Ontologies & Web Service Annotation

The lowdown ... or low-down?
Ontologies & Web Service Annotation

The lowdown ... or low-down?
Ontologies & Software Annotation

The lowdown ... or low-down ?
Ontologies & Software Annotation

lowdown - “The fundamental, though not generally known, facts on (about) a person, situation, etc.; the ‘inside story’.”

low-down - “degraded, abject.”
Ontologies & Software Annotation
Software Description
Software Documentation
“Sooner or later someone (or something) else will have to understand the programs you write.”

“A very high percentage of production programs are used and maintained by someone other than the original author. Poor documentation inflicts a vicious, never ending circle of crises.”

“There are two familiar variants of the documentation problem:
   a) you write all the docs that you know you need and you pay a terrible price for it, or
   b) you don't write all the docs you need and you pay a terrible price for that.”

“Many programmers enjoy writing documentation if the documentation standards aren't unreasonable.”

“Software engineers create and use simple yet powerful documentation, and tend to ignore complex and time-consuming documentation.”

“Good documentation is a sign of the professional pride a programmer puts into a program.”

“Knowing that a program can be understood and amended by someone else ought to be one of the programmer's criteria for success. “

“Programs must be written for people as well as computers.”
Application Software Documentation

not ...

system software
programming software
Application Software

- Inputs
- Application
- Outputs
- Other parameters
Applications

Operation 1
- Inputs
- Other Parameters
- Operation 1
- Outputs
- Application

Operation 2
- Inputs
- Other Parameters
- Operation 2
- Outputs
Applications

Inputs -> Operations -> Outputs

Other parameters
Application Software

- **Inputs**
  - Type
  - Format

- **Operations**
  - What is done?
  - How (algorithm)?

- **Outputs**
  - Type
  - Format

- **Other parameters**
  - Type
  - Format
Application interfaces

- Desktop GUI
- Application
- Web application
- Command-line UI
- Web service
- Scripts, workflows etc.
Application Description

Application description =

- Function
  - What it does (operations)
  - How it does it (algorithms - if needed)
- Inputs, outputs and other parameters
  - Basic types of data
  - Data formats
- How to invoke it (interface)
- Other things (depending on scenario)
Scientists must handle an increasingly large, complex and diverse set of software (tools and data).

They demand increasingly better ways to handle resources in everyday tasks:

- Organise
- Find
- Compare
- Select
- Use
- Connect

This all requires descriptions of tools and data that are:

- Scientifically relevant
- Consistent
- Comprehensive
- Machine-understandable
- Searchable

This in turn requires

- Controlled vocabulary
- Documentation / annotation / curation effort
- Resource registry / catalogue / collections
Use cases boil down to:
- Finding stuff
  - "Service discovery"
- Using / connecting stuff
  - "Service (inter)operability"

Use cases have different documentation / annotation requirements.

Exact requirements depend on scenario.

Service (inter)operability requires more!
- must describe how service can be used (e.g. WSDL)

In any case:
- Document your software!
- Use ontologies/controlled vocabularies
- Use standards, e.g. SAWSDL for WSDL file (web service) annotations