- ## What is it, and why should I care?

  - Business Inferencing is machine visibility into operational data, semantics, and business rules

    - Previously, any comparable capabilities were via highly proprietary metadata markup embedded inside tools

  - Business Inferencing enables an application layer to infer business rules – both implicit and explicit

    - Thus enabling machine access to implicit business facts – as they have been modeled, or implied inside business ontologies – allowing automated use of implicit data/rules

    - It is used as a platform for application development

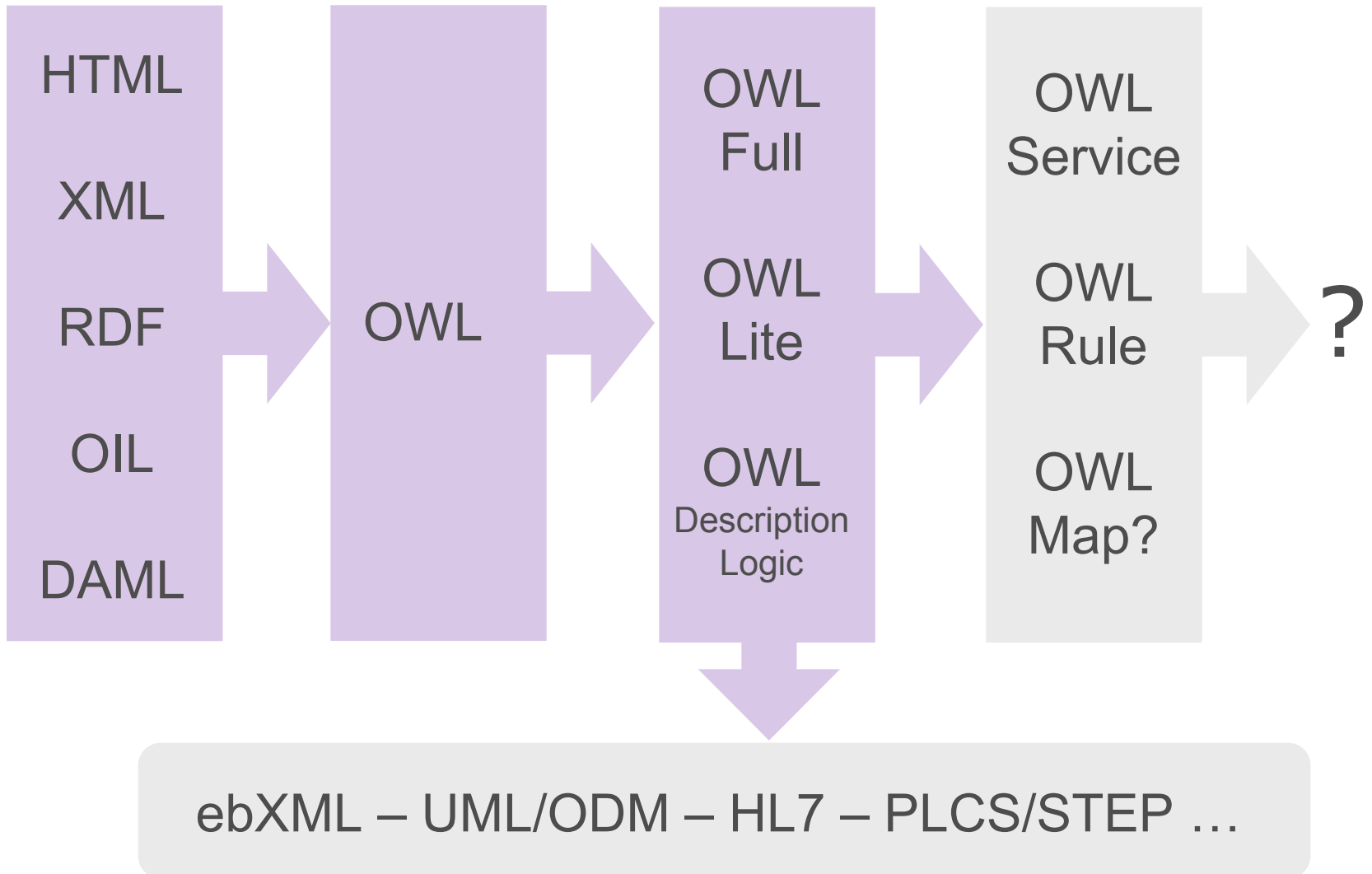   - Replaces the business rules tier and manages business vocabularies at the infrastructure level – **saves $$$**

⊙ **What's different, why can't the established vendors simply add-in these capabilities?**

- Semantic Data Integration is the use of ontology as a mediating vocabulary for disparate underlying sources – a virtual hub and spoke

  - Unlike previous "business object" or "bus" style approaches, ontologies are conceptual languages at a higher abstraction – they don't have to map 1:1 with underlying systems

- Most vendors are committed to their data architectures, OWL is best used in the "core" – not as an "add-on" to an existing COTS product.

  - Full automation will not come "for free" with simple plug-ins, however, **improvements are always achievable**

What is OWL's Roadmap?

HTML
XML
RDF
OIL
DAML

→

OWL

→

OWL Full
OWL Lite
OWL Description Logic

→

OWL Service
OWL Rule
OWL Map?

→

?

ebXML – UML/ODM – HL7 – PLCS/STEP …

Semantic Technology Seminar

- **Semantics are loosely-coupled**
  - Characteristic
    - OWL ontologies are schema representations, independent of application code and RDF models
    - OWL models are easily stored and referenced in a loosely-coupled registry/repository style architecture
  - Benefits
    - Semantics are late-bound, thereby supporting an evolutionary – not static – network model for changing data meanings and business rules
    - Semantics may be easily federated
    - Semantics may be loosely-coupled to instance data

⦿ **Semantics are machine-actionable**

- Characteristic
    - OWL is syntax (not graphical) grounded in XML & RDF
    - OWL uses consistent, standard schema semantics
    - Supports well-scoped classes, properties (class relationships), instances and instance relationships

- Benefits
    - Parsers, modelers, reasoners, and transformers are available today
    - DL guarantees 100% decidability and computational completeness – any two DL reasoners should come up with the same (all possible) answers to queries

Why OWL Matters – Reason #3

- ◉ **OWL is more expressive**
  - Characteristic
    - Rich set of built-in simple properties, property characteristics and restrictions
    - Not just hierarchical or taxonomic (like most XML)
    - Not just two-dimensional (like ER/RDBMS)
    - Allowable, functional, multiple inheritance
  - Benefit
    - More closely models "real-world"
    - Axioms may be used to model rules directly into the model (compare with OCL-type approaches)

Why OWL Matters – Reason #2

- ◉ **OWL is more precise**

  - Characteristic

    - Relationships are atomic and unambiguous

      - Unlike UML/ER/XML, properties have stand-alone meaning

    - Disallows over-riding attributes (no semantic ambiguity)

    - DL enforces consistency

    - Within a context, semantics can be 100% unambiguous

  - Benefit

    - Reasoners can accommodate uncertain/unknown data

    - Both explicit and **implicit** facts are available via a mediated query capability

- **OWL is a "cute" acronym (yet appropriate)**
  - Characteristic
    - OWL = wisdom
    - OWL is named for the owl in Winnie the Pooh (who spelled his own name "WOL")
  - Benefit
    - Makes people smile and laugh!

Jeff Pollock
Vice President, Technology

jeff.pollock@networkinference.com