Keeping modular and platform-independent software up-to-date: benefits from the Semantic Web

Olivier Dameron

SMI - Stanford University

8th International Protégé Conference – July 18-21, 2005

Problem

- Keeping local installation of Protégé and related software up-to-date
  - Protégé (1/week)
  - Plugins (1/day)
  - RacerPro, Jess
- Potential roll-back to a previous version
- ... is a tedious task, even for one single machine !

Layout

- Requirement analysis
  - comparison of several approaches
  - a RDF-based approach is necessary
  - existing format (DOAP) needs to be extended
- Solution proposed
  - general principles
  - adaptation to Protégé

Requirements

- Automatic
  - retrieval of available version (is there a new one?)
  - download and install if necessary
- Efficient (avoid unnecessary network traffic)
- Installation should be clean and customizable
  - destination directory
  - roll-back (at least manually)
  - local config (DB drivers, link to local ontologies...)
- Platform-independent (like Protégé)
- Extensible

Principle (Methods)

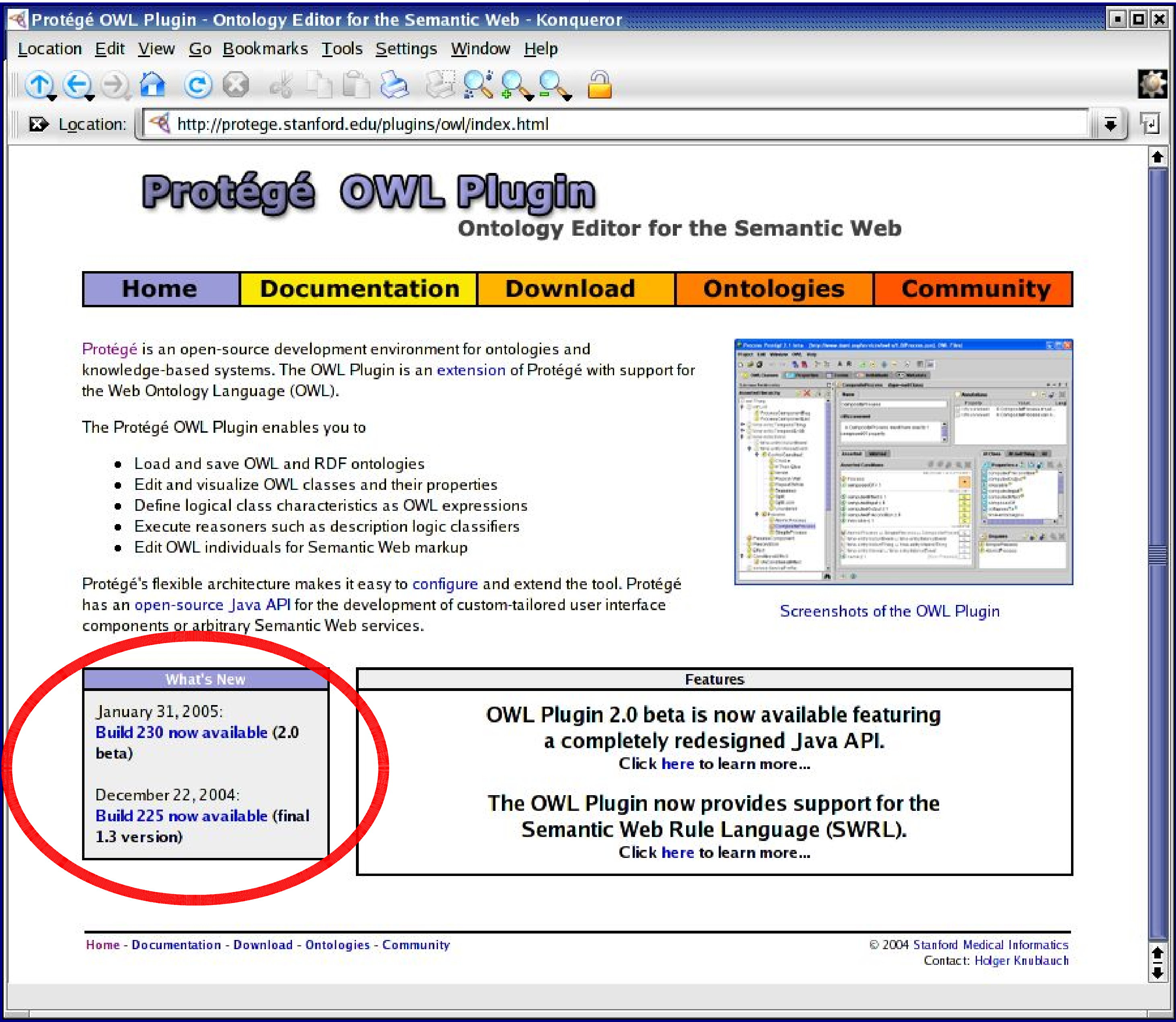
- For each software item (Protégé, plugin, reasoner…):
  - Find the current available version
  - Compare with local version
  - If necessary, update (without messing the previous versions)
- Apply local customization

Principle (Methods)

- For each software item (Protégé, plugin, Racer...):
  - **Find the current available version**
  - Compare with local version
  - If necessary, update (without messing the previous version)
- Apply local customization

Difficult!

Finding the latest version number: the dirty way


Protégé OWL Plugin - Ontology Editor for the Semantic Web - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://protege.stanford.edu/plugins/owl/index.html
Protégé OWL Plugin
Ontology Editor for the Semantic Web
Home
Documentation
Download
Ontologies
Community

Protégé is an open-source development environment for ontologies and knowledge-based systems. The OWL Plugin is an extension of Protégé with support for the Web Ontology Language (OWL).

The Protégé OWL Plugin enables you to

- Load and save OWL and RDF ontologies
- Edit and visualize OWL classes and their properties
- Define logical class characteristics as OWL expressions
- Execute reasoners such as description logic classifiers
- Edit OWL individuals for Semantic Web markup

Protégé's flexible architecture makes it easy to configure and extend the tool. Protégé has an open-source Java API for the development of custom-tailored user interface components or arbitrary Semantic Web services.

Screenshots of the OWL Plugin

What's New

January 31, 2005:
Build 230 now available (2.0 beta)

December 22, 2004:
Build 225 now available (final 1.3 version)

Features

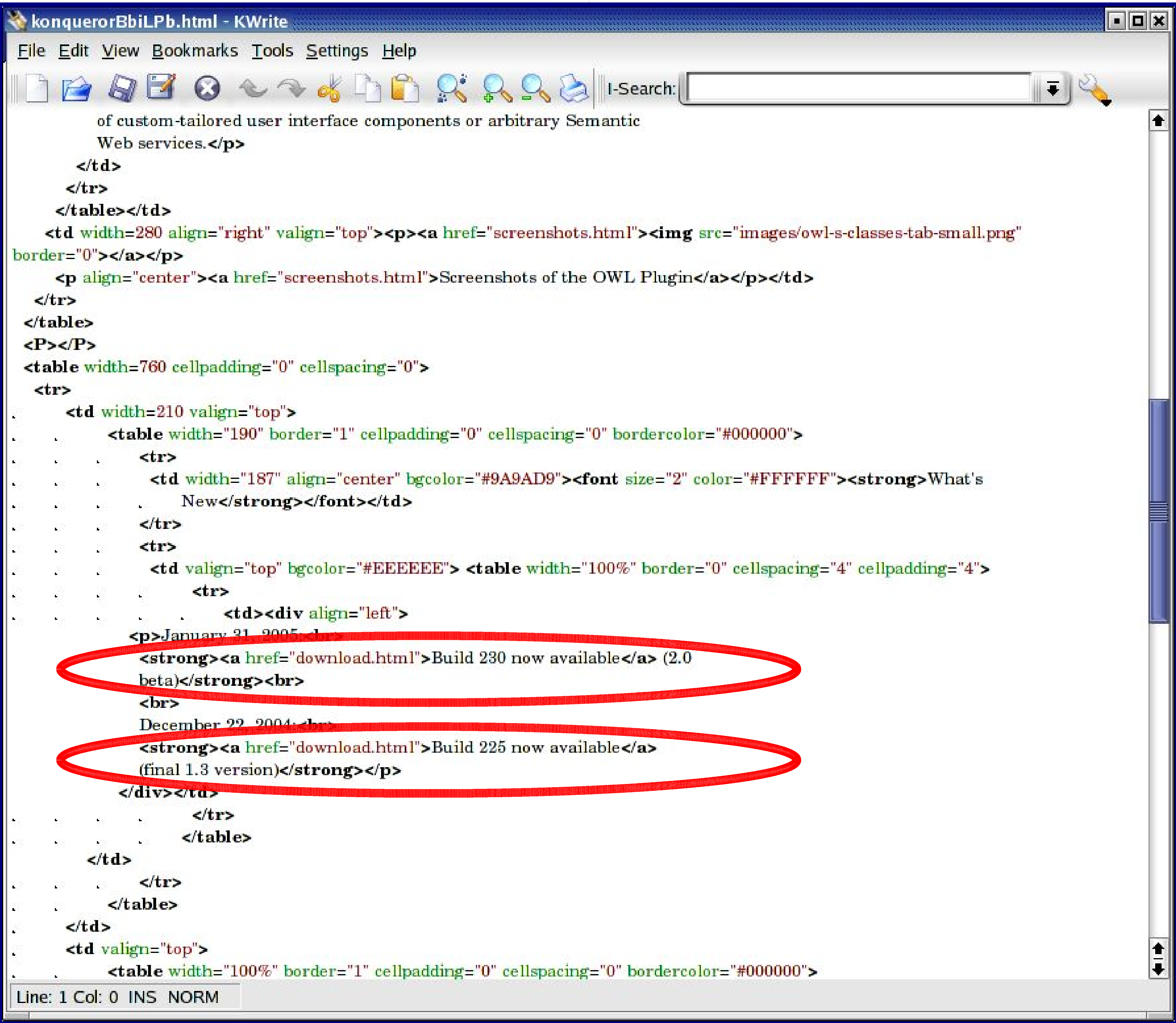
OWL Plugin 2.0 beta is now available featuring a completely redesigned Java API.
Click here to learn more...

The OWL Plugin now provides support for the Semantic Web Rule Language (SWRL).
Click here to learn more...

Home - Documentation - Download - Ontologies - Community

© 2004 Stanford Medical Informatics
Contact: Holger Knublauch

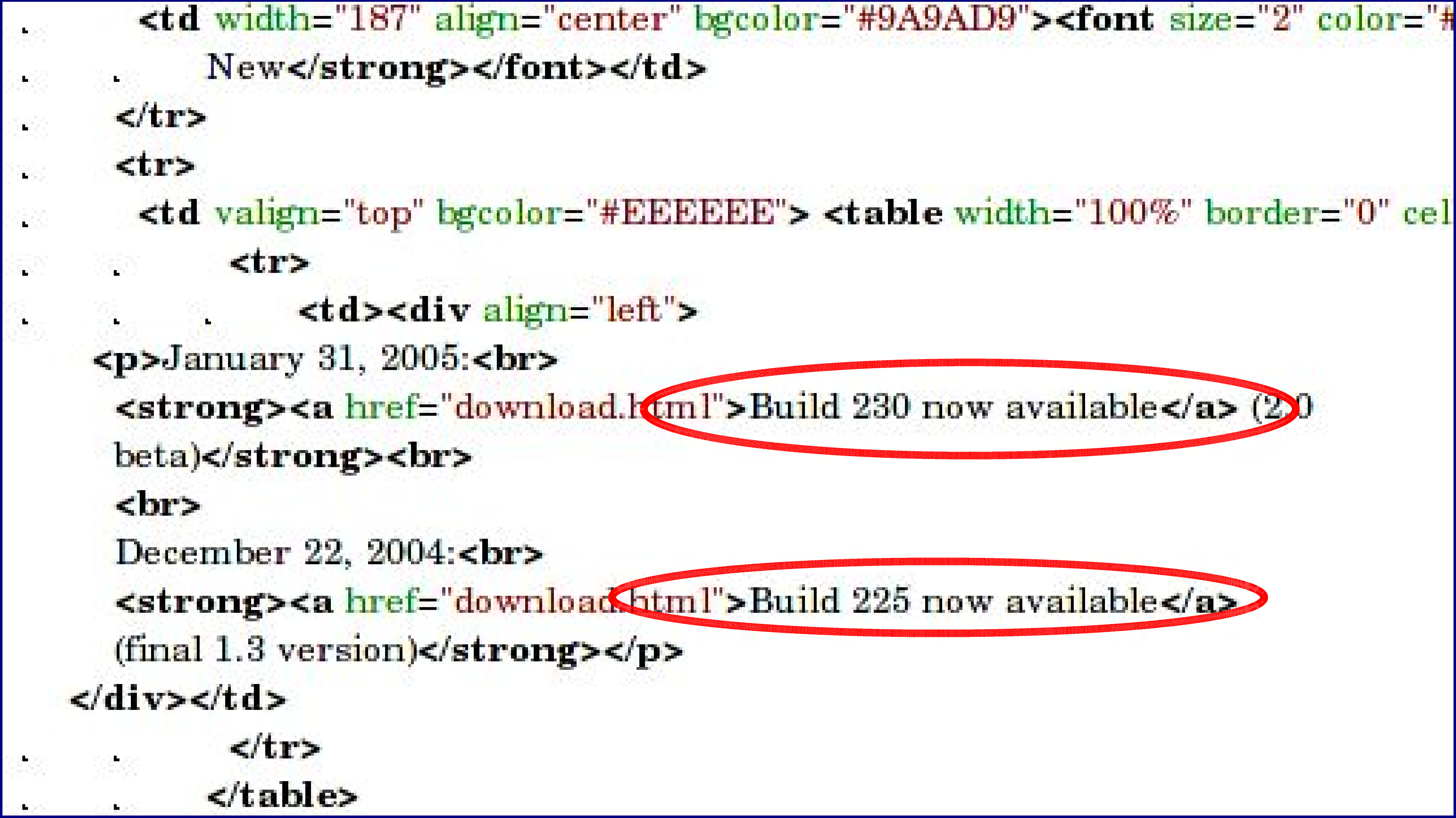
Finding the latest version number: the dirty way



konquerorBbiLPb.html - KWrite
File Edit View Bookmarks Tools Settings Help
I-Search:

```
        of custom-tailored user interface components or arbitrary Semantic
        Web services.</p>
      </td>
     </tr>
    </table></td>
   <td width=280 align="right" valign="top"><p><a href="screenshots.html"><img src="images/owl-s-classes-tab-small.png"
border="0"></a></p>
    <p align="center"><a href="screenshots.html">Screenshots of the OWL Plugin</a></p></td>
  </tr>
 </table>
 <P></P>
 <table width=760 cellpadding="0" cellspacing="0">
  <tr>
     <td width=210 valign="top">
         <table width="190" border="1" cellpadding="0" cellspacing="0" bordercolor="#000000">
            <tr>
             <td width="187" align="center" bgcolor="#9A9AD9"><font size="2" color="#FFFFFF"><strong>What's
                New</strong></font></td>
            </tr>
            <tr>
             <td valign="top" bgcolor="#EEEEEE"> <table width="100%" border="0" cellspacing="4" cellpadding="4">
                  <tr>
                     <td><div align="left">
           <p>January 31, 2005:<br>
            <strong><a href="download.html">Build 230 now available</a> (2.0
            beta)</strong><br>
            <br>
            December 22, 2004:<br>
            <strong><a href="download.html">Build 225 now available</a>
            (final 1.3 version)</strong></p>
          </div></td>
                  </tr>
                 </table>
       </td>
            </tr>
         </table>
     </td>
     <td valign="top">
         <table width="100%" border="1" cellpadding="0" cellspacing="0" bordercolor="#000000">
```

Finding the latest version number: the dirty way



```
.         <td width="187" align="center" bgcolor="#9A9AD9"><font size="2" color="#
.     .        New</strong></font></td>
.       </tr>
.       <tr>
.         <td valign="top" bgcolor="#EEEEEE"> <table width="100%" border="0" cel
.     .          <tr>
.     .      .         <td><div align="left">
     <p>January 31, 2005:<br>
       <strong><a href="download.html">Build 230 now available</a> (2.0
       beta)</strong><br>
       <br>
       December 22, 2004:<br>
       <strong><a href="download.html">Build 225 now available</a>
       (final 1.3 version)</strong></p>
   </div></td>
.     .          </tr>
.     .        </table>
```

Finding the latest version number: the dirty way

Parse the HTML code of the page

- gory grep and regexp manipulations
- requires to find a keyword
- HTML is for humans, not for (smart) applications!
- what if the item developer changes the HTML code ?

Finding the latest version number: the dirty way

Parse the HTML code of the page

- gory grep and regexp manipulations
- requires to find a keyword on the same line
- HTML is for humans, not for (smart) applications!
- what if the item developer changes the HTML code ?

- **This is just the wrong approach**

Finding the latest version number: the dirty way

Parse the HTML code of the page

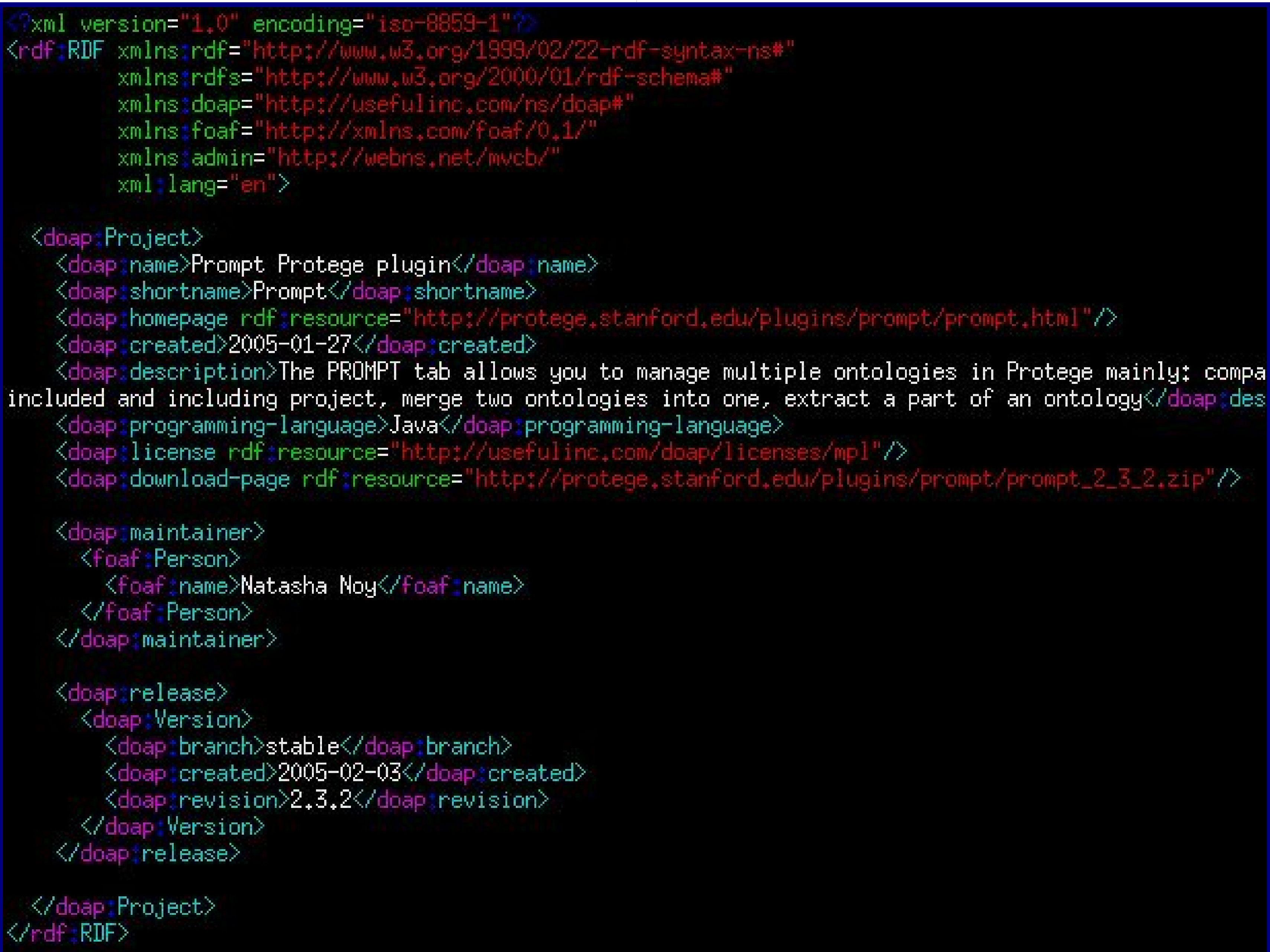
- gory grep and regexp manipulations
- requires to find a keyword on the same line
- HTML is for humans, not for (smart) applications!
- what if the item developer changes the HTML code ?

- This is just the wrong approach
- **Unfortunately, this was the case for most items: Protégé, OWL-plugin, Racer,...**

Finding the latest version number: a somewhat better way

- Use XML descriptions
  - no DTD or schema available
  - XML is OK for a shared understanding of a data structure
- Use RDF description
  - the DOAP project [http://usefulinc.com/doap]
  - using RDF allows to specify the semantics of the project description

Finding the latest version number: a somewhat better way



```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:doap="http://usefulinc.com/ns/doap#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:admin="http://webns.net/mvcb/"
         xml:lang="en">

  <doap:Project>
    <doap:name>Prompt Protege plugin</doap:name>
    <doap:shortname>Prompt</doap:shortname>
    <doap:homepage rdf:resource="http://protege.stanford.edu/plugins/prompt/prompt.html"/>
    <doap:created>2005-01-27</doap:created>
    <doap:description>The PROMPT tab allows you to manage multiple ontologies in Protege mainly: compa
included and including project, merge two ontologies into one, extract a part of an ontology</doap:des
    <doap:programming-language>Java</doap:programming-language>
    <doap:license rdf:resource="http://usefulinc.com/doap/licenses/mpl"/>
    <doap:download-page rdf:resource="http://protege.stanford.edu/plugins/prompt/prompt_2_3_2.zip"/>

    <doap:maintainer>
      <foaf:Person>
        <foaf:name>Natasha Noy</foaf:name>
      </foaf:Person>
    </doap:maintainer>

    <doap:release>
      <doap:Version>
        <doap:branch>stable</doap:branch>
        <doap:created>2005-02-03</doap:created>
        <doap:revision>2.3.2</doap:revision>
      </doap:Version>
    </doap:release>

  </doap:Project>
</rdf:RDF>
```

Finding the latest version number: a somewhat better way

- (+) Version and Download URL can be retrieved from the project's DOAP description
- (+) The DOAP description can be automatically generated
- (+) A DOAP description refers to DOAP RDFS
- (-) DOAP needs to be extended for representing various distributions of a single project (architecture, flavor, JVM,...)
- (-) The DOAP description is parsed syntactically :-(

Why syntactic (i.e. Xpath-like) parsing of DOAP is bad:

-
  ```
  <rdf:Description rdf:about=”checkProtege">
    <rdf:type rdf:resource=”doap:Project” />
    <doap:download-page rdf:resource=”http...” />
  ```

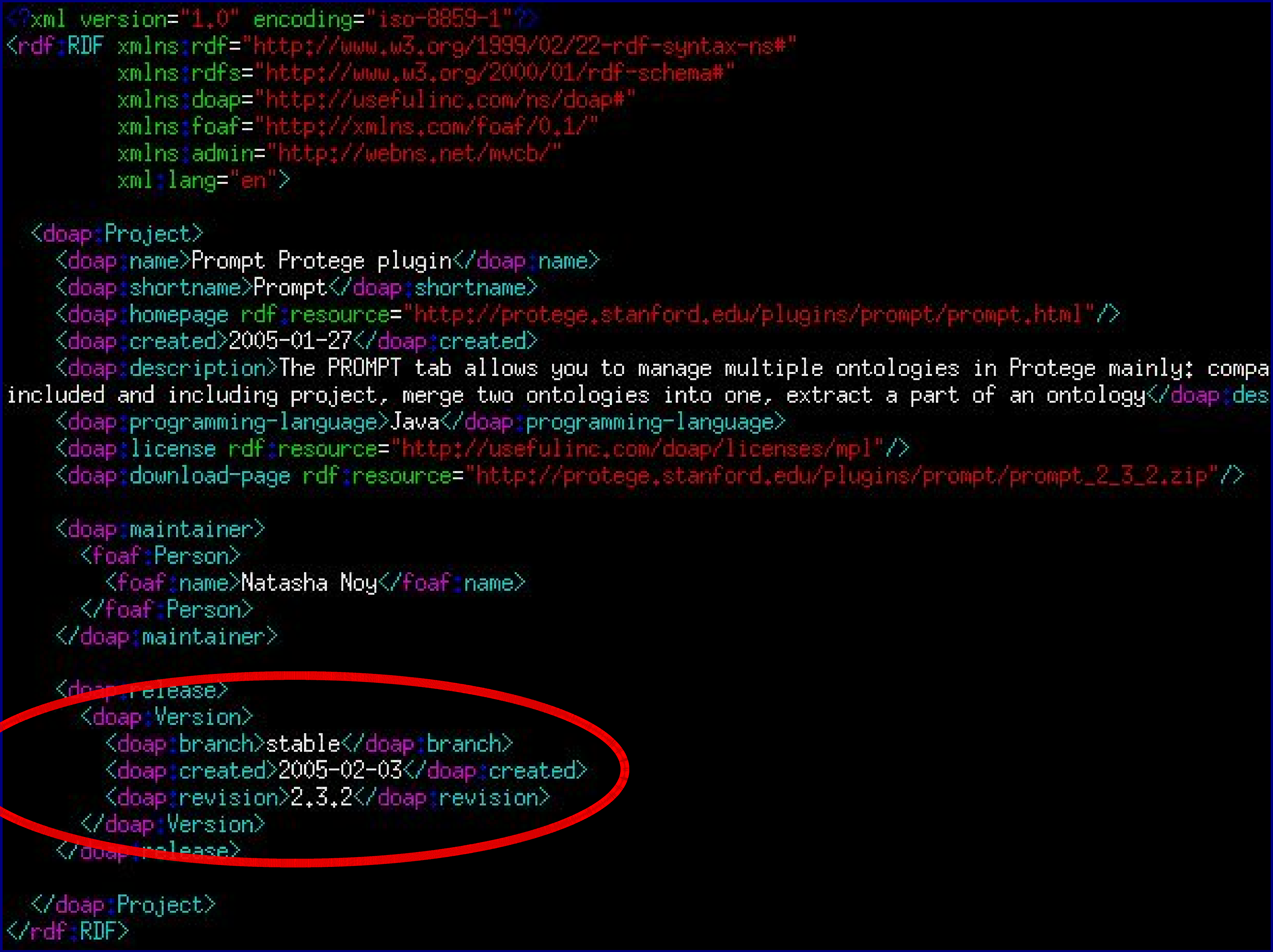
-
  ```
  <doap:Project rdf:about=”checkProtege”>
    <doap:download-page rdf:resource=”http...” />
  ```

-
  ```
  <doap:Project rdf:about=”checkProtege”
    doap:download-page=”http://smi...”/>
  ```

are all valid RDF descriptions representing the same thing

Finding the latest version number: the Semantic Web way

- RDF query of the DOAP descriptions
  - abstract from multiple RDF syntax
  - allow developers to leverage RDFS expressivity
    - specialize classes and relations
    - add new relations (e.g. for multiple download URL of Protégé)
- Implementation choice: Sesame SeRQL (could be SPARQL as well...)



```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:doap="http://usefulinc.com/ns/doap#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:admin="http://webns.net/mvcb/"
         xml:lang="en">

  <doap:Project>
    <doap:name>Prompt Protege plugin</doap:name>
    <doap:shortname>Prompt</doap:shortname>
    <doap:homepage rdf:resource="http://protege.stanford.edu/plugins/prompt/prompt.html"/>
    <doap:created>2005-01-27</doap:created>
    <doap:description>The PROMPT tab allows you to manage multiple ontologies in Protege mainly: compa
included and including project, merge two ontologies into one, extract a part of an ontology</doap:des
    <doap:programming-language>Java</doap:programming-language>
    <doap:license rdf:resource="http://usefulinc.com/doap/licenses/mpl"/>
    <doap:download-page rdf:resource="http://protege.stanford.edu/plugins/prompt/prompt_2_3_2.zip"/>

    <doap:maintainer>
      <foaf:Person>
        <foaf:name>Natasha Noy</foaf:name>
      </foaf:Person>
    </doap:maintainer>

    <doap:release>
      <doap:Version>
        <doap:branch>stable</doap:branch>
        <doap:created>2005-02-03</doap:created>
        <doap:revision>2.3.2</doap:revision>
      </doap:Version>
    </doap:release>

  </doap:Project>
</rdf:RDF>
```

Retrieve the version number of the stable release of Prompt

RDF query (SeRQL):

```
SELECT revision
FROM
    {Version} doap:revision {revision},
    {Version} doap:branch {Branch}
WHERE Branch like "stable"
USING NAMESPACE
    doap = <http://usefulinc.com/ns/doap#>
```

Processing RDF queries

- So far, we have been using standard libraries
- Requiring every client to install a RDF query engine doesn't look like a sensible expectation
- Need for remote and shared ontology-manipulation capabilities...
- ... accessible to client, regardless of their implementation details (os, ...)

Processing RDF queries: OWS

- Need for shared ontology-manipulation capabilities...
- ... accessible to client, regardless of their implementation details (os, ...)
- That's what **Ontology Web Services** are for! [dameron et al. ISWC'04]
  - Generic ontology manipulation functions
  - implemented as Web Services

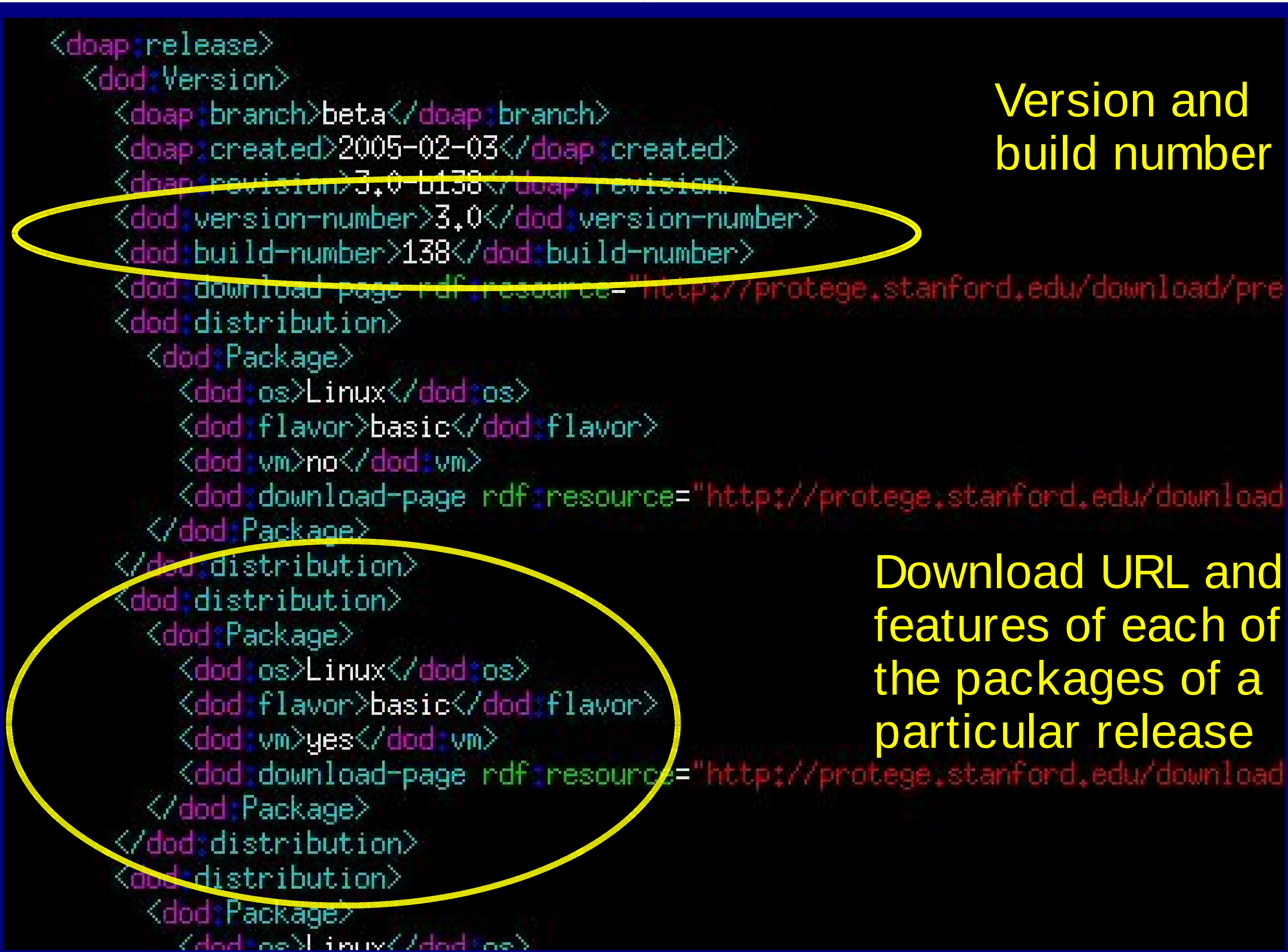
Processing RDF queries: OWS

- Wrapped Sesame SeRQL engine in a Web Service: [http://smi-protege.stanford.edu:8080/axis/services/rdfQuery]
- Parameters: RDF document + SeRQL query
- Bonus: WSDL description comes for free
- Extra bonus: we even have an OWL-S description for it (although nobody uses it)
- Clients only need standard WS library
  - Python: SOAPpy
  - Java: Axis

Enhancing DOAP for Protégé

- Reused DOAP's RDF Schema [http://usefulinc.com/ns/doap#]
- Specialized relationships [http://smi.stanford.edu/people/dameron/ontology/rdf/doap-od.rdf]
  - Multiple releases (stable vs beta) having each:
    - a version number
    - a build number
  - A single release can have multiple packages, having each:
    - a specific download URL
    - architecture constraints (OS, flavor, JVM,...)

Enhancing DOAP for Protégé



```
<doap:release>
  <dod:Version>
    <doap:branch>beta</doap:branch>
    <doap:created>2005-02-03</doap:created>
    <doap:revision>3.0-b138</doap:revision>
    <dod:version-number>3.0</dod:version-number>
    <dod:build-number>138</dod:build-number>
    <dod:download-page rdf:resource="http://protege.stanford.edu/download/pre
    <dod:distribution>
      <dod:Package>
        <dod:os>Linux</dod:os>
        <dod:flavor>basic</dod:flavor>
        <dod:vm>no</dod:vm>
        <dod:download-page rdf:resource="http://protege.stanford.edu/download
      </dod:Package>
    </dod:distribution>
    <dod:distribution>
      <dod:Package>
        <dod:os>Linux</dod:os>
        <dod:flavor>basic</dod:flavor>
        <dod:vm>yes</dod:vm>
        <dod:download-page rdf:resource="http://protege.stanford.edu/download
      </dod:Package>
    </dod:distribution>
    <dod:distribution>
      <dod:Package>
        <dod:os>Linux</dod:os>
```

Version and build number

Download URL and features of each of the packages of a particular release

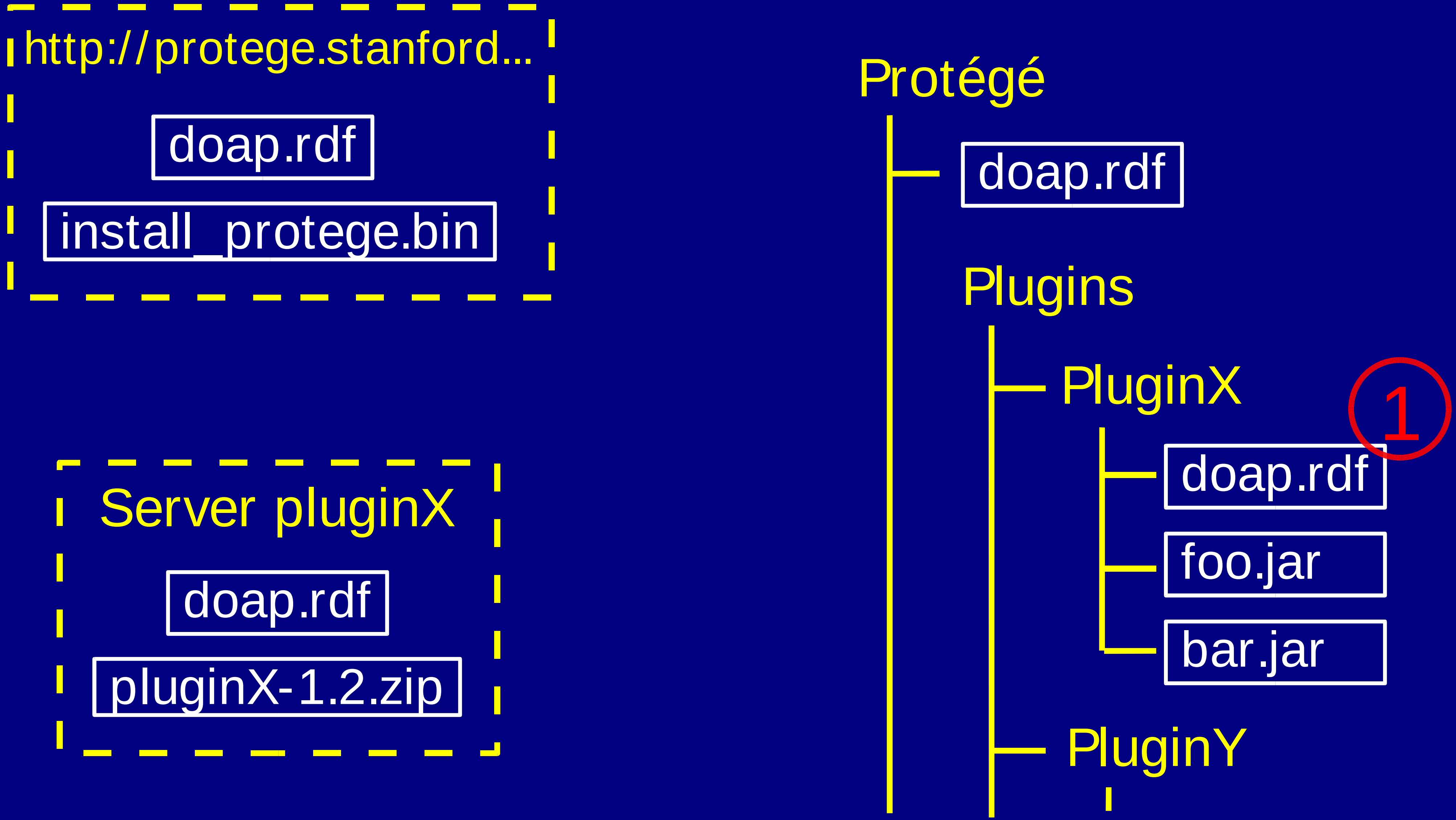
Enhancing DOAP for Protégé

- Thanks to RDF(S), the enhanced DOAP description of Protégé is still a valid DOAP file
- Therefore:
  - the previous query is still valid
  - we only have to devise a more specific RDF query to retrieve the additional information

Implementation

- Python script: checkProtege.py
  - fully automated
  - requires Python
- Protégé plugin : Automatic Update manager
  - interactive (need to click :-)

Implementation principle



http://protege.stanford...

doap.rdf

install_protege.bin

Server pluginX

doap.rdf

pluginX-1.2.zip

Protégé

doap.rdf

Plugins

PluginX

1

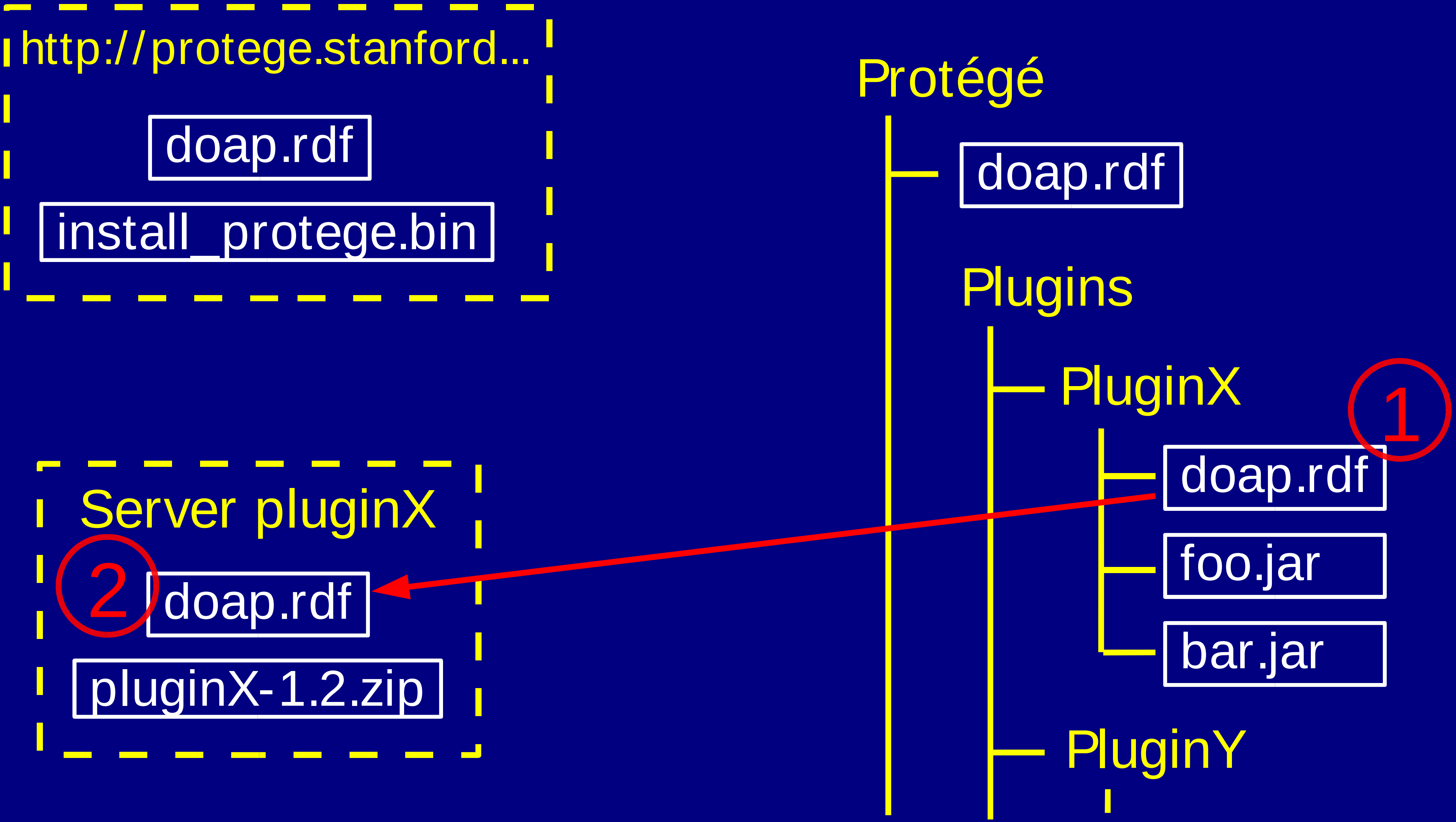
doap.rdf

foo.jar

bar.jar

PluginY

Implementation principle



http://protege.stanford...

doap.rdf

install_protege.bin

Server pluginX

2

doap.rdf

pluginX-1.2.zip

Protégé

doap.rdf

Plugins

PluginX

1

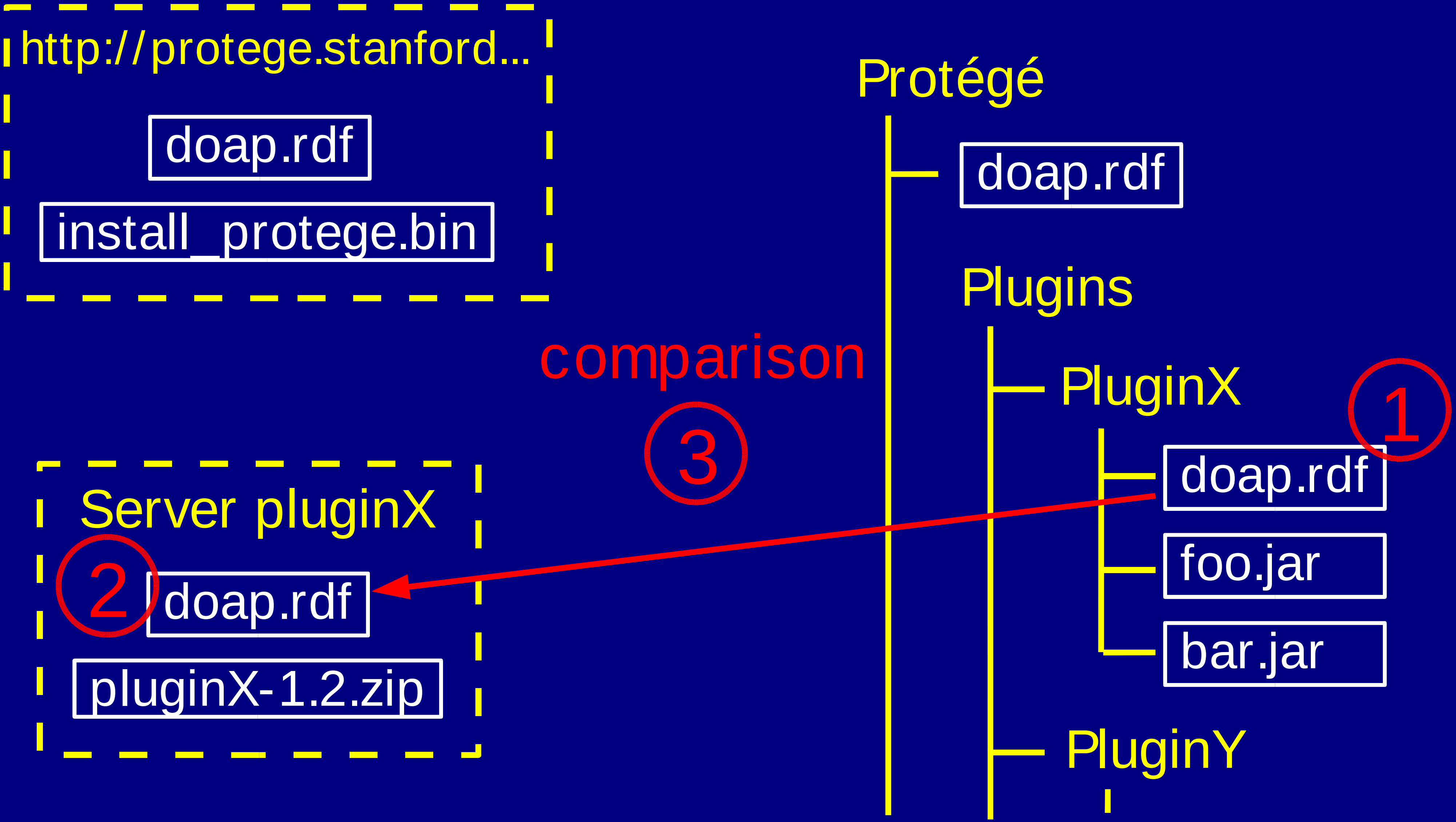
doap.rdf

foo.jar

bar.jar

PluginY

Implementation principle



http://protege.stanford...

doap.rdf

install_protege.bin

Server pluginX

doap.rdf

pluginX-1.2.zip

comparison

Protégé

doap.rdf

Plugins

PluginX

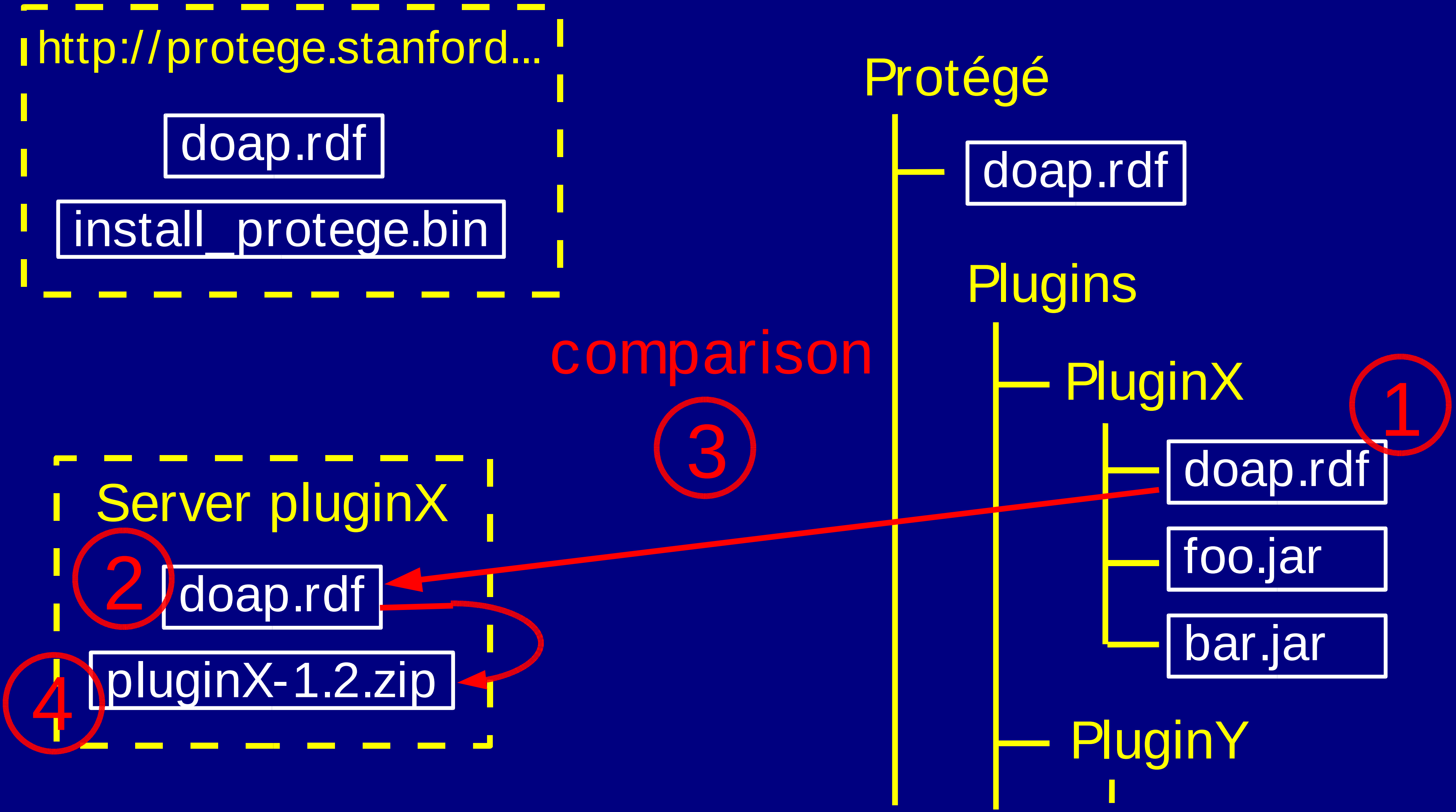
doap.rdf

foo.jar

bar.jar

PluginY

Implementation principle



http://protege.stanford...

doap.rdf

install_protege.bin

Protégé

doap.rdf

Plugins

PluginX

doap.rdf

foo.jar

bar.jar

PluginY

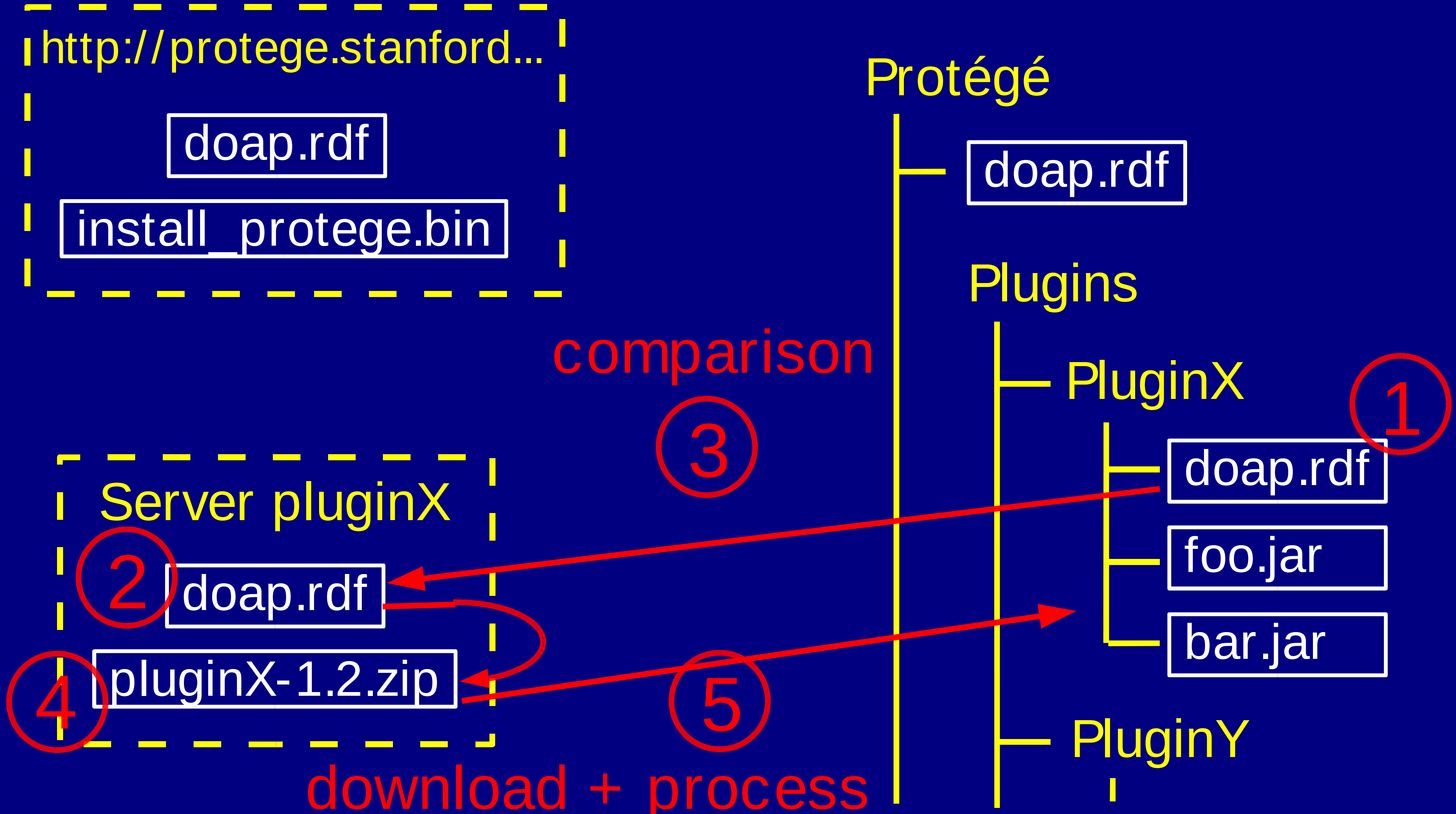
Server pluginX

doap.rdf

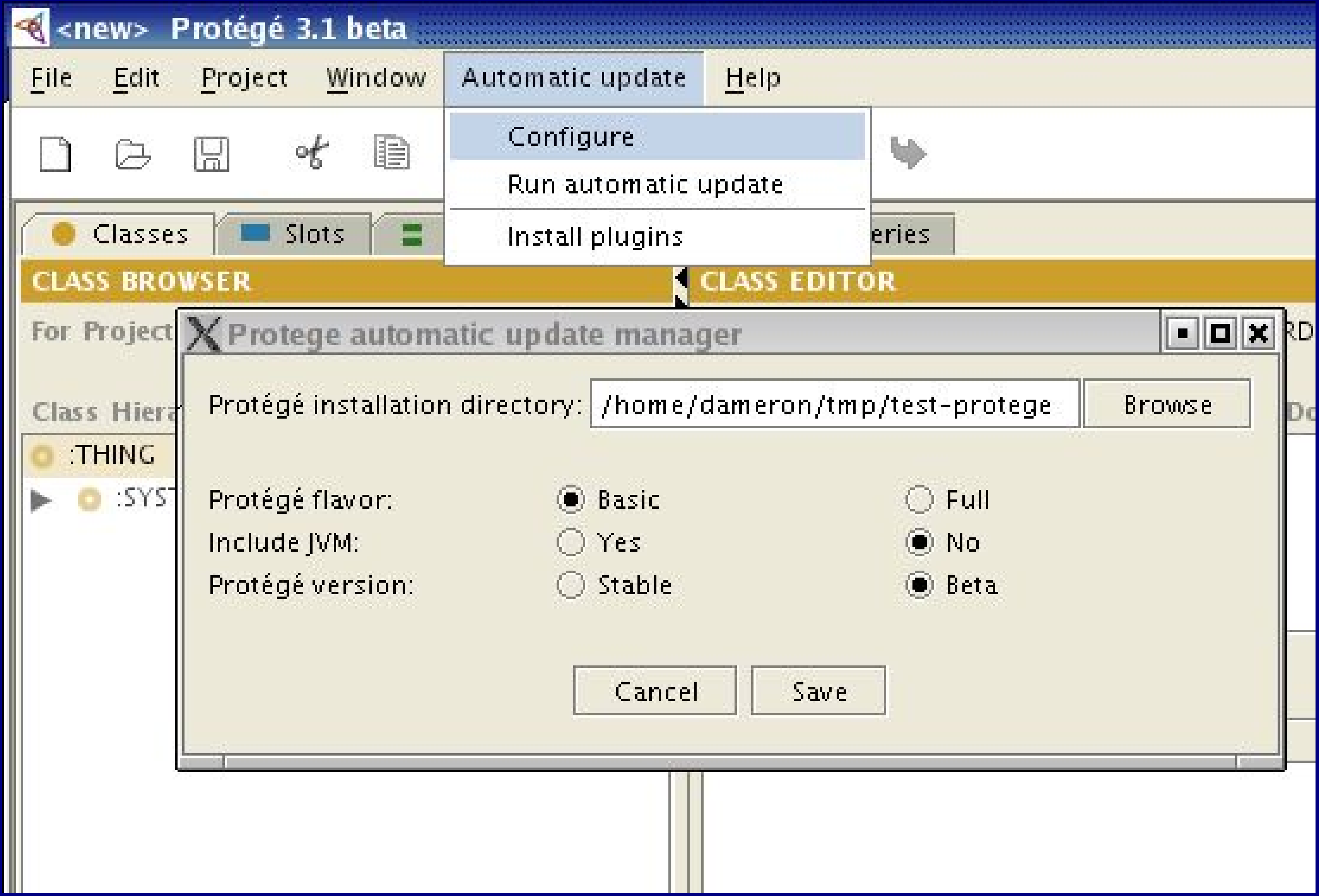
pluginX-1.2.zip

comparison

Implementation principle



Implementation



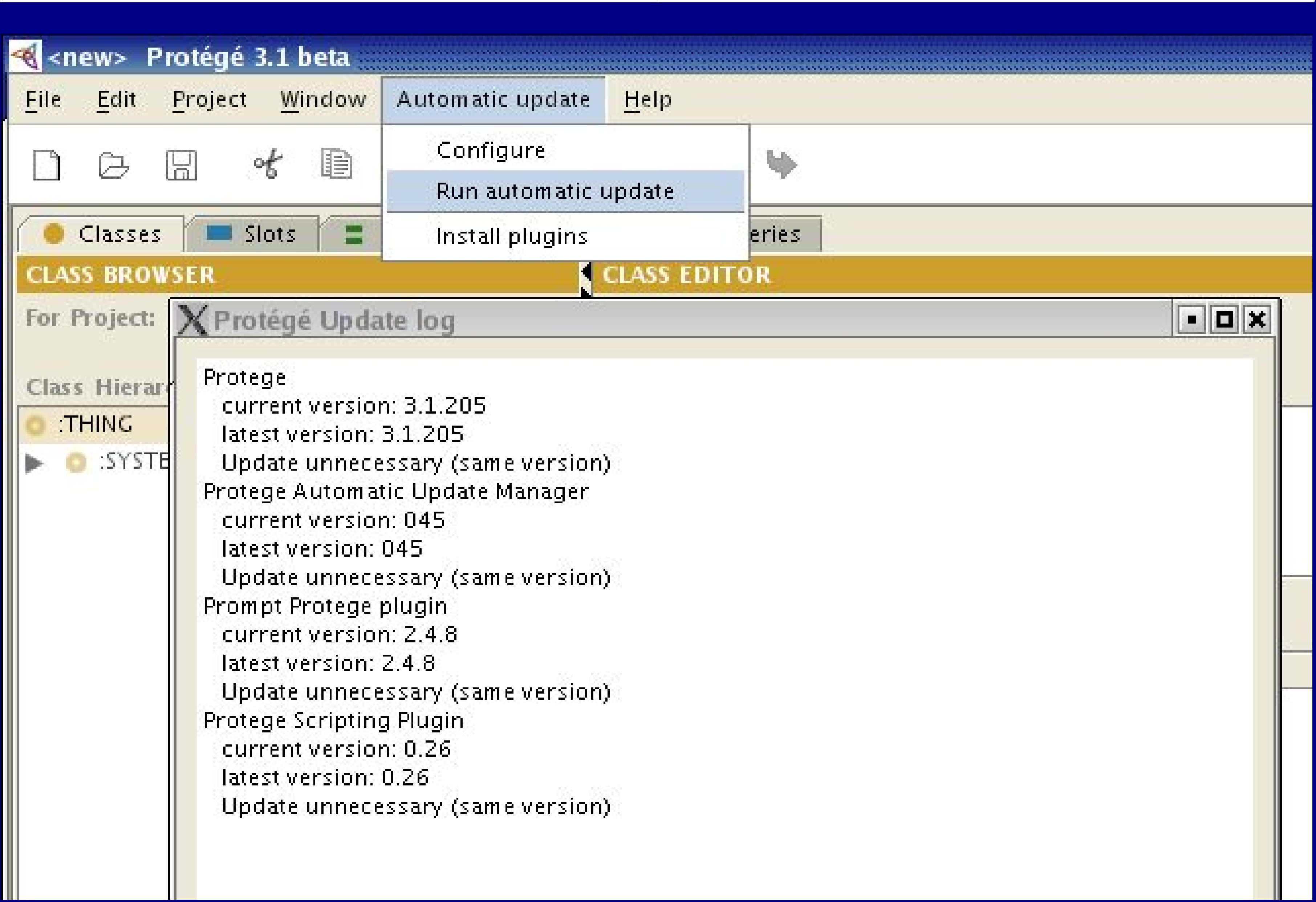
Protege automatic update manager

Protégé installation directory: /home/dameron/tmp/test-protege Browse

Protégé flavor:	● Basic	○ Full
Include JVM:	○ Yes	● No
Protégé version:	○ Stable	● Beta

Cancel Save

Implementation



```
Protege
  current version: 3.1.205
  latest version: 3.1.205
  Update unnecessary (same version)
Protege Automatic Update Manager
  current version: 045
  latest version: 045
  Update unnecessary (same version)
Prompt Protege plugin
  current version: 2.4.8
  latest version: 2.4.8
  Update unnecessary (same version)
Protege Scripting Plugin
  current version: 0.26
  latest version: 0.26
  Update unnecessary (same version)
```

Automatic self updates

- The previous principle can be applied to checkProtege itself !
- When executed, it checks if a newer version of itself is available
- If so, update itself
- Procede with Protégé et al.

Support (so far)

- Protégé
- Plugins:
  - Prompt
  - Script console
  - OWL-S ?
  - Automatic update plugin
  - **<your plugin here>**

Conclusion

- A plugin for keeping up-to-date a platform-independent and highly customizable software
- It also takes care of himself
- It Relies on semantic information provided as RDF(S) -> extensibility
- Process this information using external generic ontology-manipulation functions implemented as Web Services (OWS)

Discussion

- Other classic software update programs (apt-get, rpm, emerge):
  - are usually not supported on Windows
  - do not support user-specific config requirements
  - rely on a fixed syntax
  - require repositories (centralized or distributed)

Discussion

- Is using OWS overkill?
  - yes: most DOAP documents are alike (because developers create them by copy-paste)
  - NO: it is necessary
    - because using a syntactic approach to address an intrinsically semantic problem will always be a kludge
    - because it allows semantic scalability
  - perspective: other ontology manipulation functions (mapping...) also implemented as OWS
    - e.g. semwebcentral2doap, sourceforge2doap,...

Perspectives

- Provide doap files for Protégé and the major plugins (easy enough) (?)
  - can be automated (e.g. ant script)
  - freshmeat2doap, sourceforge2doap and semwebcentral2 doap
- Represent (and handle) dependencies
  - between software (e.g. prompt requires Protégé)
  - between specific versions of software
    - myPlugin-2.12 requires Protégé-3.1
    - Protégé-3.1 requires java-1.4