l'Observatoire de Paris

IVOA

Hawaii Interop 2013

Results of the public RFC for PDL:

answers and modifications for meeting the comments.

C.M. Zwölf, P. Harrison, J. Garrido, J. E. Ruiz, F. Le Petit

LERMA
Laboratoire d'Etude du Rayonnement
et de la Matière en Astrophysique

MANCHESTER
1824
The University of Manchester

INSTITUTO de ASTROFISICA de ANDALUCIA
CSIC
I·A·A

LUTH
Laboratoire Univers et Théories

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- Easy to fix comments:
  - The word "unique" is used a lot in the document, but the scope is not clear → we left only necessary “unique” by specifying the context of uniqueness.
  - The document itself needs a good english editorial clean-up → It has not been written by a native English speaker, but it was technically correct and precise. We have rewritten it to change the style.

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- Easy to fix comments:
  - The word "unique" is used a lot in the document, but the scope is not clear → we left only necessary “unique” by specifying the context of uniqueness.
  - The document itself needs a good english editorial clean-up → It has not been written by a native English speaker, but it was technically correct and precise. We have rewritten it to change the style.

- This is pretty complicated!
  - We are introducing all the features of a rich descriptive grammar. It is complicated but this not means that all the service-descriptions are complex.
    - Introduction of a new paragraph (2.3): PDL descriptions could be very simple in case of simple services. For example the PDL description associated with a common cone search service is very simple: http://www.myexperiment.org/files/999/versions/4/download/AMIGA-PDL-Description.xml.

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- Do we have a significant demand for these capabilities, particularly for expressing complex dependencies amongst parameters?
  - Yes we had. It's for answering these demands we worked on PDL
    - One of the difficulty often mentioned by the community to provide online codes is that, those ones are often complex to use and users may do mistake with online simulation codes. They could use them outside of their validity domain. The description of parameters with PDL allows to constrain those ones in validity domains, and so PDL answers this fear of the theorist community.
    - Workflow astro-community (cf. WF meeting @Heidelberg Interop) has needs for tools allowing a real scientific interoperability (not only based on computer types). PDL allows this. (See Taverna demo at the end of this presentation).

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- Do we have a significant demand for these capabilities, particularly for expressing complex dependencies amongst parameters?
  - Yes we had. It's for answering these demands we worked on PDL
    - One of the difficulty often mentioned by the community to provide online codes is that, those ones are often complex to use and users may do mistake with online simulation codes. They could use them outside of their validity domain. The description of parameters with PDL allows to constrain those ones in validity domains, and so PDL answers this fear of the theorist community.
    - Workflow astro-community (cf. WF meeting @Heidelberg Interop) has needs for tools allowing a real scientific interoperability (not only based on computer types). PDL allows this. (See Taverna demo at the end of this presentation).

- Could there be a simple core with extensions?
    - As already said, PDL is simple for simple description case.

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- Do we have a significant demand for these capabilities, particularly for expressing complex dependencies amongst parameters?
  - Yes we had. It's for answering these demands we worked on PDL
    - One of the difficulty often mentioned by the community to provide online codes is that, those ones are often complex to use and users may do mistake with online simulation codes. They could use them outside of their validity domain. The description of parameters with PDL allows to constrain those ones in validity domains, and so PDL answers this fear of the theorist community.
    - Workflow astro-community (cf. WF meeting @Heidelberg Interop) has needs for tools allowing a real scientific interoperability (not only based on computer types). PDL allows this. (See Taverna demo at the end of this presentation).

- Could there be a simple core with extensions?
    - As already said, PDL is simple for simple description case.

- We need some more people to read through carefully, especially some XML experts.
    - Many experts (WF4ever, ER-flow, scientists involved in Alma Science model) has adopted PDL. XML is robust.

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- UWS must not be confused with PDL.
  - Yes of course!! We rewrote paragraph 2.4.1 (PDL in the IVOA architecture)
    - If PDL describe the nature, the hierarchy of parameters and their constraints, it does not describe how this parameters are transmitted to a service, nor how these parameters will be processed by the described service.
    - This means that the separation between PDL and UWS is clear and one can be used without the other without infringing any rules of those standards

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- UWS must not be confused with PDL.
  - Yes of course!! We rewrote paragraph 2.4.1 (PDL in the IVOA architecture)
    - If PDL describe the nature, the hierarchy of parameters and their constraints, it does not describe how this parameters are transmitted to a service, nor how these parameters will be processed by the described service.
    - This means that the separation between PDL and UWS is clear and one can be used without the other without infringing any rules of those standards

- It seems quite strange to have a PDL server, PDL can be used on server and client side, but we should have a PDL server as PDL is a grammar?
  - We rewrote section 11 (Remarks for software components implementing PDL) for discussion. Details on the next slide

.

Main comments from the public RFC (summer 2013, from 20/08 to 16/09)

- Section 11 reviewed (a discussion):
  - We have connoted PDL with the term grammar. If we consider it just as a grammar, then a specific description should be considered as an implementation.
  - We remember that, since a PDL description is detailed, it is a priori possible to write once for all generic software components. These components will be automatically configured by a PDL description for becoming ad hoc software for the described code. Moreover checking algorithms could also be generated automatically starting from a description instance. In our implementations we wanted to check practically that these concepts implied in the definition of PDL really works. The development of operational services (as the Paris-Durham shock code) also permits to ensure the coherence of the core grammar and to verify if the PDL's capabilities could meet the description needs of state of the art simulation codes.
  - At the present (Fall 2013) four software elements are implemented around PDL: the Client, the Server, the Astro-Taverna Plugin, the Description editor
  - All these developments validate the concepts of automatic generation of algorithms and the possibility of configuring (to a highly specialized behavior) generic software components. This is very important since it reduces drastically the development time for building services based on the PDL grammar. This is essential in a scientific context where only few scientists have access to software engineer for their IVOA developments.