General Procedural Knowledge for the Semantic Web
Paolo Pareti

THE UNIVERSITY OF EDINBURGH

A lot of unstructured know-how
Have you ever wondered how to do something and found the answer on the Web?

Popular websites containing know-how:

eHow™
> 1.000.000 know-how articles

wikiHow
The world's how to manual.
> 150.000 know-how articles

Recipes, step-by-step instructions and do-it-yourself guides are among the most common examples of know-how (or Procedural Knowledge).

Their format can vary, but they are often described using unstructured text, pictures or video.

Some of them are publicly available on the Web while others are private (e.g. the standard operating procedures of an organisation).

Semantic Representation
Connecting unstructured know-how resources to a machine understandable representation

Web Resources
How to make pancakes:
1. Prepare the mix
2. Pour the mix in a hot pan
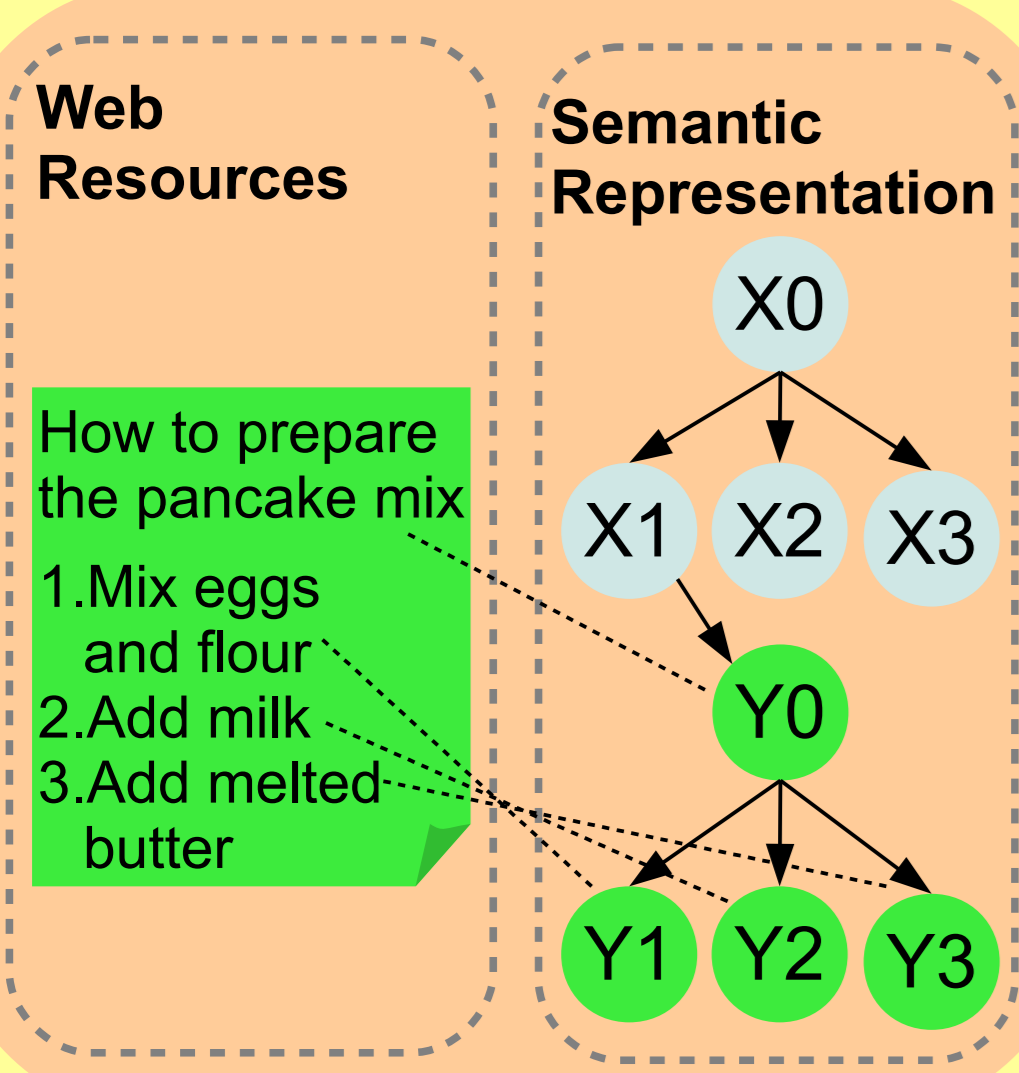3. Cook until golden
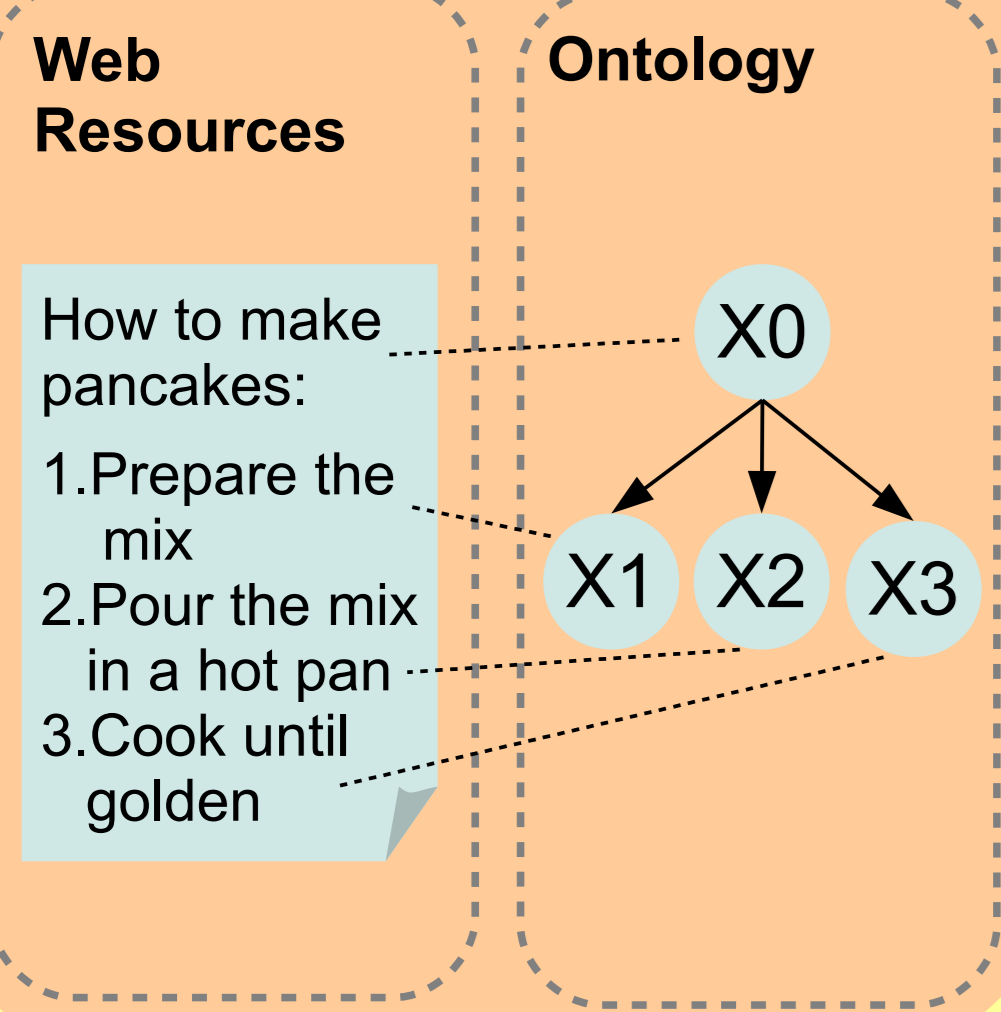
Ontology
X0
X1 X2 X3

The resources describing a process (e.g. the text where the instructions are written) can be connected to the ontological entities that represent them.

In this example, a web page containing the instructions on how to make pancakes is annotated semantically. The text fragment "Prepare the mix" is connected to the ontological entity X1, which is a sub-step of the entity X0 ("make pancakes").

Semantic Connections
Making the relation between different processes explicit

Web Resources
How to prepare the pancake mix
1. Mix eggs and flour
2. Add milk
3. Add melted butter

Semantic Representation
X0
X1 X2 X3
Y0
Y1 Y2 Y3

How can you prepare the pancake mix? In this example the entity Y0 ("prepare the pancake mix") is connected to the entity X1 ("Prepare the mix") from the previous example.

This makes it explicit that the three steps Y1, Y2 and Y3 can be performed not only to achieve the process Y0, but also the process X1 described in the other Web page.

How can programs understand Procedural Knowledge?

Knowledge can be machine-understandable if it is represented semantically using a formal representation of a conceptualization (an ontology). An ontology defines the meaning of a set of concepts (e.g. an *Action* or a *Condition*) and a set of relations (e.g. the sub-class relation).

A Procedural Ontology can describe concepts such as actions, inputs, outputs, sub-steps and workflow operators (e.g. the if-then-else construct, loops etc...).

The Semantic Web is a Web of machine understandable resources built on top of the existing Web infrastructure. Among other things, it defines languages to represent ontologies (e.g. RDFS and OWL) and ways to interact with them (e.g. the SPARQL query language and the OWL reasoners).

Automate
If a process is described with sufficient level of detail, a system might be able to automate it (or to automate some of its steps). For example it could be automated by a Web Service or maybe even by a robotic agent.

Enhance
A system that understands know-how could recognize what activity you are trying to achieve (Activity Recognition) and assist you in completing the remaining steps. If you are trying to reach a certain destination, for example, it could give you the directions on how to reach it.

Search
It will be easier to find know-how as an intelligent system could use advanced filters in the search. For example excluding processes that require a tool you don't have or that would take too much time according to your preferences.

Why should general know-how (Procedural Knowledge) be machine understandable?

Manual and Automatic annotation
How to get from unstructured resources to a formal ontological model

Unstructured resources describing know-how

manual or automatic annotation
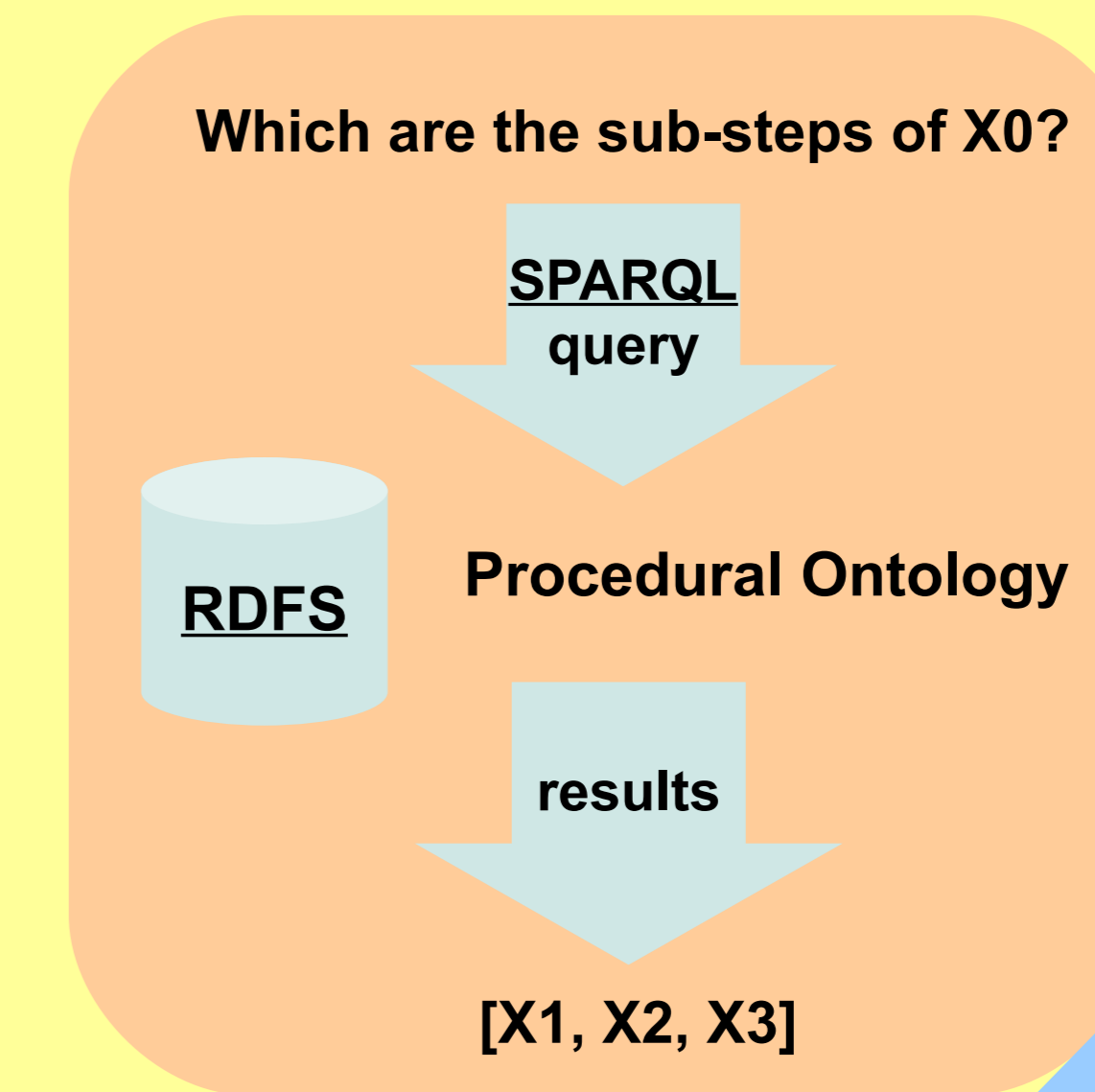
Procedural Ontology
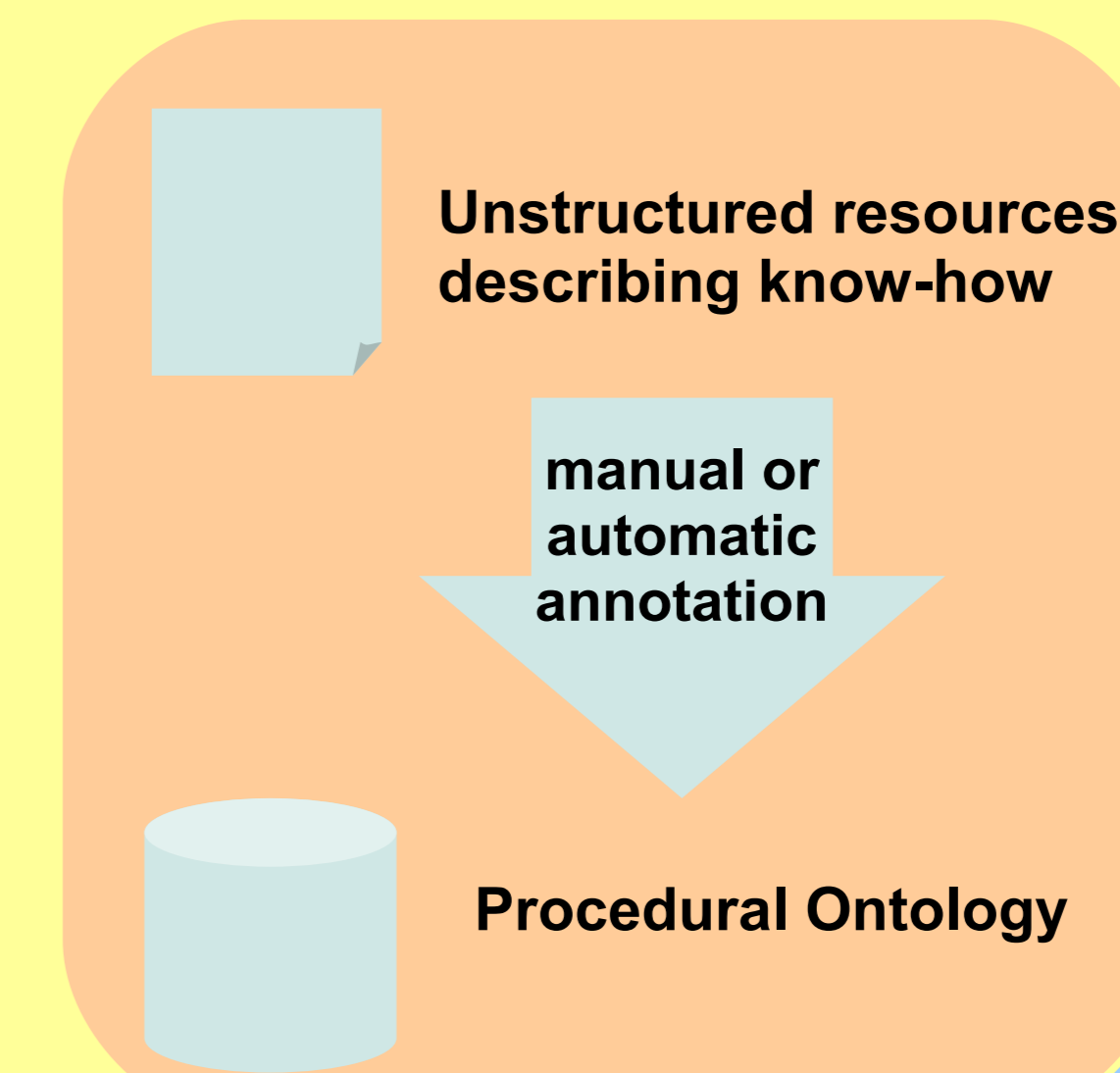
I propose to use both manual and automatic annotation techniques.
• **Manual annotation**
Using an annotation tool, users can annotate procedures manually. In order to overcome the scalability limitations of this approach, Social Computation strategies could be used.
• **Automatic annotation**
Several attempts have been made to extract procedural knowledge automatically, usually by means of NLP. These approaches however suffer from limitations in accuracy.

RDFS and SPARQL
A lightweight ontological framework built on top of Semantic Web technologies

Which are the sub-steps of X0?

SPARQL query

RDFS → **Procedural Ontology**

results

[X1, X2, X3]

To make its adoption easier, this framework will be built on top of Semantic Web standards.
• **RDFS**
The Procedural Ontology will be encoded in RDFS, a simple language to define vocabularies of terms.
• **SPARQL**
The SPARQL query language will be used to interact with the procedural knowledge bases. Software clients (e.g. javascript functions) will be able to query an RDFS knowledge base to find out, for example, all the known sub-steps of a procedure.

A general Procedural Ontology

A0
C1 alternative E1
precondition has effect
I1 has input X0 has output O1
sub-steps
X1 X2 X3
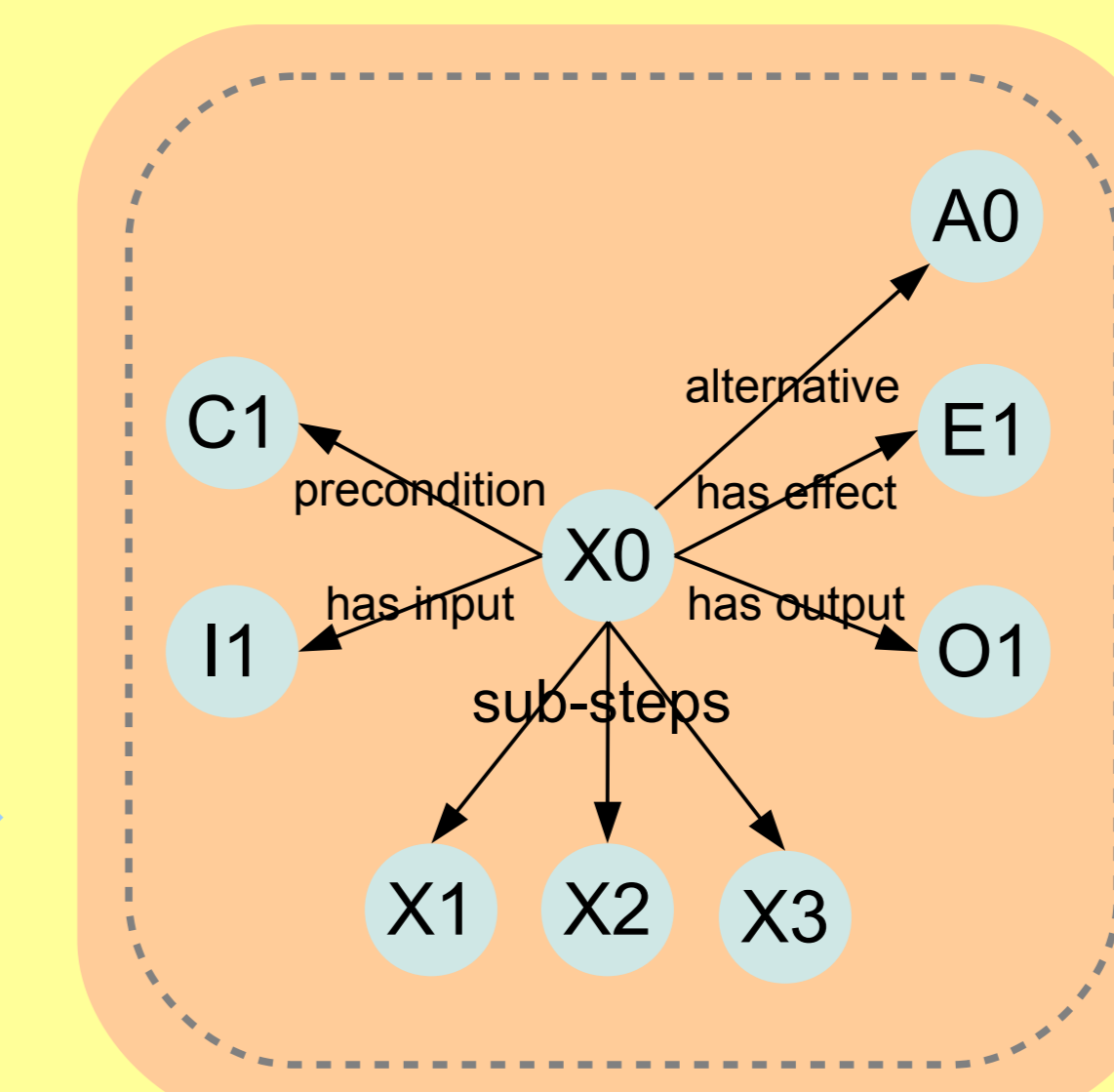
The Procedural Ontology I propose allows procedures to be described along several dimensions:

• Sub-steps allow a procedure to be seen at different levels of granularity
• The objects involved in the procedure (e.g. inputs and outputs)
• The conditions that can be true or false (e.g. preconditions and effects)
• Alternative processes that achieve the same task
• etc...