Semi structured data, XML, RDF

***Describe general properies of semi-structured data.***

Irregular structure: consist heterogeneous elements, some may be incomplete, some may record extra information.

Implicit structure: some computation is required to obtain the data, the correspondence between the parse-tree and the logical representation of the data is not always immediate.

Partial structure: parts of the data may lack structure; other parts may only unveil some very sketchy structure.

A posteriori ’data guide’: data guide can’t be ignored while querying, and it changes fast, object can change type/class, difference between data guide and data is blurred.

***Model a given scenario or set of data in the OEM model.***

***Explain a Lorel query.***

Lorel was developed at Stanford to enable queries over semistructured OEM databases, based on OQL, with modifications and enhancements to support semistructured data.

The main novelties of the Lorel language are: (i) the extensive use of coercion to relieve the user from the strict typing of OQL, which is inappropriate for semistructured data; and (ii) powerful path expressions, which permit a flexible form of declarative navigational access and are particularly suitable when the details of the structure are not known to the user. Lorel also includes a declarative update language.

***OEM Definition:***

OEM itself is not particularly original, and the work presented here adapts easily to any graph-structured data model. In OEM, each object contains an object identifier (oid) and a value. A value may be atomic or complex.

***Describe the general idea and define the properties of a data guide.***

A DataGuide for an OEM source object s is an OEM object d such that every label path of s has exactly one data path instance in d, and every label path of d is a label path of s.

Conciseness: specify that a DataGuide describes every unique label path of a source exactly once, regardless of the number of times it appears in that source.

Accuracy: specify that the DataGuide encodes no label path that does not appear in that source.

Convenience: require that a DataGuide itself be an OEM object so we can store and access it using the same techniques available for processing OEM databases.

***Define minimal and strong data guides.***

Minimal data guide: the smallest possible dataGuide.

Strong data guides: Intuitively, we are interested in DataGuides where each set of label paths that share the same (singleton) target set in the DataGuide is exactly the set of label paths that share the same target set in the source.

***Given a description of data, construct minimal and strong data guides for that data.***

***Describe the general properties and concepts of XML.***

XML stands for EXtensible Markup Language.

XML was designed to store and transport data.

XML was designed to be both human- and machine-readable.

Order tree: similar to semi-structured proposal

Extensible: new kinds of data can be integrated

Flexible: easy to mix different kinds of data.

***DTD Definition:***

Document Type Definition defines the legal building blocks of an XML document, defines the document structure with a list of legal elements and attributes.

Elements, Attributes, Entities, PCDATA- parsed character data, CDATA- character data.

***XML Schema vs. DTD***

XML Schema are extensible to future additions, XML Schema are richer and more powerful than DTDs, are written in XML, support data types, support namespaces.

Facilitate information exchange.

***RDF Definition:***

Framework for describing resources on the web, designed to be read and understood by computers, not designed for being displayed to people. Written in XML, a W3C recommendation.

Resource have a URI, Property is a resource that has a name, a property value is the value of property.

Data comes as a set of triples (subject, predicate, object)

● Subject: resources ● Predicate: properties ● Object: literals or resources

***Model a given scenario (text or ER) using XML.***

***Model a given scenario (text or ER) using RDF.***

***Use and understand DTD, XML schema, RDF schema***

***Write an XPath query.***

***Write an XQuery query.***

***Explain a SPARQL query; show the result of a SPARQL query on given data.***

Pattern matching

***Explain what Linked (Open) Data is.***

*Structured data, interlinked data on the WEB.*

*The RDF data model and URIs*

*The SPARQL Query Language*

*The linked data publishing principles*