

W3. Check List — Junyi

- ▼ design issues in schema design
 - ▼ common
 - ▼ RDF unique issues
 - ▼ adopt a scheme for IRIs
 - ▼ PG unique issues
 - ▼ decide whether a value should be represented as a property or as a node
- ▼ Design Workflow
 - ▼ RDF
 - ▼ Guideline: linked data principles
 - ▼ Use URIs as names for things.
 - ▼ identify the items of interest (resources: information/non-information)
 - ▼ constructed URI be like (3 features)
 - ▼ simple
Short, mnemonic
 - ▼ stable
remain this way as long as possible → it is best to keep implementation-specific bits and pieces such as ".php" and ".asp" out of the URIs.
 - ▼ manageable
be defined in a way that they can be fully managed by the publisher.
 - ▼ Use HTTP URIs so that people can look up those names.
 - ▼ We identify resources using Uniform Resource Identifiers (URIs). We restrict ourselves to using HTTP URIs only and avoid other URI schemes

- ▼ looking up names [URI dereferencing] return
 - ▼ non-information resource → its description in RDF expressed in an XML notation.
 - ▼ information object → representation of its current state (e.g., a text document, an image, a video, etc.)
- ▼ When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
 - ▼ return
 - ▼ a knowledge graph in RDF. The data should reuse standardized vocabularies to name the IRIs used in describing the RDF data.
 - ▼ pre-existing vocabularies and/or creating a new vocabulary
- ▼ Include links to other URI s, so that they can discover more things.
 - ▼ kinds of links:
 - ▼ relationship links
 - ▼ foaf:
 - ▼ point at related things in other data sources such as other people, places or genes. → background info, bibliographic data...
 - ▼ identity links
 - ▼ #sameAs
 - ▼ point at URI aliases used by other data sources to identify the same real-world object or abstract concept → retrieve further descriptions about an entity, and serve an important social function as they enable different views of the world to be expressed on the WWW of Data.
 - ▼ vocabulary links
 - ▼ rdfs:
 - ```
@prefix dbpedia: <http://dbpedia.org/ontology/>
big:sme#SmallMediumEnterprise Gifs:subClassOf dbpedia:Company
```

▼ to the definitions of the vocabulary terms that are used to represent the data, as well as from these definitions to the definitions of related terms in other vocabularies. → make data self-descriptive and enable Linked Data applications to understand and integrate data across vocabularies.

## ▼ PG

▼ choosing nodes, node labels, node properties; edges and edge properties.

▼ The basic design questions are

▼ whether to model a piece of information as a property, label or as a separate object

▼ considerations

▼ naturalness of labels, whether the labels might change over a period of time, runtime query performance, and the cardinality of values. → Queries can work with these sets instead of the whole graph

▼ a set, it can be viewed as a class. The question of whether to introduce a new label can be restated as whether to introduce a new class?

▼ labels

▼ The labels in a property graph model are used to group nodes into sets. All nodes labeled with the same label belong to the same set.

▼ When to introduce Relationships between Objects

▼ considerations: (1) the membership in the class changes with time; (2) wish to achieve better query performance

▼ when to introduce relation properties (associating weights or confidence with a relationship or to associate provenance or other meta data with a relationship.)

▼ If the use case is such that much of the query evaluation can be done without using the relationship properties, and they are required only for final filtering of the results

▼ how to handle higher arity relationships (Handling non-binary Relationships)

▼ reification for relationships with any arity

creating a new node type for the relation, and by introducing node properties for the different arguments of that relation.