Software Ontologies and Metadata Schemes References
Morane Gruenpeter
September 2017

A list for collecting and organizing software ontologies and metadata schemes describing software components and software source code.

1 Preliminary references

1. The term ontology [1]
3. Software citation principles [3]

2 dedicated software ontologies

1. DOAP- Description Of A Project[4]
2. SEON - Software Evolution ONtologies [5]

3 scholarly ecosystem

1. OntoSoft for scientific software[7]
2. DataCite https://schema.datacite.org/
3. TOTEM - the Trustworthy Online Technical Environment Metadata Registry http://www.keep-totem.co.uk/
4 linked data

1. Wikidata [8]
   - Software: https://www.wikidata.org/wiki/Q7397

2. schema.org:
   - SoftwareApplication: http://schema.org/SoftwareApplication

5 digital preservation schemes

1. Dublin Core (DCMI): http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=terms#dcmitype-Software

2. PRONOM -


6 package management vocabularies

Found mostly in the source code itself as manifest files. The name of the file is defined below by the context or development ecosystem in which it is used:
Table 1: manifest files for package management

<table>
<thead>
<tr>
<th>context</th>
<th>filename</th>
<th>in CodeMeta</th>
</tr>
</thead>
<tbody>
<tr>
<td>java- Maven</td>
<td>pom.xml</td>
<td>yes</td>
</tr>
<tr>
<td>Octave</td>
<td>DESCRIPTION</td>
<td>yes</td>
</tr>
<tr>
<td>R package</td>
<td>DESCRIPTION</td>
<td>yes</td>
</tr>
<tr>
<td>ruby gems</td>
<td>.gemspec or Rakefile</td>
<td>yes</td>
</tr>
<tr>
<td>Javascript npm</td>
<td>package.json</td>
<td>yes</td>
</tr>
<tr>
<td>Perl CPAN::META</td>
<td>META.json, META.yml, .spec</td>
<td>yes</td>
</tr>
<tr>
<td>Dart</td>
<td>subspec.yaml</td>
<td>no</td>
</tr>
<tr>
<td>Debian package</td>
<td>debian/upstream/metadata</td>
<td>yes</td>
</tr>
<tr>
<td>puppet</td>
<td>metadata.json</td>
<td>no</td>
</tr>
<tr>
<td>PyPI</td>
<td>setup.py</td>
<td>yes</td>
</tr>
<tr>
<td>Scientific software</td>
<td>CITATION</td>
<td>no</td>
</tr>
<tr>
<td>CodeMeta</td>
<td>CODE, code.json, codemeta.json</td>
<td>yes</td>
</tr>
<tr>
<td>Java gradle</td>
<td>gradle.properties</td>
<td>no</td>
</tr>
<tr>
<td>Jekyll</td>
<td>_config.yml -no</td>
<td>no</td>
</tr>
<tr>
<td>clojure</td>
<td>project.clj or build.boot</td>
<td>no</td>
</tr>
<tr>
<td>haskell</td>
<td>project.cabal</td>
<td>no</td>
</tr>
<tr>
<td>scala</td>
<td>build.sbt</td>
<td>no</td>
</tr>
<tr>
<td>Ocaml</td>
<td>opam</td>
<td>no</td>
</tr>
</tbody>
</table>

A list of the most popular terms used in the package management domain:

```python
code_management_popular_terms = {
    "developmentStatus": [],
    "version": [],
    "operatingSystem": [],
    "description": [],
    "keywords": [],
    "issueTracker": [],
    "name": [],
    "author": [],
    "relatedLink": [],
    "url": [],
    "type": [],
    "license": [],
    "maintainer": [],
    "email": [],
```

3
"softwareRequirements": [],
"identifier": [],
"codeRepository": []
}

References


