Semantic Web

Carlo Meghini

Istituto di Scienza e Tecnologie della Informazione
Consiglio Nazionale delle Ricerche – Pisa

2017-18

Today's Lecture: Introduction

The Semantic Web vision

The idea of the SW was described by the inventor of the web, Tim Berners Lee, and two other persons in a seminal paper that appeared in the year 2000.

At that time, the web was still in its infancy, yet several important elements of the SW had appeared on the scene, such as URIs, XML and RDF.

During the following 17 years the idea has turned into a reality that includes languages, software tools and systems, data, people and an entire market sector that is part of the *data economy.*

The development of the SW is promoted mainly by the World Wide Web Committee (W3C), an organization funded by the major software industries and gevernmental agencies (included CNR).

The idea of the SW was conveyed through a visionary example.

Lucy needs to see a specialist and then has to have a series of physical therapy sessions.

Lucy instructs her SW agent to make a plan for her. The agent:

1. Retrieves information about Lucy's prescribed treatment from the doctor's agent.
2. Looks up several lists of providers, checking for the ones in-plan for Lucy's insurance within a 20-mile radius of her home and with a rating of excellent or very good on trusted rating services.
3. Tries to find a match between available appointment times (supplied by the agents of individual providers through their Web sites) and Lucy's busy schedules.
4. In a few minutes the agent presents her with a plan.
5. Lucy does not like the plan: it includes an Hospital on a busy street.
6. She sets her agent to redo the search with stricter preferences about location.

7. Almost instantly the new plan was presented: a much closer clinic and earlier times, but there were two warning notes.
   - First, Lucy would have to reschedule a couple of his less important appointments. She checked what they were – not a problem.
   - The other was something about the insurance company's list failing to include this provider under physical therapists: "Service type and insurance plan status securely verified by other means," the agent reassured her. "(Details?)"

Lucy could use her agent to carry out all these tasks thanks not to the World Wide Web of today but rather the SW that it will evolve into.

Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully.

Computers can adeptly parse Web pages for layout and routine processing but in general, computers have no reliable way to process the *semantics:* to recognize that a link points to the home page of a medical doctor, that another link points to the doctor's *curriculum vitae,* and so on.

Lucy's scenario is not science fiction. It does not need revolutionary scientific progress to be achieved.

To get started, partial solutions to all important parts of the problem exist.

What are the ingredients that are necessary to make the Lucy's scenario work?

1. Formal descriptions
2. Ontologies
3. Logic
4. Agents

Formal knowledge

For the SW to function, computers must have access to collections of *formal statements* that provide *knowledge* about the relevant entities.

Web pages contain much knowledge, encoded in natural languages (text, pictures, graphics, videos) that humans can *interpret* to extract the desired knowledge. But computers are still not good enough at doing this interpretation.

The SW approach to solving the problem is not the development of superintelligent agents. Instead it proposes to replace HTML by formal languages for encoding knowledge, languages that machine can interpret and properly use.

Example

VitaSana offers Physiotherapy .	t1 OnDay Monday .
JohnPastor works-for VitaSana .	t1 At-from 10AM .
JohnPastor receives-on t1 .	t1 At-to 11AM .

This is a formal description of the VitaSana company, including the working hours of a therapist.

A machine can use this description to match the working hours of the therapist with the woking hours of Lucy and determine whether Lucy can be treated by the therapist.

Similarly to most web pages, such description:

- can be automatically generated by a program that reads the relevant data from a database
- can be automatically published on the web like any other page
- can be automatically retrieved via the HTTP protocol by an agent that requests the VitaSana web page to the appropriate HTTP server

This is all happening today thanks to Linked Data.

Ontologies

Formal descriptions are not enough: how does an agent know that

- works-for relates an employee to the company the employee works for?
- receives-on relates an employee to their working hours?
- 10AM is a time of the day?

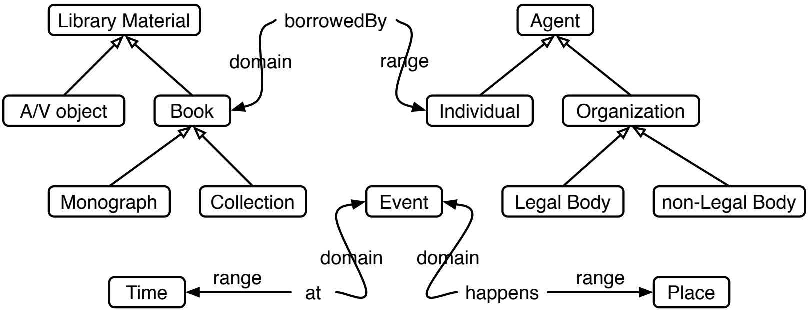
There has to be a *formal vocabulary* that defines the terms used in formal descriptions.

Such formal vocabularies are called *formal ontologies.*

In philosophy, an ontology is a theory about the nature of existence, of what types of things exist; ontology as a discipline studies such theories.

In computer science, an ontology is an explicit and formal specification of the terms of a domain, giving the relations among such terms.

For example:



Library Material
borrowedBy
Agent
domain
range
A/V object
Book
Individual
Organization
Monograph
Collection
Event
Legal Body
non-Legal Body
domain
domain
Time
range
at
happens
range
Place

Ontologies provide a shared vocabulary for describing the entities in a domain, so as to overcome differences in terminology.

- One application's zip code may be the same as another application's area code.
- Two applications may use the same term with different meanings. In university A, a course may refer to a degree (like computer science), while in university B it may mean a single subject (CS 101).

Such differences can be overcome by mapping the particular terminology to a shared ontology, thereby achieving *semantic interoperability.*

In Artificial Intelligence there is a long tradition of developing and using ontology languages, in the field of Knowledge Representation.

This is a foundation SW is built upon.

Logic

Logic is the discipline that studies the principles of reasoning.

It offers:

- formal languages for expressing knowledge, be it formal descriptions or formal vocabularies.
- well-understood formal semantics: in most logics, the meaning of sentences is defined without the need to operationalize the knowledge (declarative knowledge)
- an inferential relation allowing automated reasoners to deduce (infer) conclusions from the given knowledge, thus making implicit knowledge explicit.

Suppose we know that all monographs are books, that all books are library material, and that Wittgenstein's *Tractatus Logico-Philosophicus* is a monograph. In predicate logic the information is expressed as follows:

Example

$(\forall x) Monograph(x) \rightarrow Book(x)$
$(\forall x) Book(x) \rightarrow LibMat(x)$
$Monograph(tlp)$

Then we can infer:

Example

$Book(tlp)$
$LibMat(tlp)$
$(\forall x) Monograph(x) \rightarrow LibMat(x)$

Logic can then be used as a representation language for formal descriptions and for formal ontologies, with several advantages:

- it allows agents to use descriptions and ontologies to infer implicit knowledge, for instance for matching descriptions, without requiring any other software that the inference engine of the logic
- it can be used to extract knowledge from descriptions and ontologies by providing answers to queries
  - queries are open formulas (predicates) and answers are sets of individuals that satisfy the query in every interpretation
- it can be used to test the quality of descriptions and ontologies, such as consistency
- it can provide explanations for conclusions.

But logic must be used with care: there is a trade-off between expressive power and efficiency.

Logics that allow to express incomplete knowledge, via

- disjunction
- existential quantification
- negation

tend to be computationally expensive. And drawing certain conclusions may become impossible if noncomputability barriers are encountered.

The SW is based in a family of logics, called *Description Logics,* that are *decidable* contractions of the predicate calculus.

So far, it seems that these logics are expressive enough to capture the knowledge relevant to the SW.

Agents

Agents are pieces of software that work autonomously and proactively.

Conceptually they evolved out of the concepts of object-oriented programming and component-based software development.

A personal agent on the SW can receive some tasks and preferences, and can execute its task by performing various activities: retrieving information from the web, communicating with other agents, comparing information about user requirements and preferences, selecting certain choices, and giving answers to the user.

An example of such an agent is Lucy's private agent.

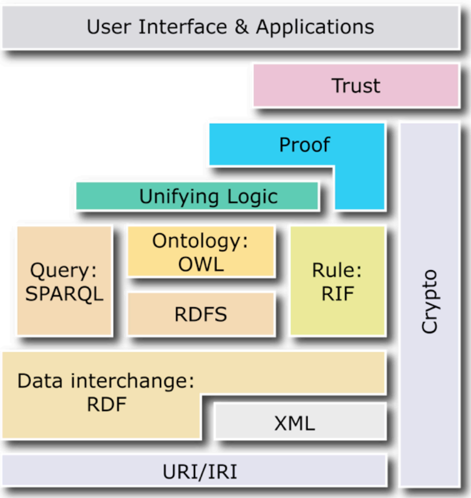
Agents will not replace human users on the SW, nor will necessarily make complex decisions.

In many, if not most, cases their role will be to collect and organize information, and present choices for the users to select from, as Lucy's personal agent did.

Agents on the SW can accomplish their tasks by relying on the technologies that we have seen so far: formal descriptions and vocabularies, and logic.

The Semantic Web Stack

The development of the Semantic Web proceeds in steps, each step building a *layer* on top of another.



Contents of the course

The course is divided into four main parts:

1. The first part is this introduction
2. The second part is devoted to RDF
3. The third part is devoted RDFS
4. the last part is devoted to OWL.

And conclusions will come at the end.

Useful Readings

- T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American 284 (May 2001): 34-43.
- Grigoris Antoniou and Frank van Harmelen. A semantic Web primer. The MIT Press, 2004. [ch. 1]