Semantic Web

Reasoning on the Web
## Where are we?

<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Semantic Web Architecture</td>
</tr>
<tr>
<td>3</td>
<td>Resource Description Framework (RDF)</td>
</tr>
<tr>
<td>4</td>
<td>Web of data</td>
</tr>
<tr>
<td>5</td>
<td>Generating Semantic Annotations</td>
</tr>
<tr>
<td>6</td>
<td>Storage and Querying</td>
</tr>
<tr>
<td>7</td>
<td>Web Ontology Language (OWL)</td>
</tr>
<tr>
<td>8</td>
<td>Rule Interchange Format (RIF)</td>
</tr>
<tr>
<td>9</td>
<td><strong>Reasoning on the Web</strong></td>
</tr>
<tr>
<td>10</td>
<td>Ontologies</td>
</tr>
<tr>
<td>11</td>
<td>Social Semantic Web</td>
</tr>
<tr>
<td>12</td>
<td>Semantic Web Services</td>
</tr>
<tr>
<td>13</td>
<td>Tools</td>
</tr>
<tr>
<td>14</td>
<td>Applications</td>
</tr>
</tbody>
</table>
1. Introduction and Motivation
2. Technical Solution
   1. Approximate Reasoning
   2. Bounded Reasoning
   3. MaRVIN
   4. LarKC
3. Illustration by a Large Example
4. Summary
5. References
Is traditional reasoning compatible with the Web?

MOTIVATION
What is Reasoning?

- **Reasoning** is the cognitive process of looking for reasons for beliefs, conclusions, actions or feelings [http://en.wikipedia.org/wiki/Reasoning]

- The world does not give us complete information
- Reasoning is the set of processes that enables us to go beyond the information given
  - Human are Mortals + Socrate is a Human => Socrate is Mortal

- Reasoning in most of the cases is based on Logic
What is Logic?

- **Logic** is the study of the principles of valid **demonstration** and inference
- Logic concerns the **structure of statements** and arguments, in **formal systems** of inference and natural language
First Order Logic in Short

• “Classical” logic
  – Based on propositional logic (Aristotle, 300 BC)
  – Developed in 19th century (Frege, 1879)

• Semi-decidable logic
  – Enumerate all true sentences
  – If a sentence is false, the algorithm might not terminate

• FOL is the basis for
  – Logic Programming: Horn Logic
  – Description Logics: 2-variable fragment

• A logic for describing object, functions and relations
  – Objects are “things” in the world: persons, cars, etc.
  – Functions take a number of objects as argument and “return” an object, depending on the arguments: addition, father-of, etc.
  – Relations hold between objects: distance, marriage, etc.
  – Often, a function can also be modeled as a relation
• Propositional logic deals only with truth-functional validity: any assignment of truth-values to the variables of the argument should make either the conclusion true or at least one of the premises false.
  – All men are mortal
  – Socrates is a man
  – Therefore, Socrates is mortal

• which upon translation into propositional logic yields:
  – A
  – B
  – Therefore C
First Order Logic in Short

- According to propositional logic, this translation is invalid.

- Propositional logic validates arguments according to their structure, and nothing in the structure of this translated argument (C follows from A and B, for arbitrary A, B, C) suggests that it is valid.

- First Order Logic satisfies such needs!

- Is this enough to apply such logic on the Web? Unfortunately no... FOL is not decidable and not efficient.

- Neither families derived from it are ready to scale a the Web size (HL and DL).
Requirements for Web-scale Reasoning

1. On the Web axioms are an indefinite growing large number
   – Web represent human knowledge, then the number of axioms is to grow constantly

2. On the Web facts are an indefinite growing large number
   – The are 30 billion pages indexed on Google, even if each page contains only 10 fact, we are behind the size of facts that can be handled by DL and HL

3. The Web is open and with no defined boundaries, completeness is not achievable
   – On the Web you don’t check all the answers, picking the first 10 is usually enough and safe lot of time

4. The Web is not consistent in its nature
   – Web is full of contradiction since enable different people to express their different view point, applying traditional logic will bring to find contradictions

5. The Web is a dynamic entity
   – Data are constantly updated, hence applying traditional reasoning will lead or to incomplete knowledge or to outdated knowledge

New paradigms for reasoning at Web-scale are needed!
Reasoning at Web-scale

TECHNICAL SOLUTION
Data Look-up on the Web

• In a large, distributed, and heterogeneous environment, classical ACID guarantees of the database world no longer scale in any sense.

• Even a simple read operation in an environment such as the Web, a peer-to-peer storage network, a set of distributed repositories, or a space, cannot guarantee completeness in the sense of assuming that if data was not returned, then it was not there.

• Similarly, a write can also not guarantee a consistent state that it is immediately replicated to all the storage facilities at once.
Limits of Traditional Reasoning Approaches

1. Small set of axioms
   - DL can deal with large set of axioms around $10^5$, but has trouble with facts

2. Small number of facts
   - HL can deal with large set of axioms around $10^5$ and around $10^6$ facts, but enables only simple logic conclusions

3. Completeness of inferences rules
   - DL and HL terminate the inference procedure when there is nothing more to be inferred.

4. Trustworthiness correctness of inference rules and consistency
   - In DL and HL we reason in term of truth. Axioms represent the truth, thus inference require a set of consistent axioms to lead to a consistent theory.

5. Static domain
   - In DL and HL, the assumption is that the knowledge is not evolving during the reasoning process.

Are these assumptions compatible with Web-scale reasoning?
Modern information retrieval applies the same principles

- In information retrieval, the notion of completeness (recall) becomes more and more meaningless in the context of Web scale information infrastructures.
- It is very unlikely that a user requests all the information relevant to a certain topic that exists on a worldwide scale, since this could easily go far beyond the amount of information processing he or she is investing in achieving a certain goal.
- Therefore, instead of investigating the full space of precision and recall, information retrieval is starting to focus more around improving precision and proper ranking of results.
Reasoning on the Web

• What holds for a simple data look-up holds in an even stronger sense for reasoning on Web scale.

• The notion of 100% completeness and correctness as usually assumed in logic-based reasoning does not even make sense anymore since the underlying fact base is changing faster than any reasoning process can process it.

• Therefore, we have to develop a notion of usability of inferred results and relate them with the resources that are requested for it.
Reasoning on the Web

Semantic Web

Logic

precision (soundness)

recall (completeness)

IR

[D. Fensel, Computer Science in 21st Century]
Soundness and Completeness

• Soundness:
  – All the reported solution are correct

• Completeness:
  – All the solution are found or if not solution is possible the system reports that no solution is possible
Overview of Web-scale Reasoning Approaches

- **Approximate Reasoning**
  - Developed in logics and artificial intelligence to deal with the scalability problem

- **Resource Bounded Reasoning**
  - Reasoning algorithms apply heuristics to solve problems according to the real available resources

- **Rule-based Reasoning for dynamic and incomplete knowledge**
  - Humans usually apply rule-based reasoning in the context of incomplete knowledge to take a decision
APPROXIMATE REASONING
Why Approximate Reasoning

• Current inference is exact:
  – “yes” or “no”

• This was OK, because until now ontologies were clean:
  – Hand-crafted, well-designed, carefully populated, well maintained,…

• BUT, ontologies will be sloppy:
  – Made by machines
  – (e.g. almost subClassOf)
  – Mapping ontologies is almost always messy
  – (e.g. almost equal)
Example the MadCow Ontology

- Cow ⊆ Vegetarian
- MadCow ⊆ Cow
- MadCow ⊆ ∃ Eat.BrainofSheep
- Sheep ⊆ Animal
- Vegetarian ⊆ ∀ Eat. ¬ (Animal ∨ PartofAnimal)
- Brain ⊆ PartofAnimal
- ....
- theMadCow ∈ MadCow
- ...

[Z. Huang et al., Reasoning with Inconsistent Ontologies]
Approximate Entailment [Schaerf & Cadoli, 1995]

• Two approximate entailment operators
  – $\vdash_1$: Complete but unsound
  – $\vdash_3$: Sound but incomplete
  – $\vdash_1$ and $\vdash_3$ are approximation of the classical consequence

• $\vdash_1$ and $\vdash_3$ are parameterized over a set of predicate letters $S$
  – $\vdash_1^S$ and $\vdash_3^S$

• $S$ determines the accuracy of the approximate entailment relations

• The more $S$ increase the more the approximation get closer to the classical entailment
Approximate Entailment [Schaerf & Cadoli, 1995]

- \( \models_{1}^{S} \): interpret everything outside of \( S \) as \textit{false}

- \( \models_{3}^{S} \): interpret everything outside of \( S \) as \textit{true} (or normal)
Effect of $\vdash_1^S$ [Schaerf & Cadoli, 1995]

\[ V = \{p, q\} \]
\[ S = \{p\} \]

\[ (p \lor q[false]) \land \neg p \]

Result: \( p \lor q \Rightarrow p \) ! Incorrect, but complete reasoning
Effect of $\vdash_3^S$ [Schaerf & Cadoli, 1995]

$V = \{p,q\}$
$S = \{p\}$

$(p \lor q[true]) \land (\neg p[true]) \land \neg q)$

Result: $(p \lor q) \not\Rightarrow \neg (\neg p \land \neg q)!$ Correct, but incomplete reasoning
Approximate Entailment [Schaerf & Cadoli, 1995]

- Anytime behavior when $S_i$ is increased
- Previous steps can be reused
- The approximation decrease when the $S$ increase

- Approximate reasoning enables for anytime behavior. Since we admit unsound and incomplete answers, we can stop the process in several points according to resource or time constraints.
• Query over knowledge base is relaxed so that it can be computed in shorter time
• The original query is decomposed in a sequence of queries that are approximations for the original query
  – $Q_1, \ldots, Q_n$
• The assumption is that the quality of the results of the sequence of queries is non-decreasing
Example [Stuckenschmidt 2006]

- The set of axioms
  - Mother ≡ Woman ∧ Parent
  - Woman ≡ Person ∧ ∃ hasGender.Female
  - Parent ≡ Person ∧ ∃ hasChild.Person
  - Grandmother ≡ Woman ∧ ∃ hasChild.(∃ hasChild.Person)

- Can be relaxed by replacing subexpressions that directly contain the slot has-gender by T
  - Mother ≡ T ∧ Parent
  - Woman ≡ Person ∧ T
  - Parent ≡ Person ∧ ∃ hasChild.Person
  - Grandmother ≡ Woman ∧ ∃ hasChild.(∃ hasChild.Person)
Using Approximations for Query Answering
[Selman and Kautz 1991]

- $S \models q$?
- If $\text{LUB} \models q$ then $S \models q$
  - (linear time)
- If $\text{GLB} \models q$ then $S \models q$
  - (linear time)
- Otherwise, use $S$ directly
  - (or return "don't know")

- Queries answered in linear time lead to improvement in overall response time to a series of queries
Example [Selman and Kautz 1991]

- Consider the theory $S$:
  - $\neg a \lor c \land \neg b \lor c \land a \lor b$

- GLBs are
  - $(a \land c)$
  - $(b \land c)$

- LUB
  - $c$

- $S \not\models \neg c$?
  - Clearly no
BOUNDDED REASONING
Bounded Rationality

• Nobel prize Herbert A. Simon revolutionized economic theories by introducing the concept of “bounded rationality” to explain human behavior

• *Bounded rationality is the concept that the rationality of individuals is limited by the information they have, the cognitive limitations of their minds, and the finite amount of time they have to make decisions*

• Heuristics can be applied to take in consideration availability of resources
Web Reasoning and Bounded Rationality

• On the Web:
  – It’s not possible to collect the complete information about a fact
  – It’s not possible to collect the complete list of axioms that model the Web
  – Resources to answer a query are limited in term of computational power and in term of time

• Bounded Rationality is a valid assumption also on the Web

• We can define heuristic to support Web-scale reasoning
  – If it takes more than 1 second to give me all the answers, then give me only the ones you find in 1 second
  – If the knowledge base is to large to be used entirely, consider only a portion of it
Application for Bounded Reasoning: Streaming Reasoning

• Data streams are unbounded sequences of time-varying data elements
  – Typical in: network monitoring, traffic engineering, sensor networks, RFID tags applications, telecom call records, financial applications, Web logs, click-streams, etc.

• While reasoners are year after year scaling up in the classical, time invariant domain of ontological knowledge, reasoning upon rapidly changing information has been neglected or forgotten

• Requirements
  – Fast processing time (Time bound)
  – Capability to deal with constantly evolving knowledge (Knowledge bound)
Stream DB + Semantics = Stream Reasoning
[Della Valle, Stream Reasoning for Urban Computing]

- RDF streams: new data formats set at the confluence of conventional data streams and of conventional atoms usually injected into reasoners
- Continuous SPARQL (C-SPARQL)
  The distinguishing feature of C-SPARQL is the support for continuous queries, i.e. SPARQL-like queries registered over RDF data streams in the context of a C-SPARQL execution environment and then continuously executed

![Problem Modeling Framework Diagram]
A Scalable Distributed Reasoner over RDF

MARVIN
MaRVIN (Massive RDF Versatile Inference Network)

• MaRVIN is:
  – a platform for distributed RDF(S) reasoning
  – a platform for processing lots of RDF data

• MaRVIN scales by:
  – distributing computation over many nodes
  – approximate (sound but incomplete) reasoning
  – anytime convergence (more complete over time)

• MaRVIN runs on:
  – in principle: any grid, using Ibis middleware
  – currently: the DAS-3 distributed supercomputer (300 nodes)
  – soon: a wide-area a peer-to-peer network
MaRVIN Architecture
MaRVIN Distribution Algorithm

- Main loop: divide-conquer-and-swap
  
  - 1. divide: split input data in chunks
  
  - 2. conquer: each node:
    - reads some chunks,
    - computes closure.
  
  - 3. swap: each node:
    - removes all triples:
      - sends some to central storage,
      - sends other to some peer
  
- repeat 2-3 ad infinitum
MaRVIN Random Data Exchange

- **Peers exchange data randomly**
  - they randomly pick some other peer, and give him some random part of their own data

- **Advantages**
  - Random exchanges maintain optimal load-balance: all peers have approximately the same amount of data. This means that no peer is overloaded and no peer is underutilised

- **Disadvantages**
  - Random exchanges are not very efficient for our task. We want to reason with triples, which means that triples with a shared key should meet at one peer to derive a consequence. With random exchanges, the chances of triples meeting is quite low
MaRVIN DHT Data Exchanges

- Peers exchange data similarly to a DHT
  - based on a hash of the data, they pick the peer responsible for this key, and give him some all pieces of their data that "belong" to him

- Advantages
  - Such a DHT-like exchange are very efficient for our task. We want to reason with triples, which means that triples with a shared key should meet at one peer to derive a consequence. With these targeted exchanges, since all triples that share a key are sent to the same peer, their chance of meeting is maximal. Within a limited amount of exchanges (depending on the bandwith they have for sending and receiving data) all data items will be at the peer "responsible" for them, and will have met their "buddy" triples to produce a consequence

- Disadvantages
  - This approach ignores load balancing. Since keys in triples are very unevenly distributed (some terms are much more popular than others), some peers will be "responsible" for much more triples than others. This means that some peers will be overloaded and others will be underutilised.
MaRVIN SpeedDate Data Exchanges

• Peers exchange data according to a **speed-dating** strategy
  – A hybrid technique between the targeted model of a DHT and the uniform model of the random exchange. Peers try to specialise and ask for only "their" data items, but also offer help to overloaded peers around them. The result is a clustering of data around the peer responsible, with close-by peers helping out.

• Advantages
  – With this approach, the peers maintain optimal load balance, as in the random distribution. On the other hand, data items meet almost as much as in the directed approach of the DHT. Thus, for our particular task, it seems that this hybrid approach gives us the best of both worlds.

• Disadvantages of this strategy
  – We don't know yet how well the strategy scales with more nodes, more keys, and more data: we don't know yet how it behaves in different circumstances. We are continuing our research to find answers to these questions.
A platform for Large Scale Web Reasoning

LARKC
What is LarKC?

Large Knowledge Collider
Simplified Overview

Identify
- Relevant Sources
- Relevant Content
- Relevant Context

Transform
- Extract Information
- Calculate Statistics
- Transform to Logic

Select
- Relevant Problems
- Relevant Methods
- Relevant Data

Reason
- Probabilistic Inference
- Classification
- Context reasoning

Decide
- Enough answers?
- Enough certainty?
- Enough effort/cost?
LarKC Architecture

Application

Pipeline Support System

Plug-in Registry

Plug-in Manager
- Plug-in API
- Query Transformer

Plug-in Manager
- Plug-in API
- Identifier

Plug-in Manager
- Plug-in API
- Info. Set Transformer

Plug-in Manager
- Plug-in API
- Selecter

Plug-in Manager
- Plug-in API
- Reasoner

Data Layer API

Data Layer

RDF Store

RDF Store

RDF Store

RDF Doc

RDF Doc

RDF Doc
LarKC Plug-in Types

- **Identify**
  - Responsible for finding resources to be used in a given pipeline
  - Information sets are for example:
    - RDF documents on the Web
    - Named graphs in a triple store
    - Natural language documents

- **Examples**
  - Sindice – Triple Pattern Query → RDF Graphs
  - Google – Keyword Query → Natural Language Document
  - Triple Store – SPARQL Query → RDF Graphs
LarKC Plug-in Types

- **Transform**
  - Given some data or a query, transform it to a different representation

```
Query ➔ Transformer ➔ Collection<Query>
```

```
InformationSet ➔ Transformer ➔ InformationSet
```

- **Examples**
  - SPARQL Query ➔ Keyword Query
  - Natural Language Document ➔ RDF Graph
  - RDF Graph ➔ RDF Graph (foaf ontology to facebook ontology)
LarKC Plug-in Types

- **Select**
  - Responsible for selecting a subset of a specified set of statements

```
SetOfStatements → Selector → SetOfStatements
```

- **Examples**
  - Collection of RDF Graphs → Data Set (Merged)
  - Collection of RDF Graphs → Labeled Set (Subset)
LarKC Plug-in Types

- **Reason**
  - Responsible for performing reasoning on a given set of statements
- **The output depends on the kind of query**

  ![Diagram](image)

  - SPARQL query $\rightarrow$ Reasoner
    - $\rightarrow$ VariableBinding
    - $\rightarrow$ SetOfStatement
    - $\rightarrow$ BooleanInformationSet

- **Examples**
  - SPARQL only (no inference)
  - RDF/RDFS/L2 inference
  - Urban shortest path calculator
  - CyC
LarKC Plug-in Types

- **Decide**
  - Responsible for constructing a pipeline and making decisions about its execution

- **Examples**
  - Scripted Decider (predefined pipeline)
  - Meta-reasoning Decider (dynamic pipeline based on metadata)
LarKC Plug-in Descriptions

SAWSDL + WSMO = WSMO-Lite

- Functional
- Non Functional
- Behavioral
LarKC Local Execution

Decider

Pipeline Support System
Plug-in Registry

Local Plug-in Manager
Local Plug-in Manager
Local Plug-in Manager
Local Plug-in Manager
Local Plug-in Manager

Plug-in API
Query Transformer
Plug-in API
Identifier
Plug-in API
Info. Set Transformer
Plug-in API
Selector
Plug-in API
Reasoner
Web Scale Reasoning on Linked Life Data

ILLUSTRATION BY A LARGER EXAMPLE
Quick Facts

- **LinkedLifeData stands (LLD) for platform to:**
  - Operate with heterogeneous data sets
  - Allow semantic data integration
  - Provide tools for knowledge access and management
  - Compliant with W3C standards and recommendations

- **Pathway and Interaction Knowledge Base (PIKB) is:**
  - Knowledge base to integrates information for gene, proteins, pathways and functional annotations
  - Is used as demonstration services
  - Publicly accessible in Internet
Objectives

- Support incremental extension of the knowledge base with highly heterogeneous data sets
- Allow straightforward updates of the information
- Provide scientists with computational support to conceptualize the breath and depth of relationships between data
- Scale up to billions of statements
Sometimes we need to ask far more questions efficiently:

Give me all proteins which interacts in cellular structure and are annotated with repressor and have at least one participants that is encoded by gene annotated with specific term and is located in chromosome X? Filter the results for Mammalia organisms!
<table>
<thead>
<tr>
<th>Database</th>
<th>Dataset</th>
<th>Schema</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniprot</td>
<td>Curated entries</td>
<td>Original by the provider</td>
<td>Protein sequences and annotations</td>
</tr>
<tr>
<td>Entrez-Gene</td>
<td>Complete</td>
<td>Custom RDF schema</td>
<td>Genes and annotation</td>
</tr>
<tr>
<td>iProClass</td>
<td>Complete</td>
<td>Custom RDF schema</td>
<td>Protein cross-references</td>
</tr>
<tr>
<td>Gene Ontology</td>
<td>Complete</td>
<td>Schema by the provider</td>
<td>Gene and gene product annotation thesaurus</td>
</tr>
<tr>
<td>BioGRID</td>
<td>Complete</td>
<td>BioPAX 2.0 (custom generated)</td>
<td>Protein interactions extracted from the literature</td>
</tr>
<tr>
<td>NCI - Pathway Interaction Database</td>
<td>Complete</td>
<td>BioPAX 2.0 (original by the provider)</td>
<td>Human pathway interaction database</td>
</tr>
<tr>
<td>The Cancer Cell Map</td>
<td>Complete</td>
<td>BioPAX 2.0 (original by the provider)</td>
<td>Cancer pathways database</td>
</tr>
<tr>
<td>Reactome</td>
<td>Complete</td>
<td>BioPAX 2.0 (original by the provider)</td>
<td>Human pathways and interactions</td>
</tr>
<tr>
<td>BioCarta</td>
<td>Complete</td>
<td>BioPAX 2.0 (original by the provider)</td>
<td>Pathway database</td>
</tr>
<tr>
<td>KEGG</td>
<td>Complete</td>
<td>BioPAX 1.0 (original by the provider)</td>
<td>Molecular Interaction</td>
</tr>
<tr>
<td>BioCyc</td>
<td>Complete</td>
<td>BioPAX 1.0 (original by the provider)</td>
<td>Pathway database</td>
</tr>
<tr>
<td>NCBI Taxonomy</td>
<td>Complete</td>
<td>Custom RDF schema</td>
<td>Organisms</td>
</tr>
</tbody>
</table>
Challenges to Overcome

- **Syntactic**
  - The way the different are serialized

- **Structure**
  - The way the different entities are represented

- **Semantic**
  - The way the different entities are interpreted

- **W3C standard serialization formats for data exchange**
- **The graph model used by RDF gives maximum flexibility**
- **Support custom R-entailment rules to derive meaning**
• Currently operates over OWLIM semantic repository in future to be implemented LarKC
• LLD - PIKB statistics:
  – Number of statements: 1,159,857,602
  – Number of explicit statements: 403,361,589
  – Number of entities: 128,948,564
• Publicly available at: http://www.linkedlifedata.com
SUMMARY
Summary

• Traditional reasoning techniques are not for Web-scale
  – Small set of axioms
  – Small number of facts
  – Completeness of inferences rules
  – Trustworthiness correctness of inference rules and consistency
  – Static domain

• Web scale reasoning needs to deal with
  – Unsound and incomplete knowledge
  – Computational resource and time limitations
• Mandatory reading:
  – Z. Huang et al. D4.1 A Survey of Web Scale Reasoning. LarKC

• Further reading:
  – http://www.larck.eu
References

• Wikipedia links:
  – http://en.wikipedia.org/wiki/Logic
  – http://en.wikipedia.org/wiki/Horn_logic
<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Semantic Web Architecture</td>
</tr>
<tr>
<td>3</td>
<td>Resource Description Framework (RDF)</td>
</tr>
<tr>
<td>4</td>
<td>Web of data</td>
</tr>
<tr>
<td>5</td>
<td>Generating Semantic Annotations</td>
</tr>
<tr>
<td>6</td>
<td>Storage and Querying</td>
</tr>
<tr>
<td>7</td>
<td>Web Ontology Language (OWL)</td>
</tr>
<tr>
<td>8</td>
<td>Rule Interchange Format (RIF)</td>
</tr>
<tr>
<td>9</td>
<td>Reasoning on the Web</td>
</tr>
<tr>
<td>10</td>
<td><strong>Ontologies</strong></td>
</tr>
<tr>
<td>11</td>
<td>Social Semantic Web</td>
</tr>
<tr>
<td>12</td>
<td>Semantic Web Services</td>
</tr>
<tr>
<td>13</td>
<td>Tools</td>
</tr>
<tr>
<td>14</td>
<td>Applications</td>
</tr>
</tbody>
</table>
Questions?