Using ontologies for R function management

Caroline Domerg, Juliette Fabre and Pascal Neveu

O. Corby
C.Faron-Zucker
E.Gennari
A. Granier
I. Mirbel
V. Negre
A. Tireau

22th July 2010

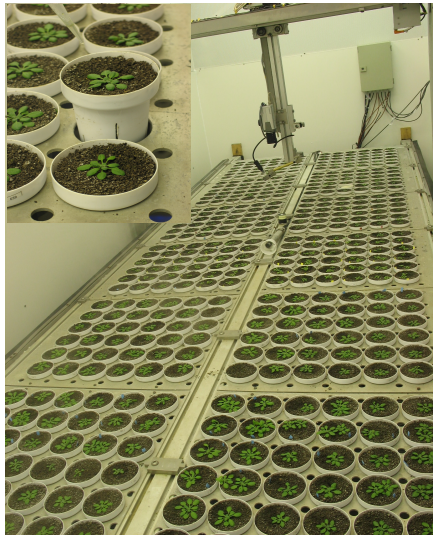
Context

Semantic Web tools

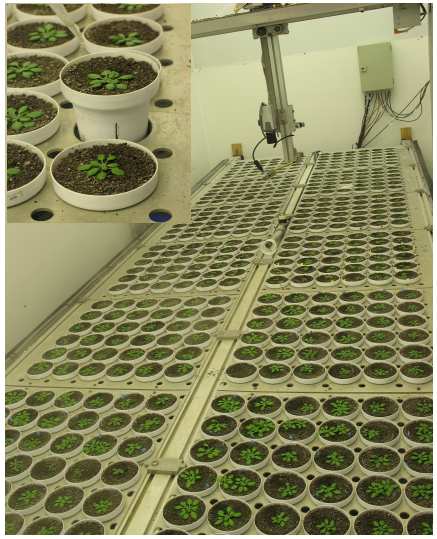
Ontology description

Web interface overview

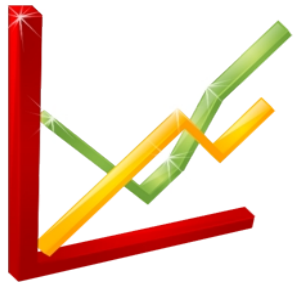
Conclusion



- Plant adaptation to climatic change
- Environmental stresses X several species
- High-throughput phenotyping
- Databases

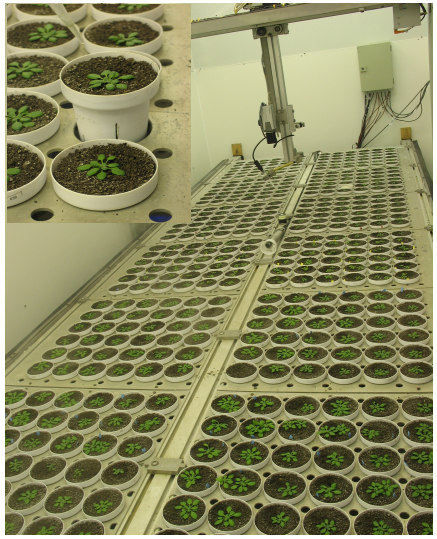


- Plant adaptation to climatic change
- Environmental stresses X several species
- High-throughput phenotyping
- Databases

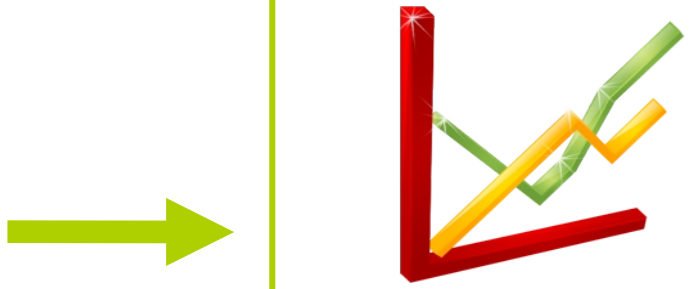


Experiment monitoring

Data analysis and modeling



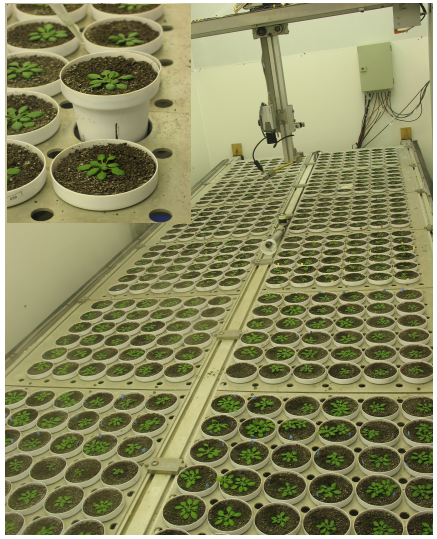
- Plant adaptation to climatic change
- Environmental stresses X several species
- High-throughput phenotyping
- Databases



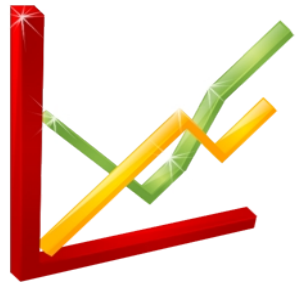
Experiment monitoring

Data analysis and modeling

High production of R functions



- Plant adaptation to climatic change
- Environmental stresses X several species
- High-throughput phenotyping
- Databases

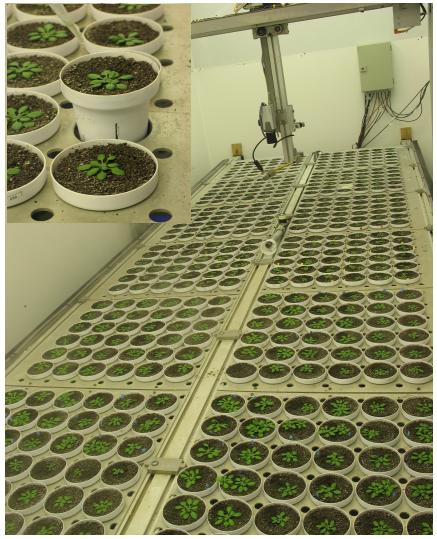


Experiment monitoring

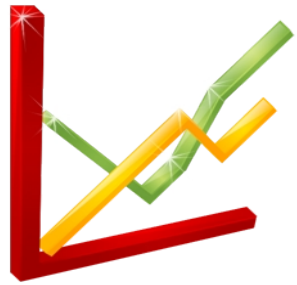
Data analysis and modeling

Many authors and turnover

High production of R functions



- Plant adaptation to climatic change
- Environmental stresses X several species
- High-throughput phenotyping
- Databases



Experiment monitoring

Data analysis and modeling

Many authors and turnover

High production of R functions

How to share, capitalize, organize and valorize these functions?

AIMS

Store and organize the functions

Give an easy and long-term access

IDEAS

Create an ontology to describe R functions

Provide a new kind of repository with reasoning and powerful search tools

TOOLS

W3C Semantic Web technologies

Ontology

Formal description of concepts and relations between concepts

Examples of concept: **Rfunction** **Argument** **Person**

Examples of relation: **hasArgument** **isANewVersionOf**

→ Provides a **controlled** vocabulary

→ Designed to be understood by **computers**

RDF, RDFS and OWL: standard tools to write ontologies

RDF

Resource: documents, images, programs, etc

Description: attributes, properties and relations

Framework: model, language and syntaxes (XML) for these descriptions

RDF

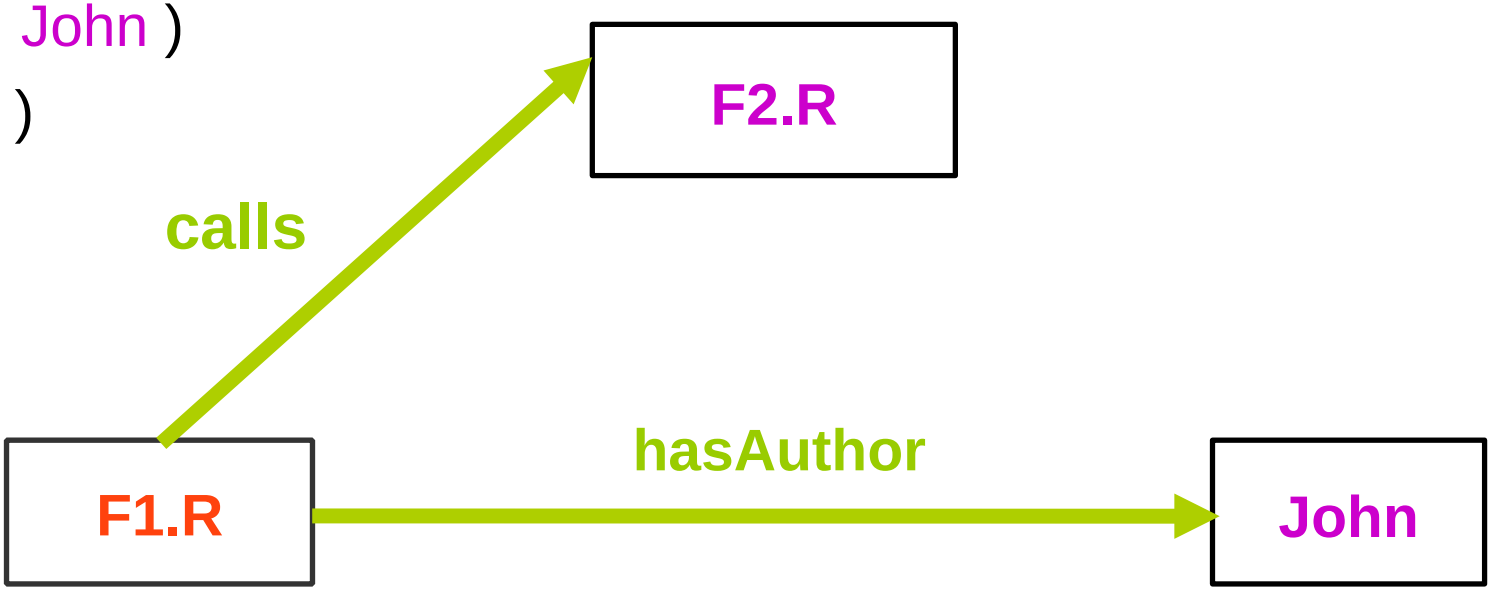
An example about R function descriptions:

→ R function attributes and properties

→ Relations between R functions

(F1.R , hasAuthor , John)

(F1.R , calls , F2.R)



RDF is a semantic graph model

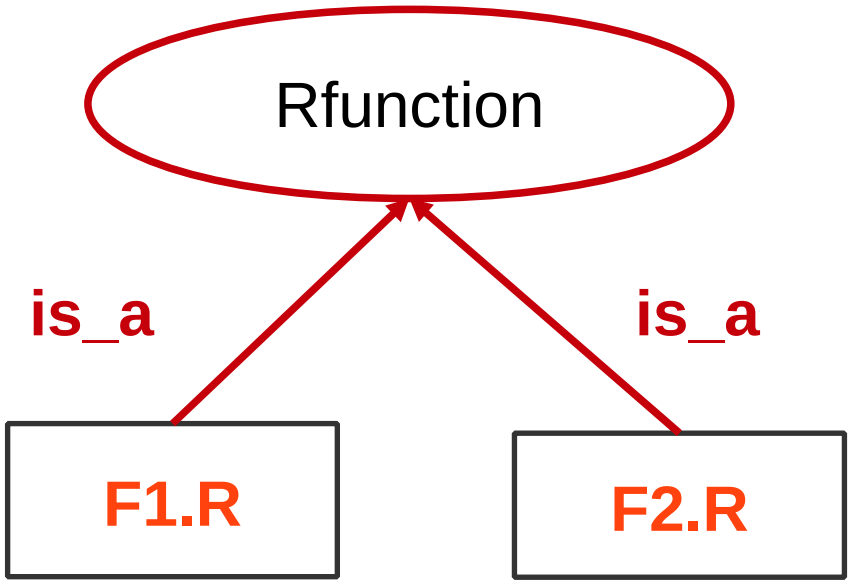
RDFS RDF Schema

Provides elements to **structure** RDF resources such as:

→ Class hierarchy

→ Property restrictions (domain, range)

Example of class: the Rfunction class



F1.R and F2.R inherit the properties and attributes of the Rfunction class

OWL Ontology Web Language

→ Built on top of RDF

→ Allows to define rules: transitivity, symmetry, inverse of, etc

➡ Provides powerful description of concepts and their relationships

Example of OWL rule: INVERSE OF

Ontology querying

RDF/OWL files

Ontology and annotations

```
<owl:ObjectProperty rdf:about="#couldBeUsedAfter">
  <rdfs:range rdf:resource="#Rfunction"/>
  <rdfs:domain rdf:resource="#Rfunction"/>
  <owl:inverseOf rdf:resource="#couldBeUsedBefore"/>
</owl:ObjectProperty>
```

SPARQL

Query

Language

for RDF/OWL

Ontology querying

RDF/OWL files

Ontology and annotations

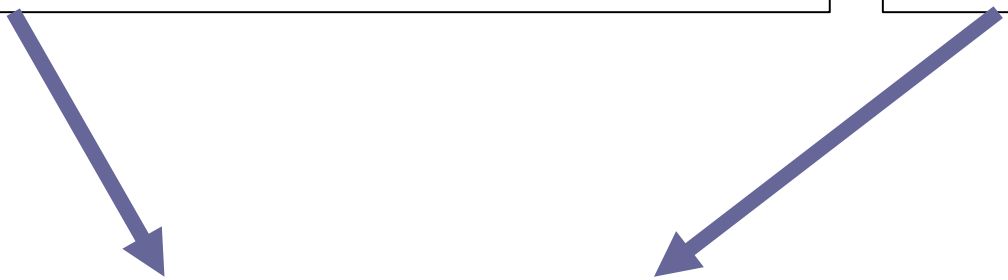
```
<owl:ObjectProperty rdf:about="#couldBeUsedAfter">
  <rdfs:range rdf:resource="#Rfunction"/>
  <rdfs:domain rdf:resource="#Rfunction"/>
  <owl:inverseOf rdf:resource="#couldBeUsedBefore"/>
</owl:ObjectProperty>
```

SPARQL

Query

Language

for RDF/OWL



CORESE (*Inria – Edelweiss*)

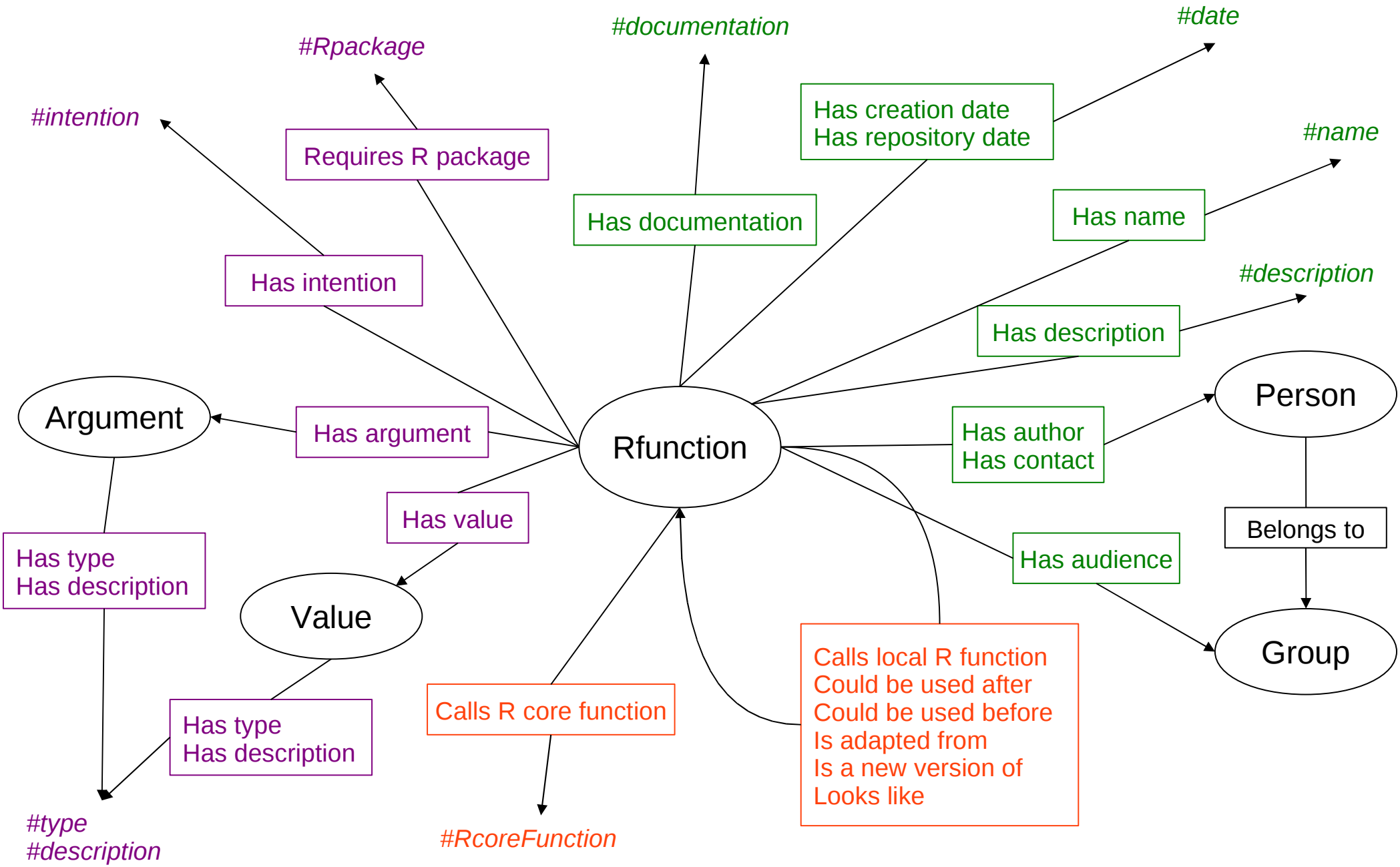
Software and engine to run:

→ rules (transitivity, etc) and to infer

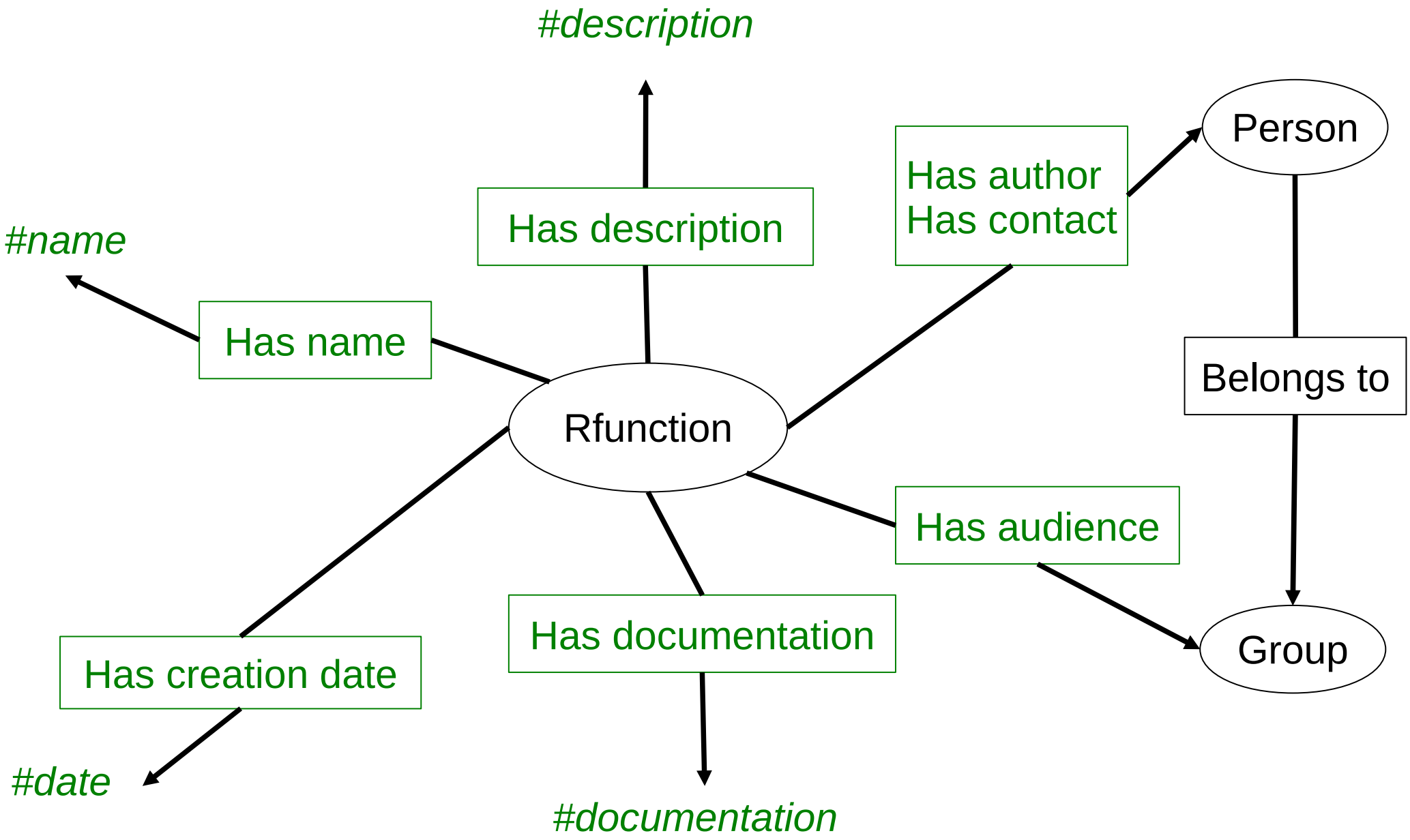
→ SPARQL queries

Query results

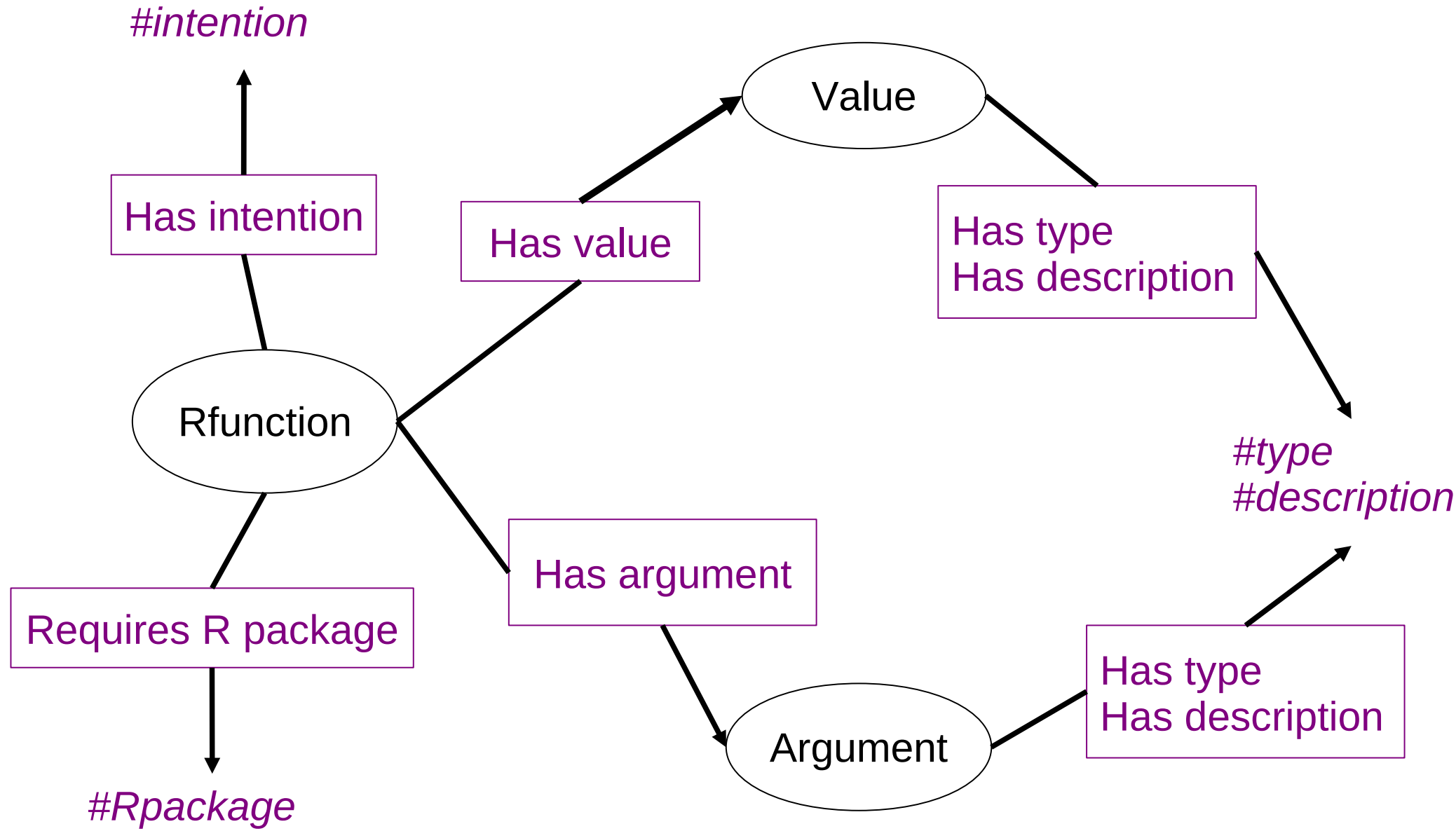
Global view


#Rpackage
#documentation
#date
#intention
Requires R package
Has creation date
Has repository date
#name
Has documentation
Has name
Has intention
#description
Has description
Argument
Has argument
Rfunction
Has author
Has contact
Person
Has value
Has type
Has description
Has audience
Belongs to
Value
Group
Has type
Has description
Calls local R function
Could be used after
Could be used before
Is adapted from
Is a new version of
Looks like
Calls R core function
#type
#description
#RcoreFunction

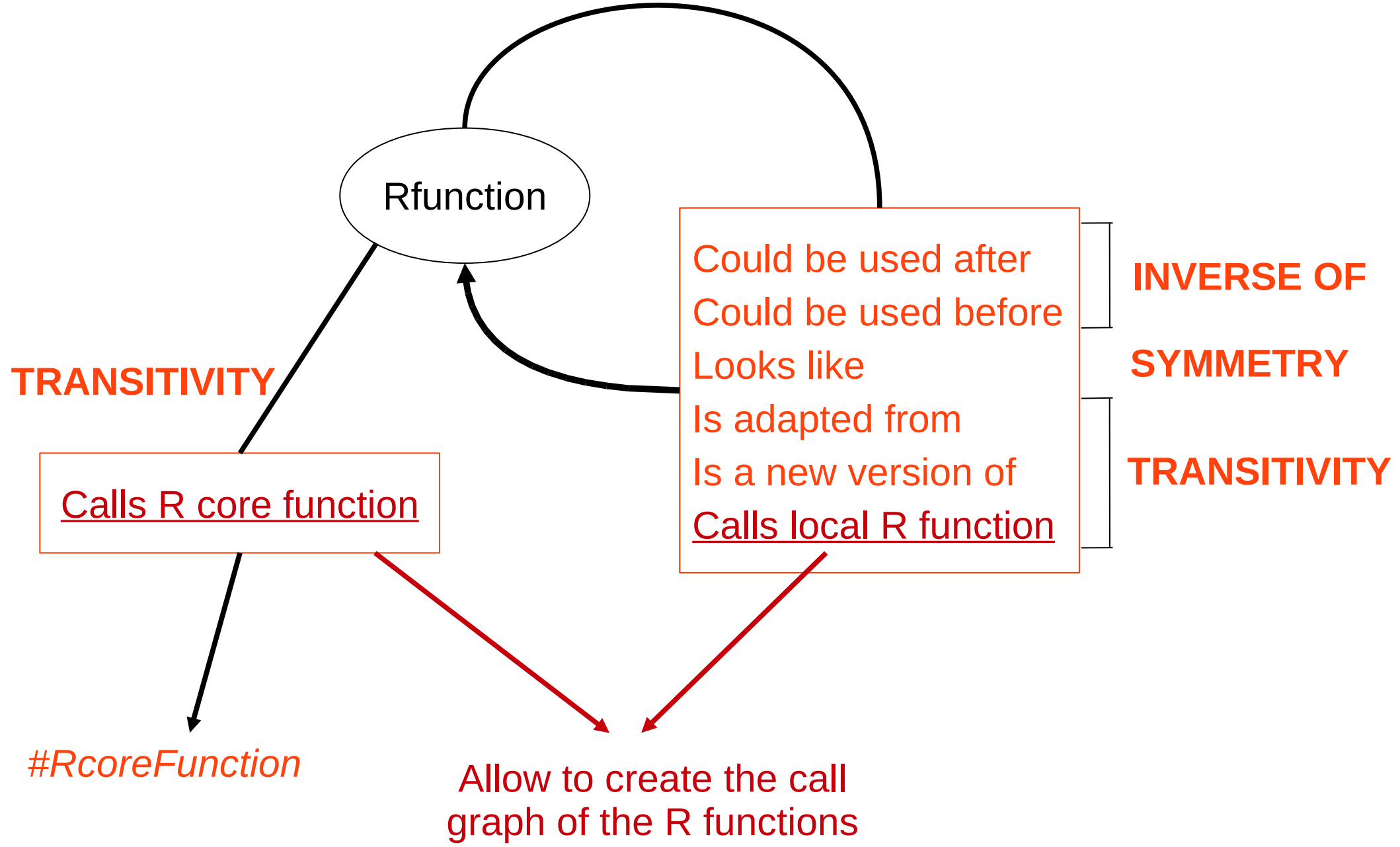
General description of R functions



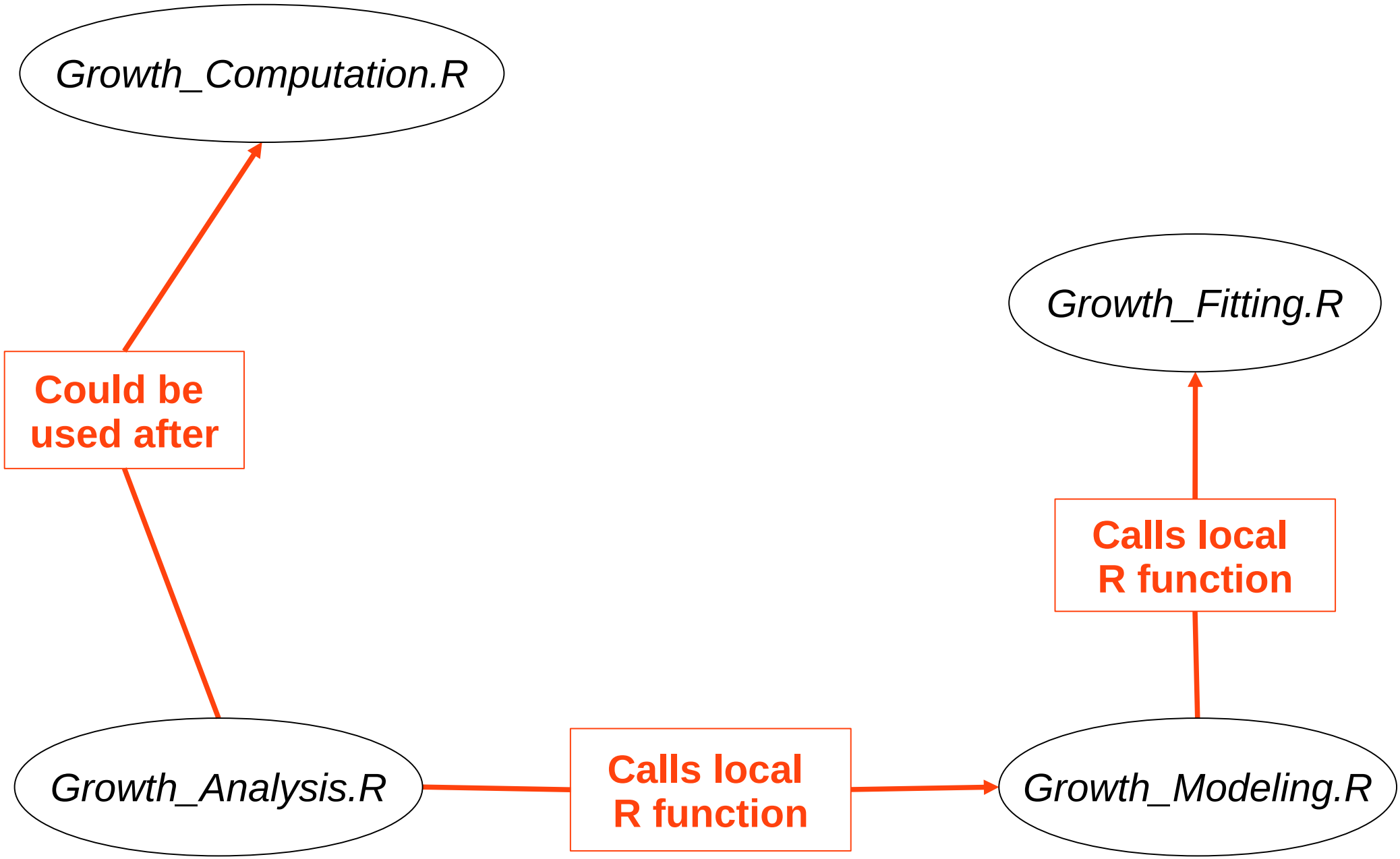
Detailed description of R functions



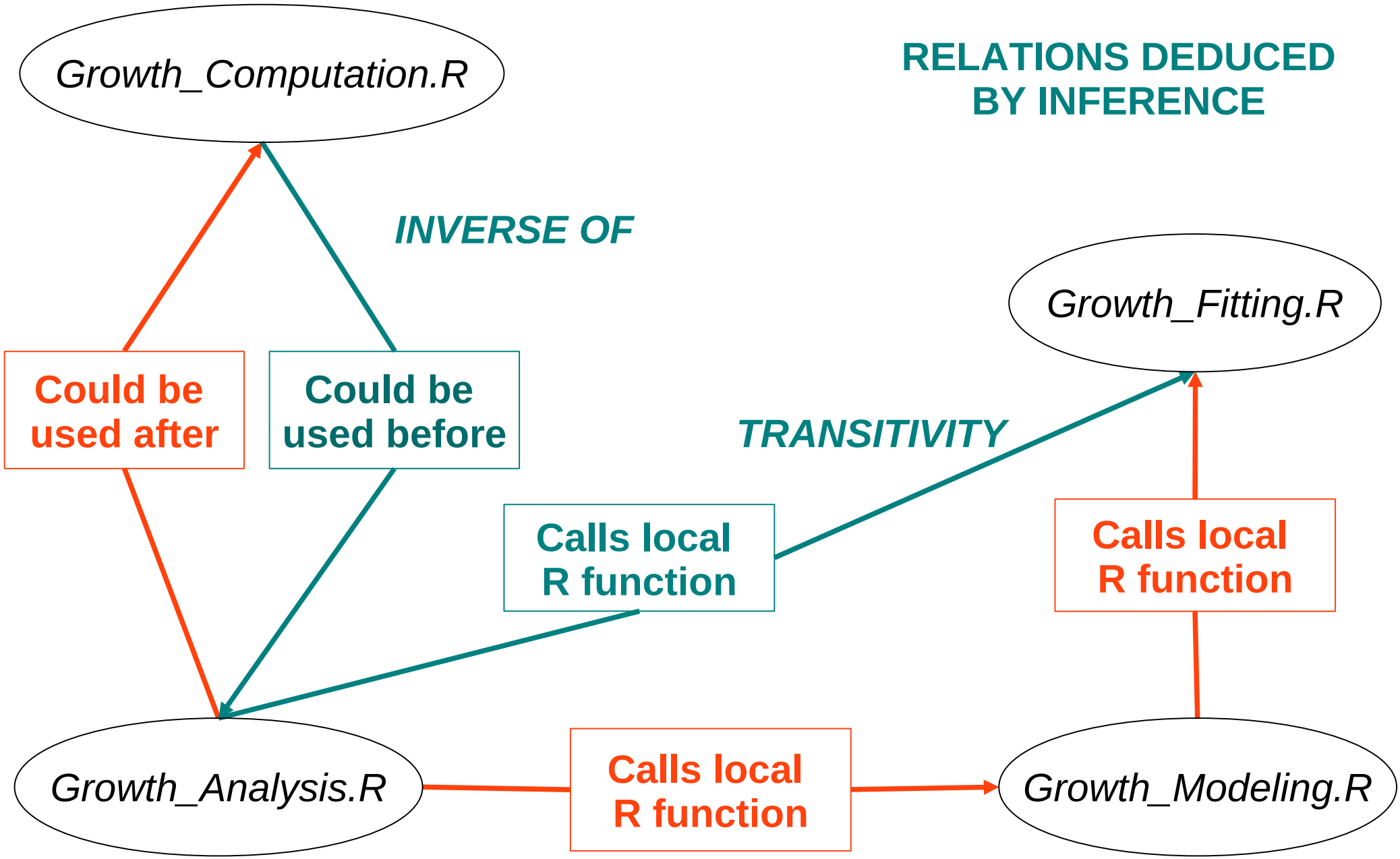
Relations with other R functions



Relations with other R functions



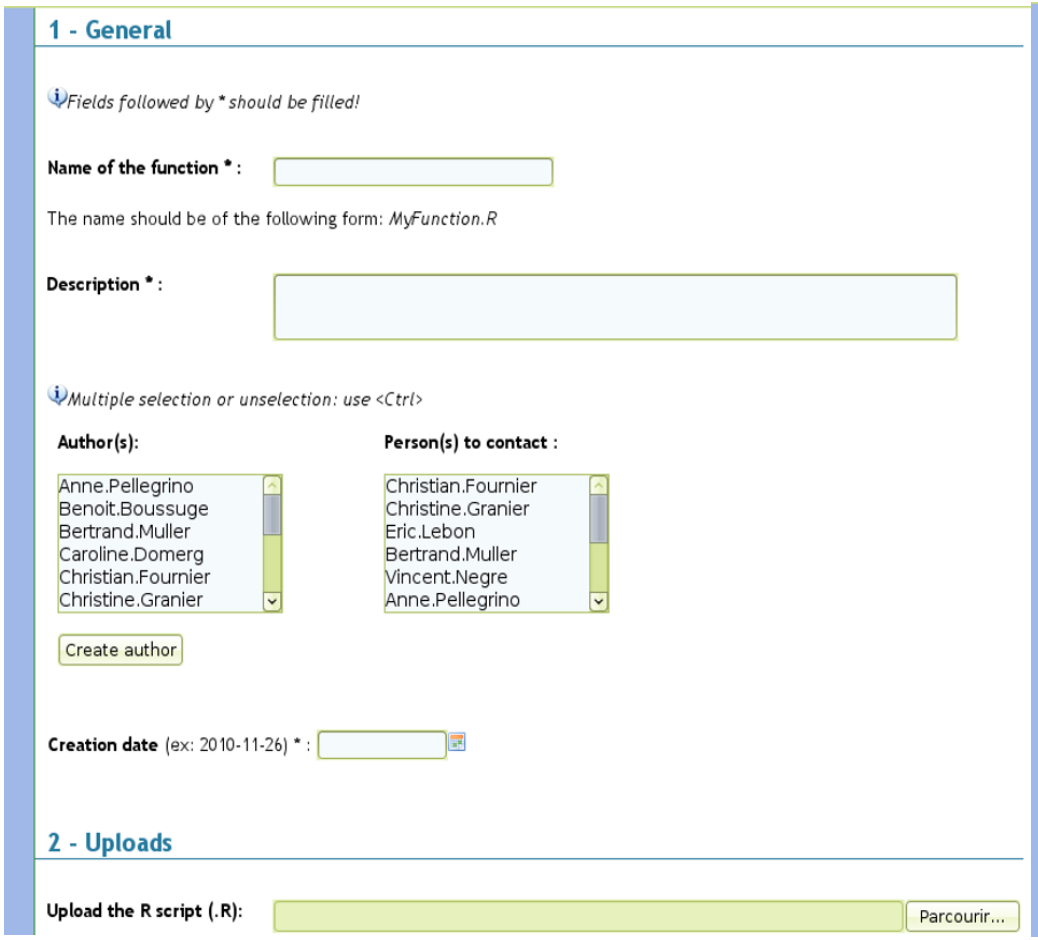
Relations with other R functions

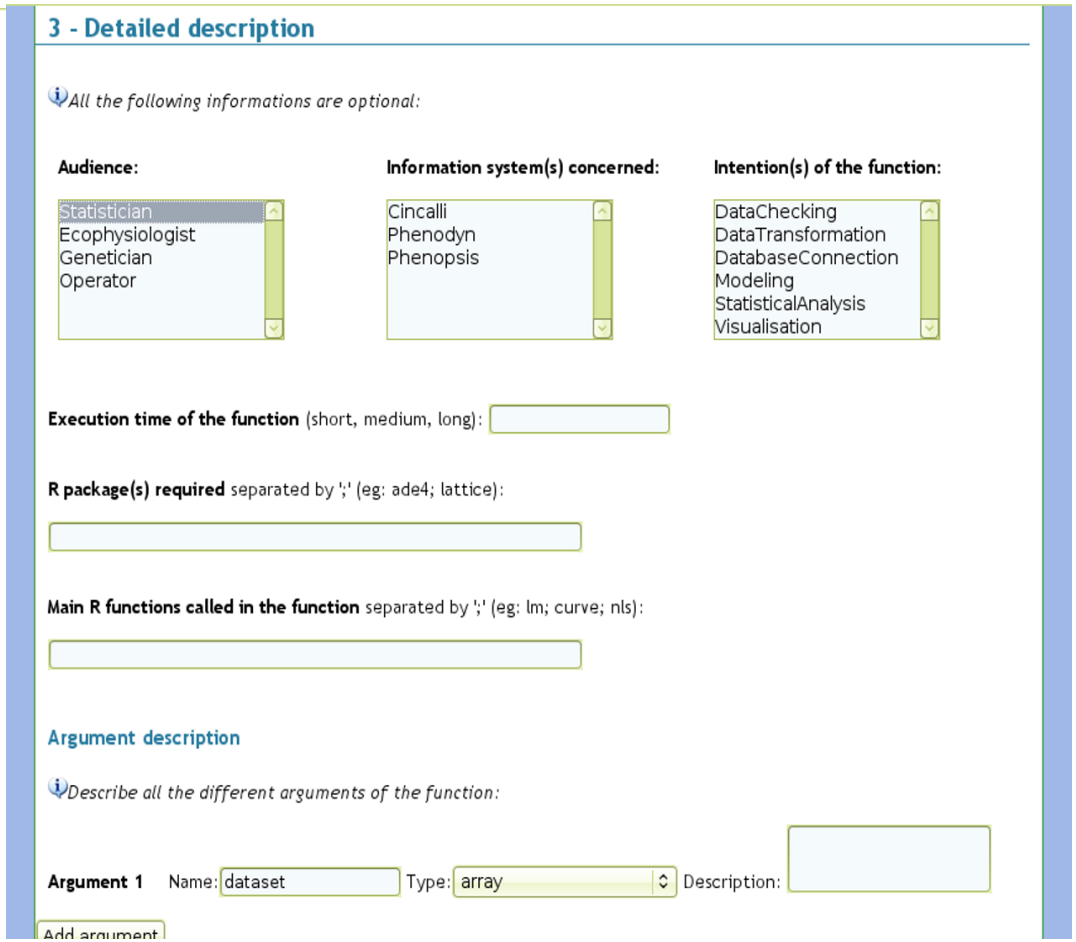


GUI for edition and creation of annotations

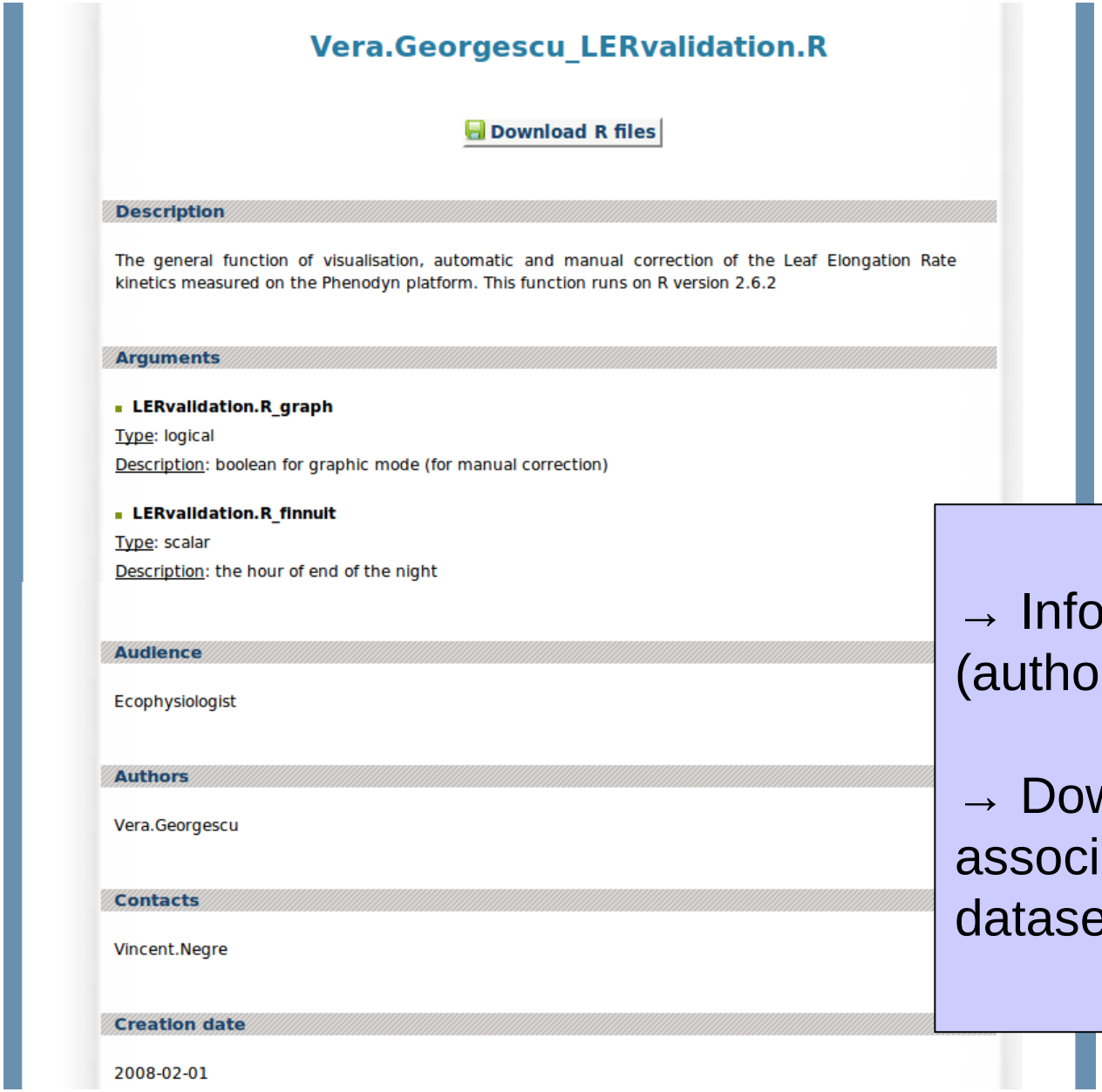
→ A few minutes thanks to pre-filled forms

→ Generation and storage of OWL file





Function consultation card



Vera.Georgescu_LERvalidation.R

Download R files

Description

The general function of visualisation, automatic and manual correction of the Leaf Elongation Rate kinetics measured on the Phenodyn platform. This function runs on R version 2.6.2

Arguments

- **LERvalidation.R_graph**

Type: logical

Description: boolean for graphic mode (for manual correction)

- **LERvalidation.R_finnuit**

Type: scalar

Description: the hour of end of the night

Audience

Ecophysiologist

Authors

Vera.Georgescu

Contacts

Vincent.Negre

Creation date

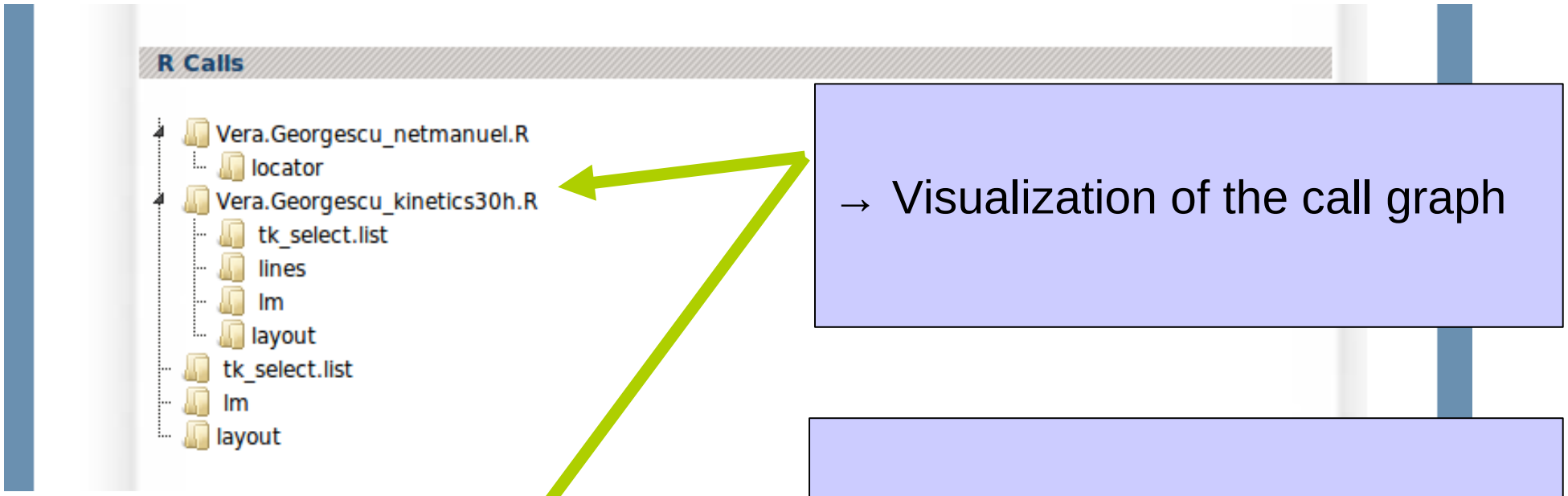
2008-02-01

→ Information about the function (author, arguments, intentions, etc)

→ Download of R function and associated files (documentation, datasets, etc)

Function consultation card

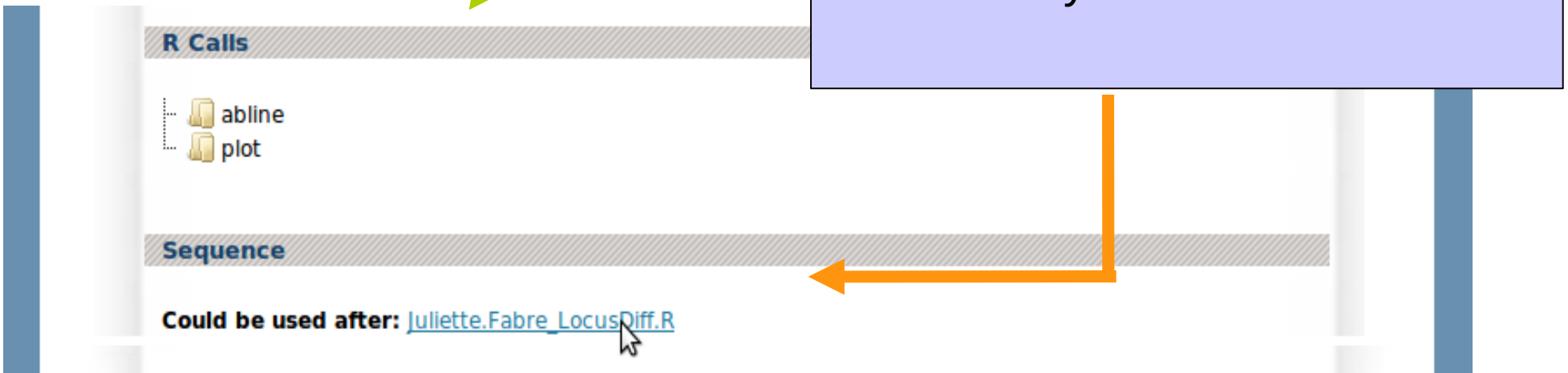
Example 1



→ Visualization of the call graph

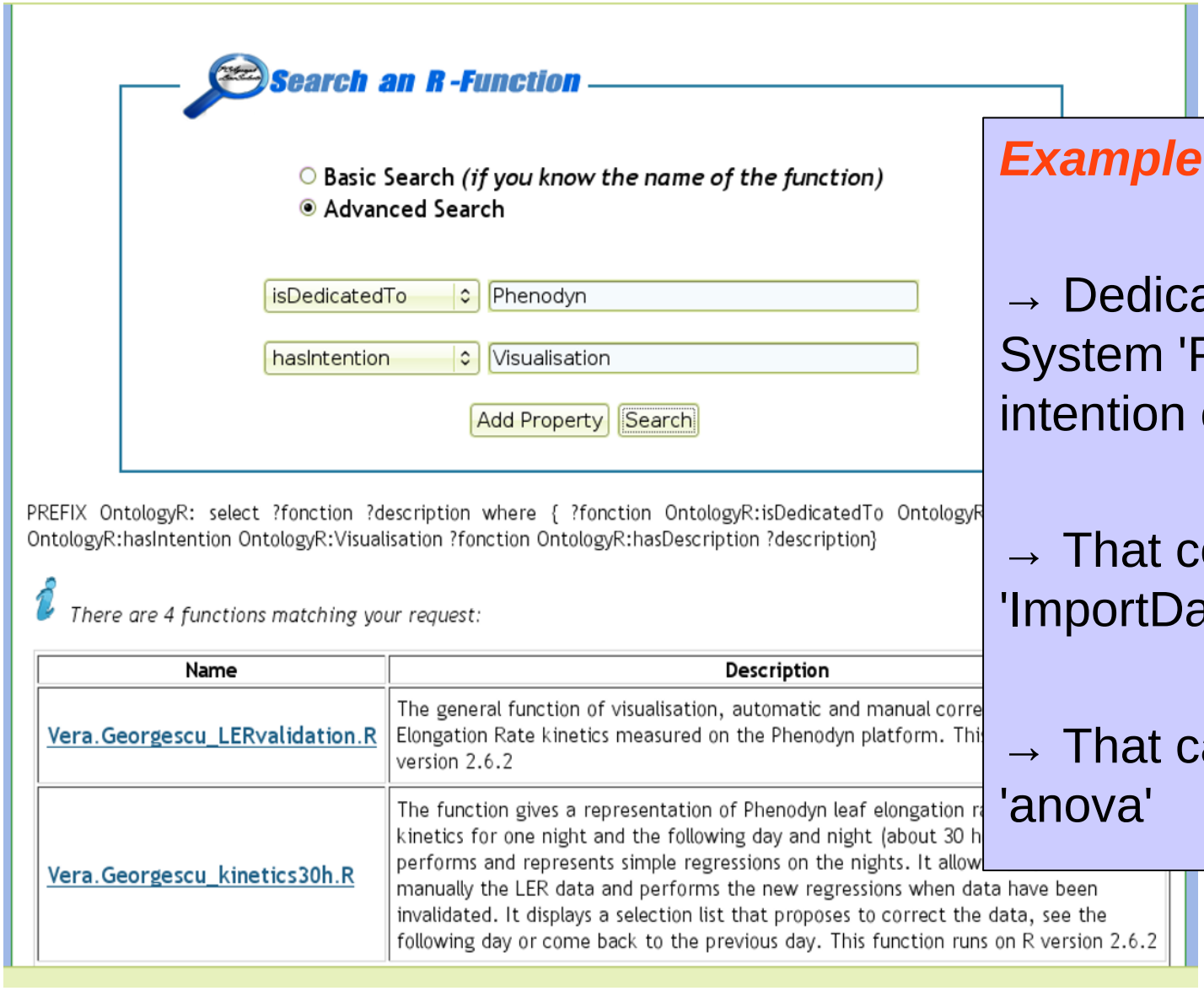
→ Hypertext links towards semantically related functions

Example 2



Powerful search tools

Build a *SPARQL* request adding conditions on the properties



Example: search the functions

→ Dedicated to the Information System 'Phenodyn' and with an intention of Visualisation

→ That could be used after 'ImportData.R'

→ That call the R core function 'anova'

Prospects

→ Add formal relations with reports, articles, etc

→ Perform more automatic extraction from R function documentation

Prospects

→ Add formal relations with reports, articles, etc

→ Perform more automatic extraction from R function documentation

Conclusions

→ Users find this repository relevant (efficient search, easy annotating)

→ Semantic Web tools allow reasoning for an 'intelligent' repository

→ Models and softwares tools are easy to adapt:

- for other fields of research
- for other programming languages
- for mathematical models