

UNIT 6

Knowledge Sharing

(9 Hours) — Knowledge Engineering (CACS458)

⚠ Please Note

These notes are generated to help with revision and **may contain errors or omissions**. Please verify important definitions, formulas and examples against your textbook, lecture notes and the official TU syllabus before relying on them in an exam.

6.1 Information Distribution and Integration

Knowledge sharing is the exchange of knowledge between people, organisations and software systems so that it can be reused. It requires that information is first *distributed* to those who need it and then *integrated* so that data from different sources can work together.

Information Distribution

- Delivering the right information to the right user at the right time.
- **Channels**: intranets, portals, e-mail, wikis, repositories, publish–subscribe feeds, the Web.
- **Push** model (system sends updates, e.g. alerts, RSS) versus **pull** model (user searches or requests).
- **Distributed knowledge-based systems**: several knowledge bases or agents on a network cooperate to solve problems.

Information Integration

Information integration combines data from multiple heterogeneous sources and presents a unified view to the user.

Type of heterogeneity	Description	Example
Syntactic	Different formats or languages	CSV vs XML vs relational tables
Structural (schema)	Different schema design	One table ‘Student’ vs two tables ‘Person’ and ‘Enrolment’
Semantic	Same term, different meaning, or different terms, same meaning	‘Class’ = classroom or a group of students; ‘Teacher’ vs ‘Instructor’

Approaches to Integration

- **Data warehouse** — copy and transform data into one central store (ETL: extract, transform, load).
- **Mediator / wrapper (virtual integration)** — a global schema queries the sources on demand through wrappers.
- **Peer-to-peer integration** — peers map their schemas directly to each other.
- **Ontology-based integration** — a shared ontology defines the common meaning; sources are mapped to it (global-as-view, local-as-view or hybrid).
- **Linked data** — publishing data on the web as interlinked RDF.

Note: Ontology mapping and alignment (finding equivalent classes and properties between ontologies) is the key technique for resolving semantic heterogeneity.

6.2 Semantic Web and Its Applications

The **Semantic Web**, proposed by Tim Berners-Lee, is an extension of the current Web in which information is given **well-defined meaning** so that computers and people can work in cooperation. Where the ordinary Web links *documents* for humans to read, the Semantic Web links *data* that machines can understand.

Basis	Traditional Web (Web 1.0/2.0)	Semantic Web (Web 3.0 view)
Content	HTML documents for humans	Machine-readable data with meaning
Links	Between pages	Between data items and concepts
Search	Keyword matching	Meaning-based, can answer questions
Technologies	HTML, HTTP, CSS	RDF, RDFS, OWL, SPARQL, URIs

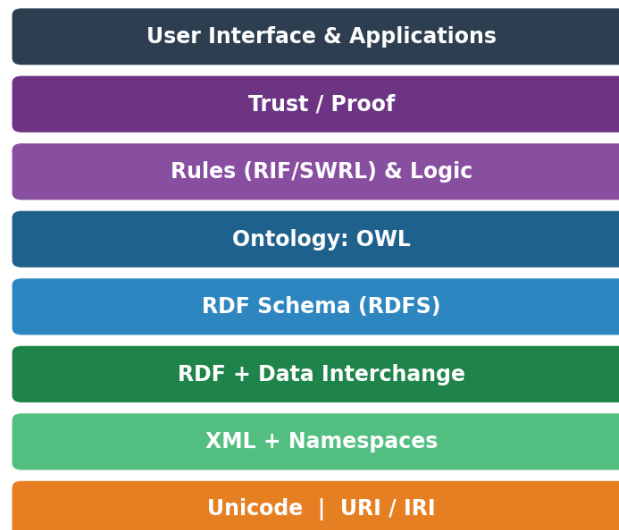


Figure 6.1: The Semantic Web layer cake (stack).

- **URI/IRI** gives a global identifier to every resource; **Unicode** allows all languages.
- **XML** provides syntax for structured documents.
- **RDF, RDFS** give a data model and a simple schema vocabulary; **OWL** gives rich ontologies.
- **SPARQL** is the query language; **rules, logic, proof and trust** are higher layers.

Applications of the Semantic Web

- **Semantic search** and question answering (knowledge panels in search engines).
- **Data integration** in enterprises, e-government and libraries.
- **Healthcare and life sciences** — linking drug, gene and disease databases.
- **E-commerce** — schema.org markup and Google's Knowledge Graph help products and offers to be found.
- **E-learning** — personalised resources for students.
- **Intelligent agents** — automatically scheduling, comparing and booking services.
- **Cultural heritage and digital libraries** — Europeana, Dublin Core metadata.

6.2.1 RDF and Linked Data

The **Resource Description Framework (RDF)** is the W3C data model for the Semantic Web. Every statement is a **triple: subject – predicate – object**. Subjects and predicates are identified by URIs; an object is a URI or a literal value. A collection of triples forms a directed, labelled graph.

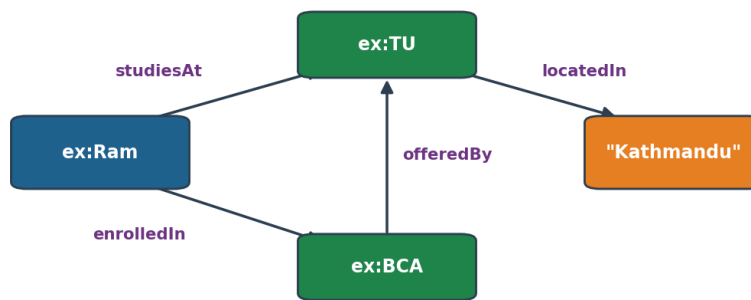


Figure 6.2: An RDF graph — each arrow with its two nodes is one triple.

Subject	Predicate	Object
ex:Ram	ex:studiesAt	ex:TU
ex:TU	ex:locatedIn	"Kathmandu"
ex:Ram	ex:enrolledIn	ex:BCA
ex:BCA	ex:offeredBy	ex:TU

RDF Serialisations

- **RDF/XML**, **Turtle** (compact, human-friendly), **N-Triples**, **JSON-LD** (for web APIs).
- **Blank nodes** represent unnamed resources; **literals** may carry datatypes (xsd:integer) or language tags ("Kathmandu"@en).
- **Reification** lets one make statements about statements.

Example 6.1 — Turtle syntax

```

@prefix ex: <http://example.org/> .
ex:Ram ex:studiesAt ex:TU .
ex:Ram ex:enrolledIn ex:BCA .
ex:TU ex:locatedIn "Kathmandu" .

```

Meaning

Three triples: Ram studies at TU; Ram is enrolled in BCA; TU is located in Kathmandu. Together they form the graph of Figure 6.2.

RDF Schema (RDFS)

RDFS adds a basic vocabulary to describe classes and properties: *rdfs:Class*, *rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range*, *rdfs:label*, *rdfs:comment*. It enables simple inference: if *ex:Student rdfs:subClassOf ex:Person* and *ex:Ram rdf:type ex:Student*, then *ex:Ram* is also an *ex:Person*.

SPARQL

SPARQL is the query language for RDF. Example: *SELECT ?name WHERE { ?s ex:studiesAt ex:TU . ?s ex:hasName ?name }* — returns names of everyone who studies at TU.

Linked Data

Linked Data is a way of publishing structured data on the Web so that it can be interlinked and queried. Berners-Lee's four rules:

- Use **URIs** as names for things.
- Use **HTTP URIs** so that people can look those names up.
- When someone looks up a URI, provide useful information using standards (**RDF**, **SPARQL**).
- Include **links to other URIs** so more things can be discovered.

Together these form the **Linking Open Data (LOD) cloud**, with datasets such as DBpedia, Wikidata, GeoNames and Open Government data. **Five-star open data** scheme: (1) open licence on the web, (2) machine-readable, (3) non-proprietary format, (4) uses RDF/URIs, (5) linked to other data.

*Note: **owl:sameAs** is widely used in Linked Data to say that two URIs identify the same real-world thing.*

6.2.2 Description Logic (DL)

Description logics are a family of decidable fragments of first-order logic designed for representing structured knowledge. They are the formal foundation of OWL. A DL knowledge base has two parts:

- **TBox (terminological box)** — general knowledge about concepts and roles (the schema), e.g. *Student* $\sqsubseteq$ *Person*.
- **ABox (assertional box)** — facts about individuals, e.g. *Student*(*Ram*), *enrolledIn*(*Ram*, *BCA*).

Building block	Notation	Meaning
Concept (class)	<i>Student</i>	Set of individuals

Building block	Notation	Meaning
Role (property)	enrolledIn	Binary relation between individuals
Individual	Ram	A named object
Intersection	$C \sqcap D$	Both C and D
Union	$C \sqcup D$	C or D
Complement	$\neg C$	Everything not in C
Existential restriction	$\exists R.C$	Has at least one R-link to a C
Universal restriction	$\forall R.C$	All R-links go to C
Subsumption	$C \sqsubseteq D$	C is a subclass of D
Equivalence	$C \equiv D$	Same class

Example 6.2 — DL axioms

$BCAStudent \equiv Student \sqcap \exists enrolledIn.BCAProgramme$

$Student \sqsubseteq Person$

$Teacher \sqcap Student \sqsubseteq \perp$ (nothing is both a teacher and a student — disjointness)

Meaning

A BCA student is exactly a student who is enrolled in some BCA programme; every student is a person; teacher and student classes are disjoint.

Reasoning Services

- **Satisfiability** — can the concept have any instance?
- **Subsumption** — is C always a subclass of D? (classification of the hierarchy)
- **Consistency** — does the knowledge base contain a contradiction?
- **Instance checking** — does an individual belong to a concept?

Note: Trade-off: more expressive DLs give more modelling power but higher computational cost. $SHOIN(D)$ underlies OWL DL, and $SROIQ(D)$ underlies OWL 2.

6.2.3 Web Ontology Language (OWL)

OWL is the W3C standard language for authoring ontologies on the Semantic Web. It builds on RDF/RDFS and is based on description logic. It supports far richer constraints than RDFS and allows machine reasoning.

Species	Description
OWL Lite	Simple class hierarchy and simple constraints; easiest to implement.
OWL DL	Maximum expressiveness while keeping reasoning complete and decidable (maps to DL).
OWL Full	Maximum freedom and compatibility with RDF, but no guarantee of decidability.
OWL 2 (profiles EL, QL, RL)	Revised version with more features and efficient profiles for different needs.

Main OWL Constructs

Construct	Purpose
owl:Class, rdfs:subClassOf	Define classes and hierarchy
owl:ObjectProperty / owl:DatatypeProperty	Relation between individuals / relation to a literal
owl:equivalentClass, owl:sameAs	Equivalence of classes / individuals
owl:disjointWith	Classes with no common members
owl:inverseOf, owl:TransitiveProperty, owl:SymmetricProperty	Property characteristics (e.g. hasChild / hasParent)
owl:someValuesFrom, owl:allValuesFrom	Existential and universal restrictions
owl:minCardinality, owl:maxCardinality, owl:cardinality	Number restrictions
owl:unionOf, owl:intersectionOf, owl:complementOf	Boolean combinations of classes

Example 6.3 — OWL in Turtle

```

ex:Student a owl:Class ; rdfs:subClassOf ex:Person .
ex:Teacher a owl:Class ; rdfs:subClassOf ex:Person ; owl:disjointWith ex:Student .
ex:enrolledIn a owl:ObjectProperty ; rdfs:domain ex:Student ; rdfs:range ex:Course .

```

Inference

If `ex:Ram` `ex:enrolledIn` `ex:CACS458` is asserted, a reasoner infers that Ram is a Student (by domain) and a Person (by subclass), and CACS458 is a Course (by range). If Ram were also declared a Teacher, the reasoner would report an inconsistency because of `owl:disjointWith`.

Basis	RDFS	OWL
Expressiveness	Basic: subclass, domain, range	Rich: cardinality, disjointness, restrictions, characteristics
Reasoning	Simple inference	Powerful DL reasoning
Foundation	RDF	RDF + description logic

Note: Open World Assumption: OWL treats missing information as unknown, not false. Unlike a database, OWL reasoners do not assume that unlisted facts are untrue.

6.3 Social Web and Semantics

The **Social Web** (Web 2.0) is the part of the Web in which users create and share content and interact: social networks, blogs, wikis, microblogs and media-sharing sites. It produces huge amounts of user-generated knowledge, but it is mostly unstructured. **Combining the Social Web with semantics** gives the *Social Semantic Web*.

Social Web Knowledge Resources

- **Tagging and folksonomies** — users label content with free tags (e.g. on Flickr, Delicious). The resulting *folksonomy* is flexible but ambiguous and has no formal structure.
- **Wikis** (Wikipedia) and **DBpedia/Wikidata** — structured data extracted from collaborative content.
- **Social networks** (Facebook, LinkedIn) — graphs of people and relations.
- **Blogs, microblogs, reviews** — opinions and experiences.

Semantic Technologies for the Social Web

- **FOAF (Friend of a Friend)** — RDF vocabulary to describe people, their profiles and relationships.
- **SIOC (Semantically-Interlinked Online Communities)** — describes forums, posts, users and sites.
- **SKOS** — models thesauri and tag vocabularies.
- **Open Graph / schema.org / microformats** — markup that lets pages be understood by machines.
- **Ontologizing folksonomies** — mapping free tags to ontology concepts to remove ambiguity (e.g. ‘java’ = island or language).
- **Knowledge graphs** — Google Knowledge Graph and similar graphs link entities extracted from the Web.

Social Web	Semantic Web	Social Semantic Web
Human-generated content	Machine-processable data	Combines both strengths
Tags, wikis, networks	RDF, OWL, ontologies	FOAF, SIOC, DBpedia, tag ontologies
Flexible but ambiguous	Precise but needs expertise	Community content with shared meaning

Applications and Challenges

- Better search and recommendation using user interests expressed in FOAF.
- Expert finding, community detection and trust/reputation analysis.
- Sentiment and opinion mining from semantically annotated posts.
- **Challenges:** privacy of personal data, quality and trustworthiness of user content, scalability, multi-language vocabularies and spam.

Example 6.4 — FOAF description

ex:Ram a foaf:Person ; foaf:name "Ram Sharma" ; foaf:knows ex:Sita ; foaf:interest
ex:KnowledgeEngineering .

Meaning

Ram is a person named Ram Sharma who knows Sita and is interested in Knowledge Engineering. Many such files link together into a machine-readable social network.

Key Takeaways — Unit 6

- ✓ Knowledge sharing needs information distribution and integration; ontologies help resolve semantic heterogeneity between sources.
- ✓ The Semantic Web adds machine-understandable meaning to the Web using URIs, RDF, RDFS, OWL and SPARQL.
- ✓ RDF represents data as subject–predicate–object triples; Linked Data publishes interlinked RDF using HTTP URIs.
- ✓ Description logic (TBox + ABox) is the formal basis of OWL; OWL offers rich constructs and reasoning (Lite, DL, Full, OWL 2).
- ✓ The Social Semantic Web combines user-generated content (tags, wikis, networks) with semantics (FOAF, SIOC, SKOS, knowledge graphs).

— *End of Unit 6* —

bcalink.com