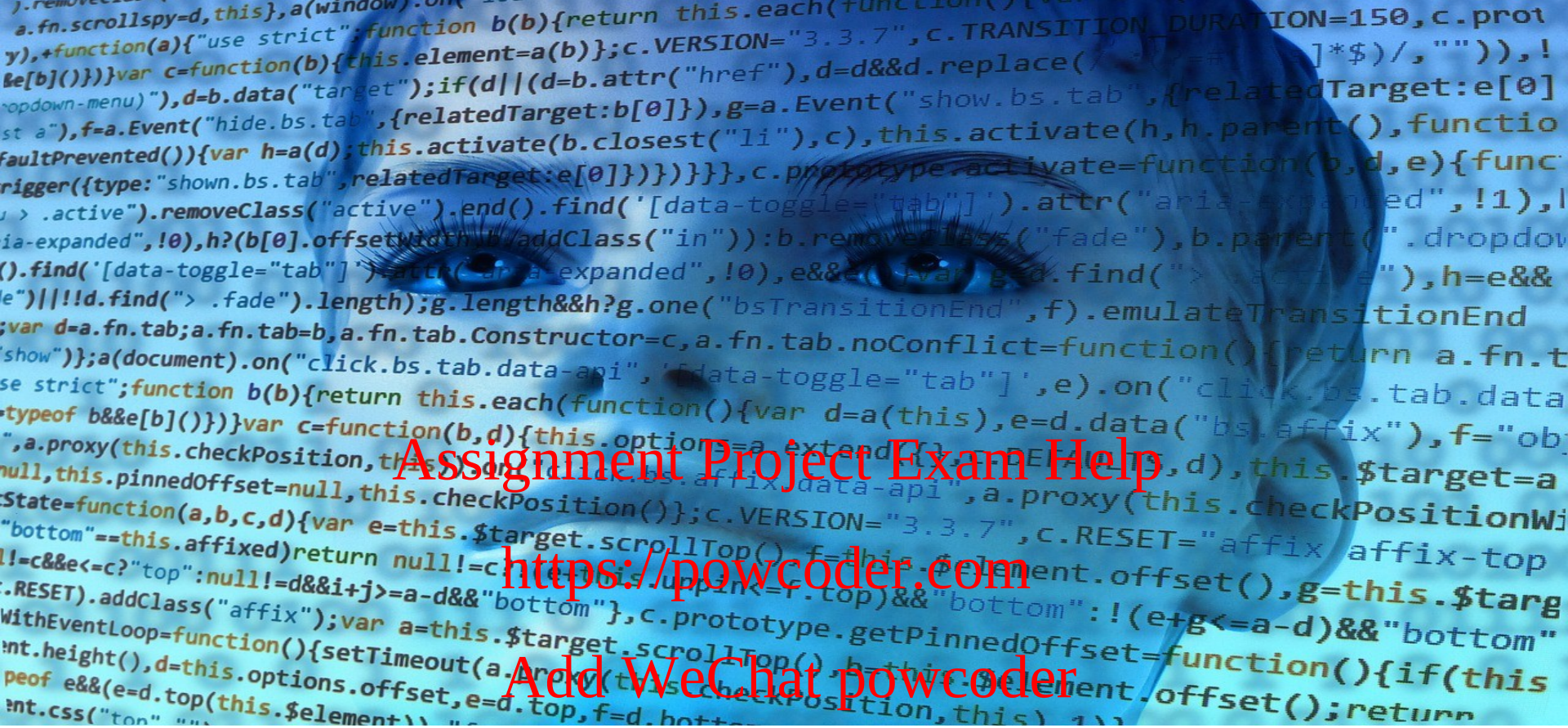
Dr. Davoud Mougouei
Decision Systems Lab
SCIT, EIS, UOW

Ph.D. in Software Engineering
M.Sc. in Computer Science
B.Eng. in Computer Engineering



[Engineering Human Values in Software through Value Programming (CHASE, 2020)]

Important Points about the Subject

1

Please read the subject outline carefully!

2

Please discuss your technical questions in the lab!

3

Please email for what cannot be discussed in the lab!

Lecture 1

The Semantic Web Vision & Structured Web Documents in XML

Chapters 1 &2 of

Grigoris Antoniou
Frank van Harmelen

Chapter 1
The Semantic Web Vision

- https://www.w3.org/standards/semanticweb/

Lecture Outline

1. **Today's Web**
2. The Semantic Web Impact
3. Semantic Web Technologies
4. A Layered Approach

What are the typical usages of the web?

Today's Web

- Today's web typical uses
  - seeking and making use of information,
  - searching for and getting in touch with other people,
  - reviewing catalogues of online stores ,
  - and ordering products by filling out forms
- Most of today's Web content is suitable for human consumption
  - Even Web content that is generated automatically from databases is usually presented without the original structural information found in databases

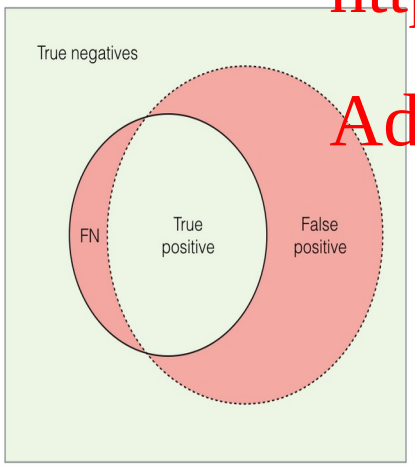
Keyword-Based Search Engines

- Current Web activities are not particularly well supported by software tools
  - Except for keyword-based search engines (e.g., Google)
- The Web's success relies on search engines.

What are the problems with the current Keyword-Based Search Engines?

Problems of Keyword-Based Search Engines

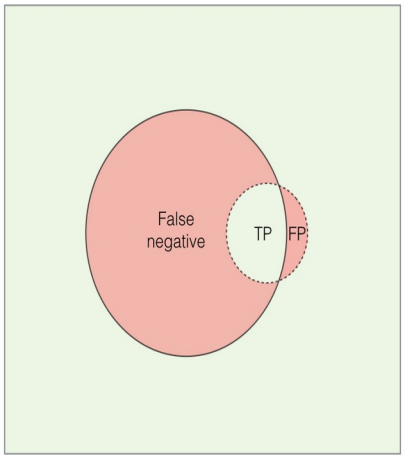
- Results are highly sensitive to vocabulary
- High recall but low precision
- Low recall



Recall = tp/(tp+fn)

Precision = tp/(tp+fp)

https://medium.com/



(A)

(B)

Problems of Keyword-Based Search Engines

- Human involvement is necessary to interpret and combine results
- Results of Web searches are not readily accessible by other software tools

The Key Problem of Today's Web

- The meaning of Web content is not machine-accessible: lack of semantics
- It is simply difficult to distinguish the meaning between these two sentences:

I am a lecturer of semantic web.

I am a lecturer of semantic web, you may think. Well, . . .

Three key points with Semantic Web Approach:

I. Representing Web content in a form that is more easily machine-processable.

II. Using intelligent techniques to take advantage of these representations.

III. Evolving out of the existing Web

Lecture Outline

1. Today's Web
2. **The Semantic Web Impact**
3. Semantic Web Technologies
4. A Layered Approach

The Semantic Web Impact – Knowledge Management

- Knowledge management concerns with (i) acquiring, (ii) accessing, and (iii) maintaining knowledge within an organization
- Key activity of large businesses: internal knowledge as an intellectual asset
- Most information is currently available in a weakly structured form (e.g. text, audio, video)

Limitations of Current Knowledge Management Technologies in four dimensions:

I. *Searching information:*

Keyword-based search engines

II. *Extracting information:*

human involvement necessary for browsing, retrieving, interpreting, combining

III. *Maintaining information:*

inconsistencies in terminology, outdated information.

IV. *Viewing information:*

Impossible to define views on Web knowledge

Semantic Web Enabled Knowledge Management

i) organizing knowledge in conceptual spaces according to its meaning.

ii) having automated tools for maintenance and knowledge discovery

iii) answering any query semantically

iv) answering any query over several documents

v) defining who may view certain parts of information (even parts of documents) will be possible.

The Semantic Web Impact – B2C Electronic Commmerce

- A typical scenario: user visits one or several online shops, browses their offers, selects and orders products.
- Ideally humans would visit all, or all major online stores, but time consuming
- Shopbots are a useful tool:
- https://www.shopbot.com.au/

Limitations of Shopbots

- They rely on wrappers: extensive programming required
- Wrappers need to be reprogrammed when an online store changes its outfit
- Wrappers extract information based on textual analysis
  - Error-prone
  - Limited information extracted

Semantic Web Enabled B2C Electronic Commerce

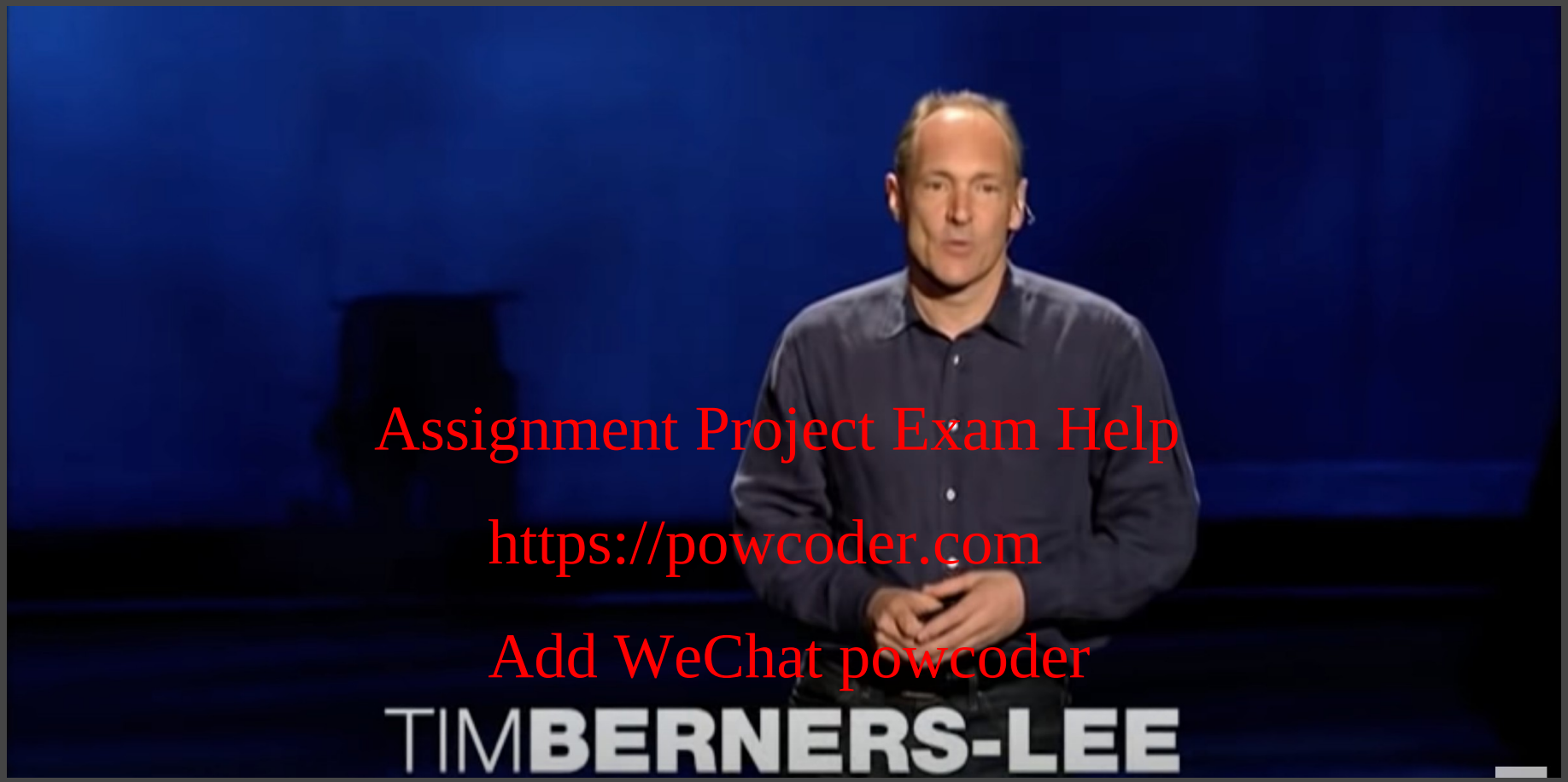
- Software agents that can interpret the product information and the terms of service.
  - Pricing and product information, delivery and privacy policies will be interpreted and compared to the user requirements.
- Information about the reputation of shops
- Sophisticated shopping agents will be able to conduct automated negotiations

The Semantic Web Impact – B2B Electronic Commerce

- Currently relies mostly on EDI (Electronic Data Interchange)
  - Isolated technology, understood only by experts
  - Difficult to program and maintain, error-prone
  - Each B2B communication requires separate programming

Semantic Web Enabled B2B Electronic Commerce (Advantages)

I. Businesses enter partnerships without much overhead
II. Differences in terminology will be resolved using standard abstract domain models
III. Data will be interchanged using translation services.
IV. Auctioning, negotiations, and drafting contracts will be carried out automatically (or semi-automatically) by software agents



- https://www.youtube.com/watch?v=OM6XIICm_qo
- https://www.youtube.com/watch?v=HeUrEh-nqtU
- https://www.youtube.com/watch?v=Zy67j0ysBfo

Lecture Outline

1. Today's Web
2. The Semantic Web Impact
3. **Semantic Web Technologies**
4. A Layered Approach

Semantic Web Technologies

- Explicit Metadata
- Ontologies
- Logic and Inference
- Agents

On HTML

- Web content is currently formatted for human readers rather than programs.
- HTML is the predominant language in which Web pages are written (directly or using tools)
- Vocabulary describes presentation

An HTML Example

```
<h1>Agilitas Physiotherapy Centre</h1>
Welcome to the home page of the Agilitas Physiotherapy Centre. Do
you feel pain? Have you had an injury? Let our staff Lisa Davenport,
Kelly Townsend (our lovely secretary) and Steve Matthews take care
of your body and soul.
<h2>Consultation hours</h2>
Mon 11am - 7pm<br>
Tue 11am - 7pm<br>
Wed 3pm - 7pm<br>
Thu 11am - 7pm<br>
Fri 11am - 3pm<p>
But note that we do not offer consultation during the weeks of the
<a href=". . .">State Of Origin games </a>
```

Problems with HTML

- Humans have no problem with this
- Machines (software agents) do:
  - How distinguish therapists from the secretary,
  - How determine exact consultation hours
  - They would have to follow the link to the State Of Origin games to find when they take place.

A Better Representation

```
<company>
    <treatmentOffered>Physiotherapy</treatmentOffered>
    <companyName>Agilitas Physiotherapy
    Centre</companyName>
    <staff>
            <therapist>Lisa Davenport</therapist>
            <therapist>Steve Matthews</therapist>
            <secretary>Kelly Townsend</secretary>
    </staff>
</company>
```

Explicit Metadata

- This representation is far more easily processable by machines
- Metadata: data about data
  - Metadata capture part of the *meaning of data*
- Semantic Web **does not** rely on text-based manipulation, **but rather** on machine-processable metadata

Ontologies

The term ontology originates from philosophy

- The study of the nature of existence

Different meaning from computer science

- An ontology is an explicit and formal specification of a conceptualization

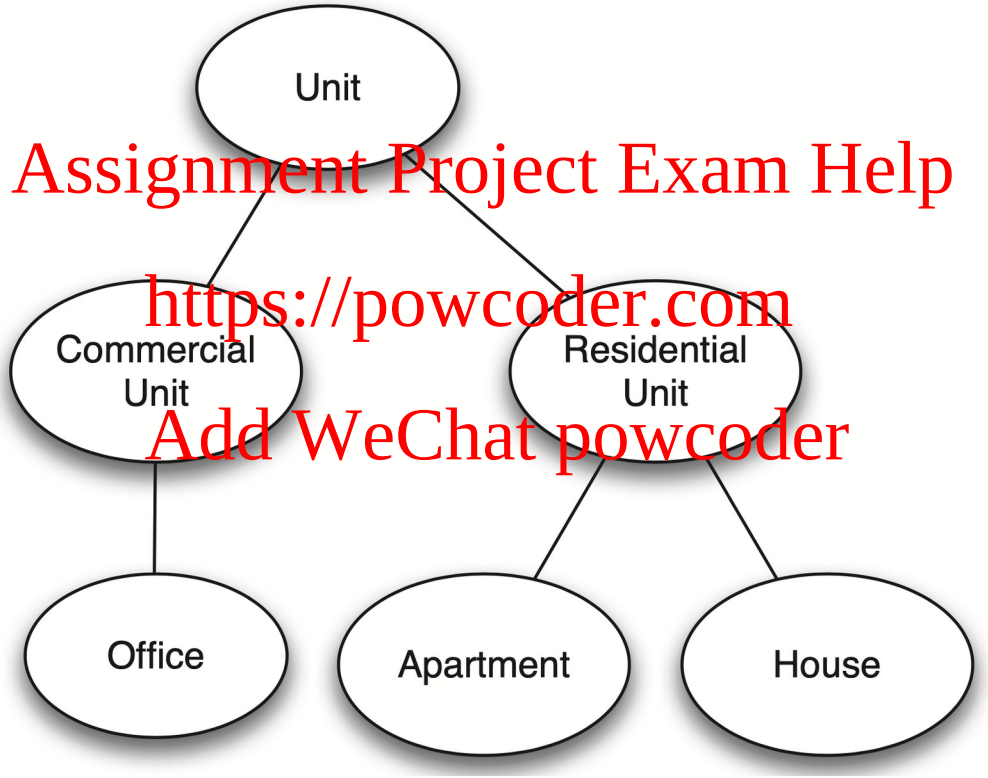
Typical Components of Ontologies

- Terms denote important concepts (classes of objects) of the domain
  - e.g. professors, staff, students, courses, departments
- Relationships between these terms: typically class hierarchies
  - a class C to be a subclass of another class C' if every object in C is also included in C'
  - e.g. all professors are staff members

Further Components of Ontologies

- Properties:
  - e.g. X teaches Y
- Value restrictions
  - e.g. only faculty members can teach courses
- Disjointness statements
  - e.g. faculty and general staff are disjoint
- Logical relationships between objects
  - e.g. every department must include at least 10 faculty members.

Example of a Class Hierarchy



The Role of Ontologies on the Web

- Ontologies provide a shared understanding of a domain: **semantic interoperability**
  - overcome differences in terminology
  - mappings between ontologies
- Ontologies are useful for the organization and navigation of Websites

The Role of Ontologies in Web Search

- Ontologies are useful for improving the accuracy of Web searches
  - search engines can look for pages that refer to a precise concept in an ontology
- Web searches can exploit generalization/specialization information
  - If a query fails to find any relevant documents, the search engine may suggest to the user a more general query.
  - If too many answers are retrieved, the search engine may suggest to the user some specializations.

What is the difference between Ontology and Taxonomy?

Web Ontology Languages

RDF (Resource Description Framework) Schema

- RDF is a data model for objects and relations between them
- RDF Schema is a vocabulary description language which

  1) Describes properties and classes of RDF resources,

  2) Provides semantics for generalization hierarchies of properties and classes

Web Ontology Languages (2)

OWL (**Web** Ontology Language) **OWL** is built on top of RDF

- A richer ontology language
- relations between classes
  - e.g., disjointness
- cardinality
  - e.g. “exactly one”
- richer typing of properties
- characteristics of properties (e.g., symmetry)

Logic and Inference

- Logic is the discipline that studies the principles of reasoning
- Formal languages for expressing knowledge
- Well-understood formal semantics
  - Declarative knowledge: we describe what holds without caring about how it can be deduced
- Automated reasoners can deduce (infer) conclusions from the given knowledge

An Inference Example

prof(X) → faculty(X)

faculty(X) → staff(X)

prof(michael)

We can deduce the following conclusions:

faculty(michael)

staff(michael)

prof(X) → staff(X)

Logic versus Ontologies

- The previous example involves knowledge typically found in ontologies
  - Logic can be used to uncover ontological knowledge that is implicitly given
  - It can also help uncover unexpected relationships and inconsistencies
- Logic is more general than ontologies
  - It can also be used by **intelligent agents** for making decisions and selecting courses of action

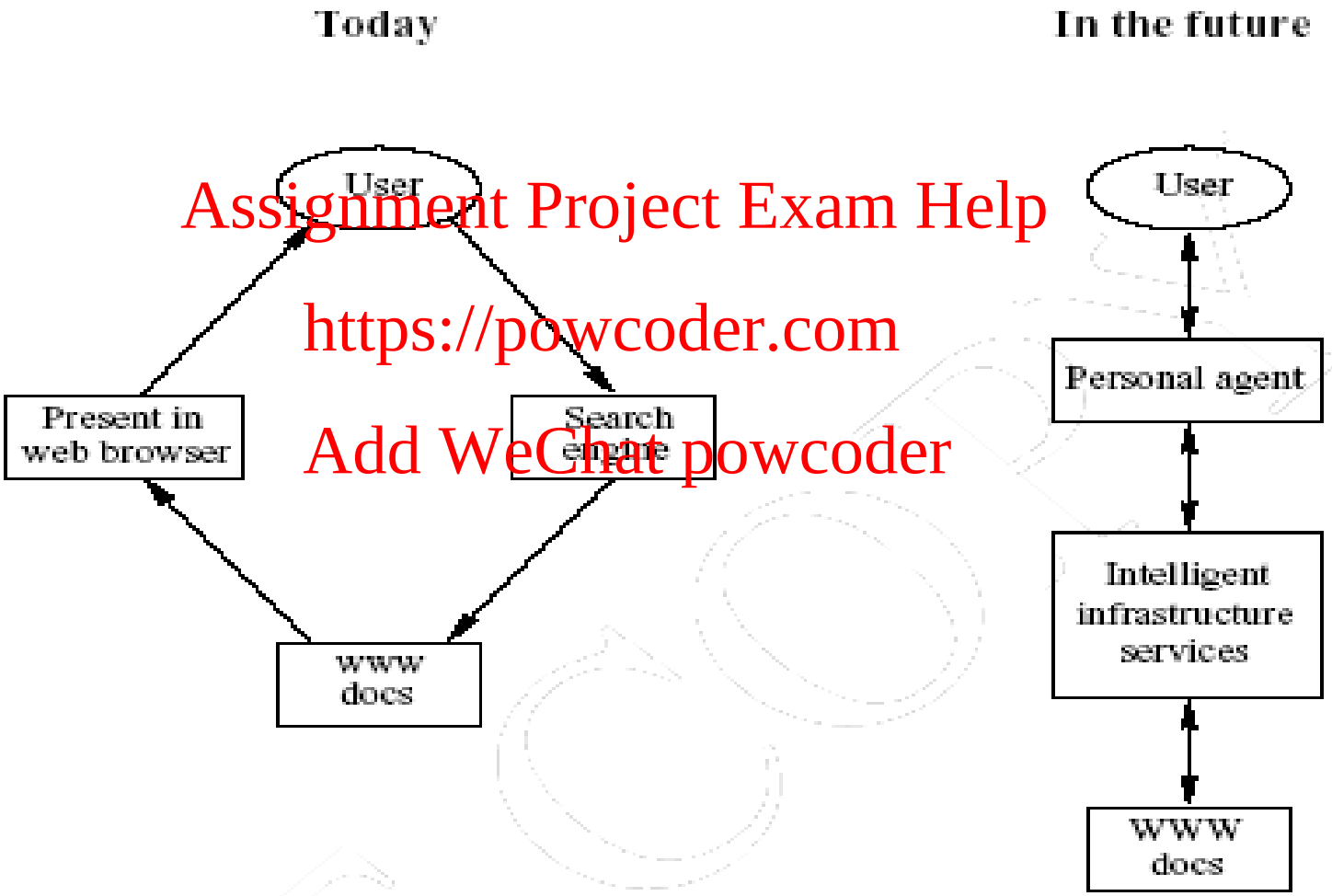
Inference and Explanations

- An important advantage of logic is that it can provide *explanations* for conclusions
- Explanations: a series of inference steps that can be retraced
- They increase users' confidence in Semantic Web agents.
- Activities between agents: create or validate proofs

Software Agents

- Software agents work autonomously and proactively
  - They evolved out of **object oriented** and **componentent-based programming**

Intelligent Personal Agents



Semantic Web Agent Technologies

- Metadata
  - Identify and extract information from Web sources
- Ontologies
  - Search websites and interpret retrieved information
  - Communicate with other agents
- Logic
  - Process retrieved information, draw conclusions

What will a personal agent on the Semantic Web do? Name its operations one after another chronically (in a time-based sequence).

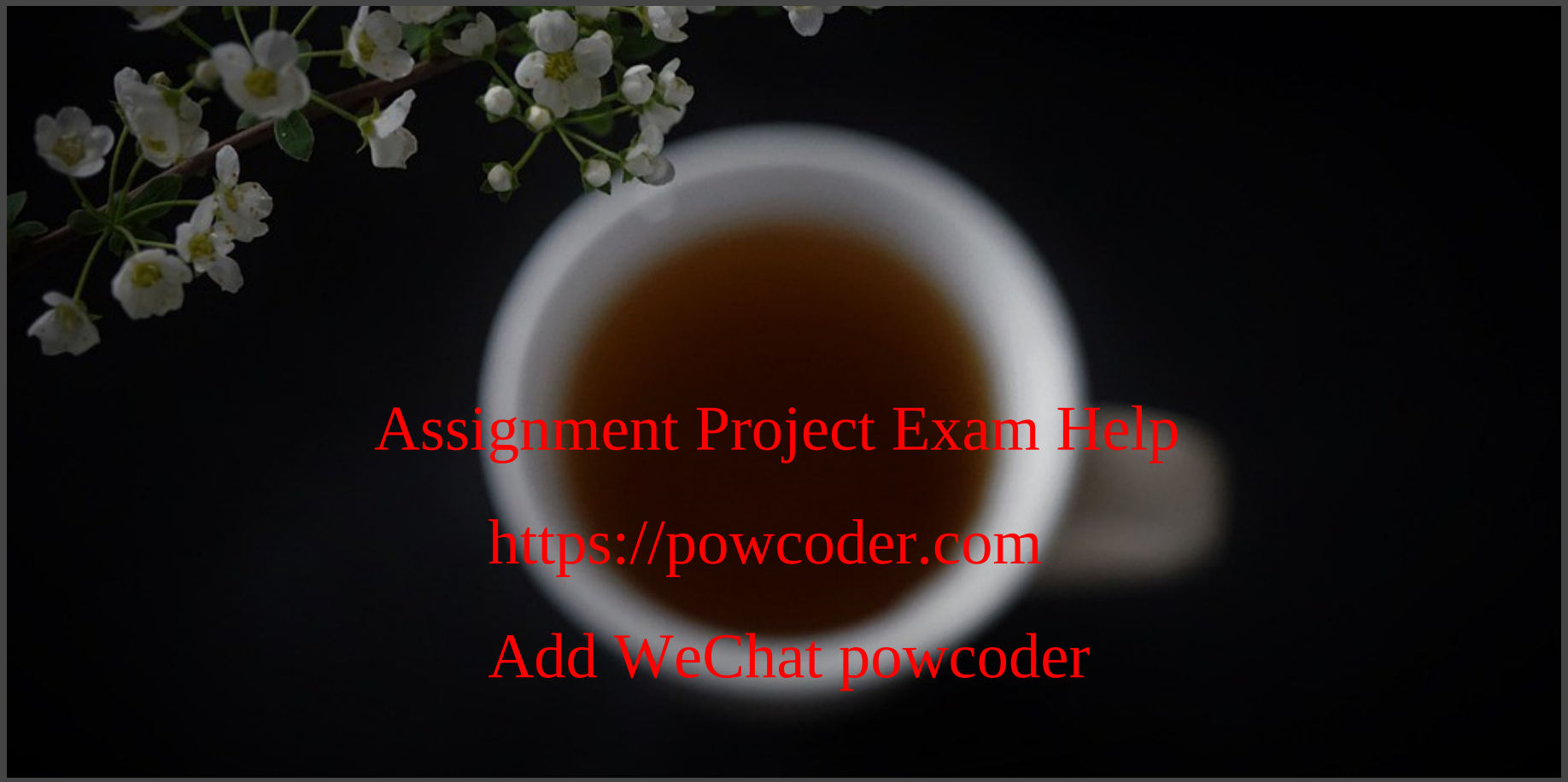
A personal agent on the Semantic Web will:

I. receive some tasks and preferences from the person

II. seek information from Web sources, communicate with other agents

III. compare information about user requirements and preferences, make certain choices

IV. give answers to the user

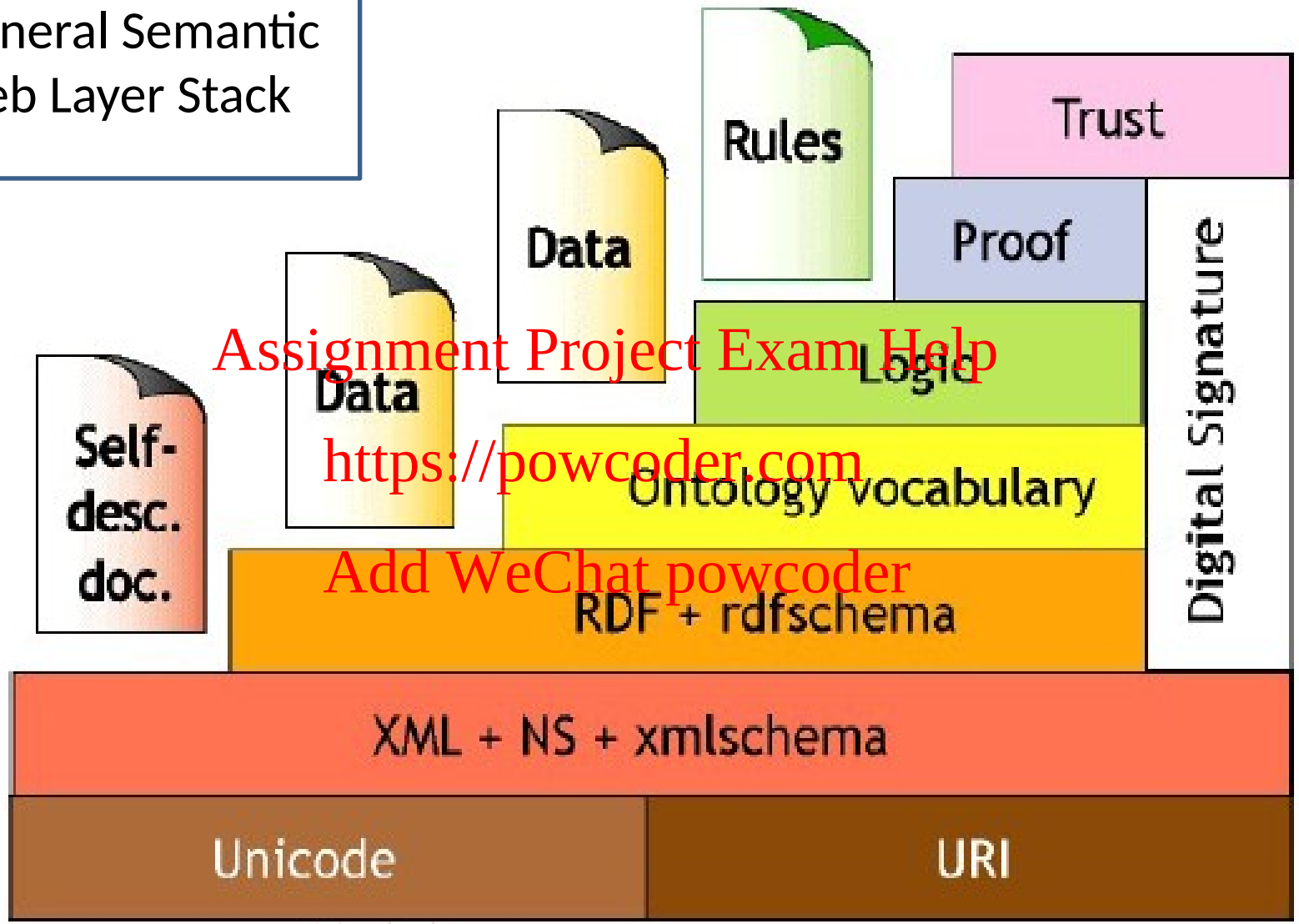

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Take a break!

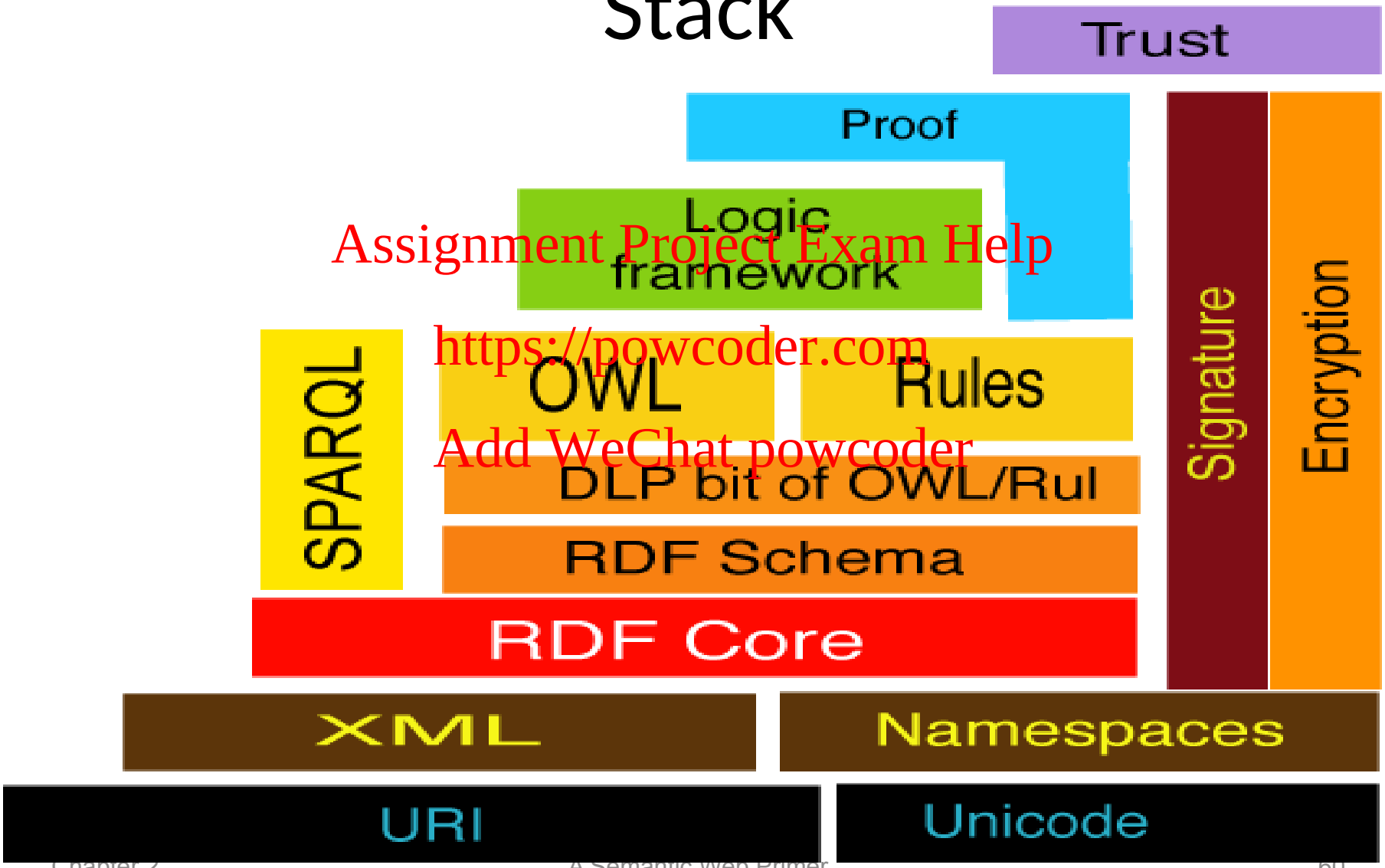
Lecture Outline

1. Today's Web
2. The Semantic Web Impact
3. Semantic Web Technologies
4. **A Layered Approach**

A General Semantic Web Layer Stack



A specific Semantic Web Layer Stack



Semantic Web Layers

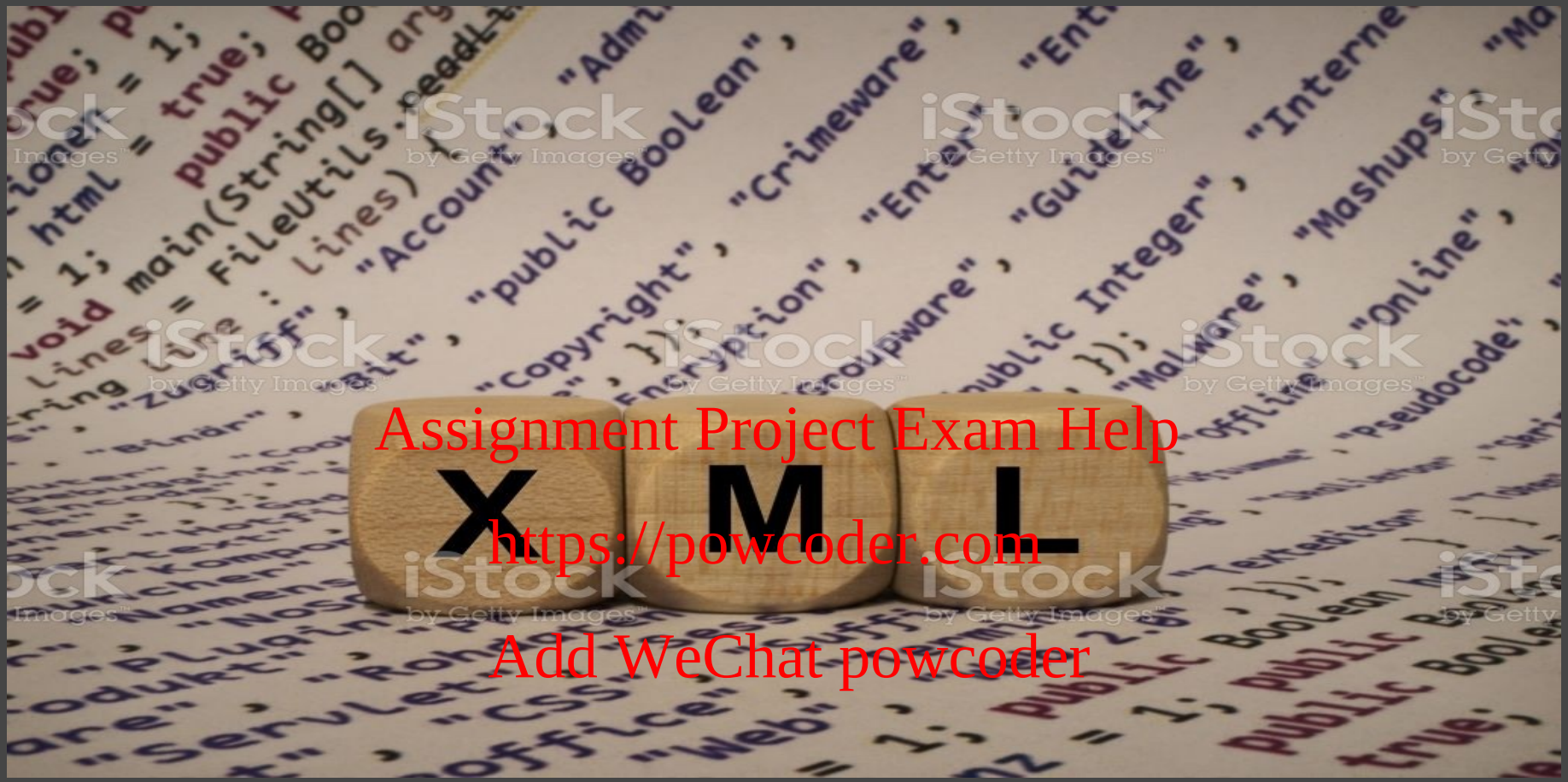
- XML layer
  - Syntactic basis
- RDF layer
  - RDF basic data model for facts
  - RDF Schema simple ontology language
- Ontology layer
  - More expressive languages than RDF Schema
  - Current Web standard: OWL

Semantic Web Layers (2)

- Logic layer
  - enhance ontology languages further
  - application-specific declarative knowledge
- Proof layer
  - Proof generation, exchange, validation
- Trust layer
  - Digital signatures
  - recommendations, rating agencies ….

Chapter 2:
Structured Web Documents in XML

1. **Introduction**
2. Detailed Description of XML
3. Structuring
   a) DTDs
   b) XML Schema
4. Namespaces
5. Accessing, querying XML documents: XPath
6. Transformations: XSLT



https://www.youtube.com/watch?v=KeLiQXqVgMI

HTML vs XML: Structural Information

- HTML documents **do not contain** structural information: pieces of the document and their relationships.
- XML more easily **accessible to machines** because
  - Every piece of information is described.
  - Relations are also defined through the **nesting structure**.
  - E.g., the **<author>** tags appear within the **<book>** tags, so they describe properties of the particular book.

HTML vs XML: Structural Information (2)

- A machine processing the XML document would be able to deduce that
  - the **author** element refers to the enclosing **book** element
- XML allows the definition of constraints on values
  - E.g. a year must be a number of four digits

HTML vs XML: Formatting

- The HTML representation provides more than the XML representation:
  - The formatting of the document is also described
- The **main** use of an HTML document is to display information. It must define formatting
- XML: **separation of content from display**
  - same information can be displayed in different ways

Lecture Outline

1. Introduction
2. Detailed Description of XML
3. Structuring
   a) DTDs
   b) XML Schema
4. Namespaces
5. Accessing, querying XML documents: XPath
6. Transformations: XSLT

The XML Language

An XML document consists of:

A prolog and a number of elements

Prolog of an XML Document

The prolog consists of

- an XML declaration and
- an optional reference to external structuring documents


```
<?xml version="1.0" encoding="UTF-16"?>

<!DOCTYPE book SYSTEM "book.dtd">
```

XML Elements

- The “things” the XML document talks about
  - E.g. books, authors, publishers
- An element consists of:
  - an opening tag
  - the content
  - a closing tag

<lecturer>David Billington</lecturer>

Content of XML Elements

- Content may be text, **or** other elements, or nothing

```
<lecturer>
        <name>David Billington</name>
        <phone> +61 – 7 – 3875 507 </phone>
</lecturer>
```


- If there is no content, then the element is called empty; it is abbreviated as follows:
  <lecturer/> for **<lecturer></lecturer>**

Is every empty element meaningless? Why?

ANSWER

- An empty element **is not** necessarily meaningless
  - It may have **some properties** in terms of attributes
- An attribute is a **name-value pair** inside the opening tag of an element

<lecturer name="David Billington" phone="+61 – 7 – 3875 507"/>

XML Attributes: An Example

```
<order orderNo="23456" customer="John Smith"
       date="October 15, 2002">
       <item itemNo="a528" quantity="1"/>
       <item itemNo="c817" quantity="3"/>
</order>
```

The Same Example without Attributes

```
<order>
    <orderNo>23456</orderNo>
    <customer>John Smith</customer>
    <date>October 15, 2002</date>
    <item>
            <itemNo>a528</itemNo>
            <quantity>1</quantity>
    </item>
    <item>
            <itemNo>c817</itemNo>
            <quantity>3</quantity>
    </item>
</order>
```

Further Components of XML Docs

- Comments
  - A piece of text that is to be ignored by parser
  - **<!-- This is a comment -->**
- Processing Instructions (PIs)
  - Define procedural attachments
  - **<?stylesheet type="text/css" href="mystyle.css"?>**

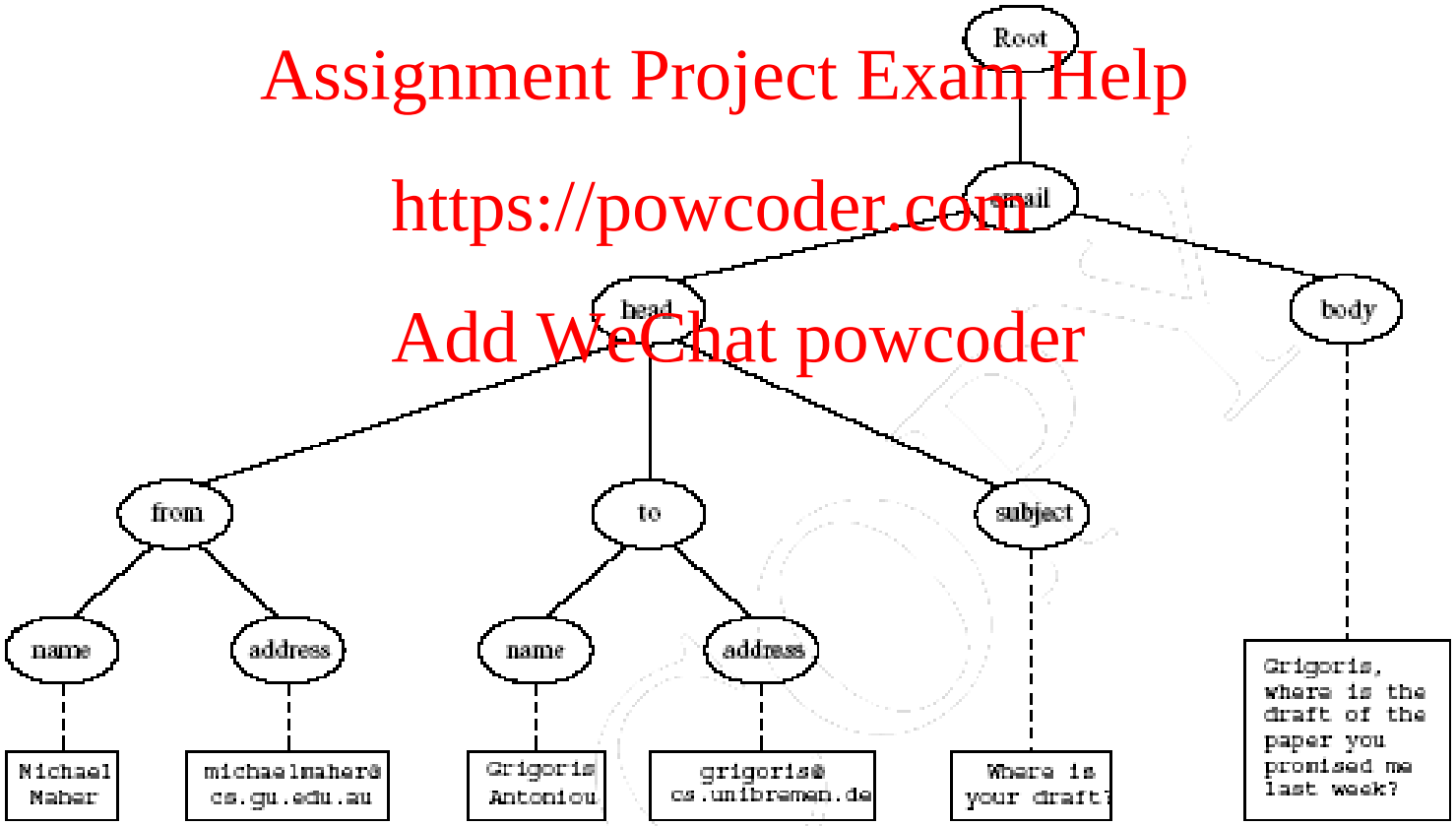
Well-Formed XML Documents

- Syntactically correct documents
- Some syntactic rules:
  - Only one outermost element (called root element)
  - Each element contains an opening and a corresponding closing tag
  - Tags may not overlap
    - **<author><name>Lee Hong</author></name>**
  - Attributes within an element have unique names

The Tree Model of XML Documents: An Example

```
<email>
    <head>
            <from name="Michael Maher"
                  address="michaelmaher@cs.gu.edu.au"/>
            <to name="Grigoris Antoniou"
                  address="grigoris@cs.uni-bremen.de"/>
            <subject>Where is your draft?</subject>
    </head>
    <body>
            Grigoris, where is the draft of the paper you promised me
            last week?
    </body>
</email>
```

The **Tree** Model of XML Documents: An Example (2)



The Tree Model of XML Docs

- The tree representation of an XML document is an ordered labeled tree:

  - There is exactly one root

  - There are no cycles
  - Each non-root node has exactly one parent
  - Each node has a label.
  - The order of elements is important
  - … but the order of attributes is not important

Lecture Outline

1. Introduction
2. Detailed Description of XML
3. **Structuring**
   a) **DTDs**
   b) XML Schema
4. Namespaces

Structuring XML Documents

https://www.w3schools.com/xml/xml_dtd.asp

- Define all the element and attribute names that may be used
- Define the structure
  - what values an attribute may take
  - which elements may or must occur within other elements, etc.
- If such structuring information exists, the document can be **validated**

Lecture Outline

1. Introduction
2. Detailed Description of XML
3. **Structuring**
   a) DTDs
   b) **XML Schema**
4. Namespaces

XML Schema

https://www.w3schools.com/xml/schema_intro.asp

- Significantly richer language for defining the structure of XML documents
- Its syntax is based on XML itself
  - not necessary to write separate tools
- Reuse and refinement of schemas
  - Expand or delete already existent schemas
- Sophisticated set of data types, compared to DTDs (which only supports strings)

Lecture Outline

1. Introduction
2. Detailed Description of XML
3. Structuring
   a) DTDs
   b) XML Schema
4. **Namespaces**

Namespaces

https://www.w3schools.com/xml/xml_namespaces.asp

- An XML document may use more than one DTD or schema
- Since each structuring document was developed independently, name **clashes** may appear.
- The **solution** is to use a different prefix for each DTD or schema.
  - **prefix:name**

An Example

```
<vu:instructors        xmlns:vu="http://www.vu.com/empDTD"
                       xmlns:gu="http://www.gu.au/empDTD"
                       xmlns:uky="http://www.uky.edu/empDTD">

<uky:faculty   uky:title="assistant professor"
                       uky:name="John Smith"
                       uky:department="Computer Science"/>

<gu:academicStaff      gu:title="lecturer"
                              gu:name="Mate Jones"
                              gu:school="Information Technology"/>

</vu:instructors>
```

Namespace Declarations

- Namespaces are declared within an element and can be used in that element and any of its children (elements and attributes)
- A namespace declaration has the form:
  - **xmlns:prefix="location"**
  - **location** is the address of the DTD or schema
- If a prefix is not specified: **xmlns="location"** then the **location** is used by default