DOCUMENT

document title

IPDA Project:

IPDA Planetary Data Access Protocol

(PDAP)v1.0 preparation

prepared by	Jesus Salgado
reference	
issue	1
revision	0
date of issue	
status	Draft
Document type	Technical Note
Distribution/*distribution*	

APPROVAL

Title *titre*	issue 1 *issue*	revision 0 *revision*

author *auteur*	date *date*

approved by *approuvé by*	date *date*

CHANGE LOG

reason for change /*raison du changement*	issue/*issue*	revision/*revision*	date/*date*

CHANGE RECORD

Issue: 1 Revision: 0

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/*paragraph(s)*

TABLE OF CONTENTS

1 PROJECT DESCRIPTION 1
2 PROTOCOL SIMPLIFICATION....................1
3 CHANGES IN VERSION 1.0 1
4 PDAP UML MODEL 2
5 FREE QUERY SYNTAX PROPOSAL....................3
6 VERSION HANDLING....................4
7 PAGINATION HANDLING 5
8 PDAP EXTENSIONS....................5
9 PDAP REGISTRATION....................6
10 PROPRIETARY DATA....................6

1 PROJECT DESCRIPTION

The purpose of this project was to release a stable version of the Planetary Data Access Protocol (PDAP) so it can be distributed to the planetary community and to allow the development of servers and clients from the different IPDA member organizations or Planetary Data community.

The project was co-lead by Jesus Salgado (ESA/PSA) and Yukio Yamamoto (JAXA).
Other project members are the PDAP co-authors:
Pedro Osuna (ESA/PSA), Steve Hughes (NASA/PDS), Sean Kelly (NASA/PDS)

2 PROTOCOL SIMPLIFICATION

The following list of issues was found to be solved in order to publish a public version of PDAP:

1. New data Types to be analyzed: Specific data types would need different flavours of the PDAP protocol to describe properly the access to them
2. Publishing of proprietary data: Due to project specific constraints not all the data to be published by a PDAP interface is public. A specific interface definition for data that requires authorization would need required.
3. Free query language: Current PDAP interface does not allow complex queries and it does not allow clients to select metadata to be returned by the server in a flexible way

These three points were found quite controversial and difficult to be solved in a short term. As this is stopping several groups the implementation of PDAP services, it has been decided to publish a simplified version of the protocol, leaving future improvements to next versions in the following way:

1. New data Types to be analyzed: An extension mechanism is proposed to handle complex or very specific data. Separate documents will describe how to publish these services and how this specific documents are related to the main PDAP specification
2. Publishing of proprietary data: Secure protocols (like https) will be allowed in future versions and authorization mechanism (e.g. http ones) will be discussed. Single sign on possible approach will be also discussed.
4. Free query language: Special keywords will allow complex queries to the PDAP services. To allow this, a specific query language should be defined and it should be described how to publish this capability at registry side.

3 CHANGES IN VERSION 1.0

The following changes have been proposed and implemented in version 1.0:

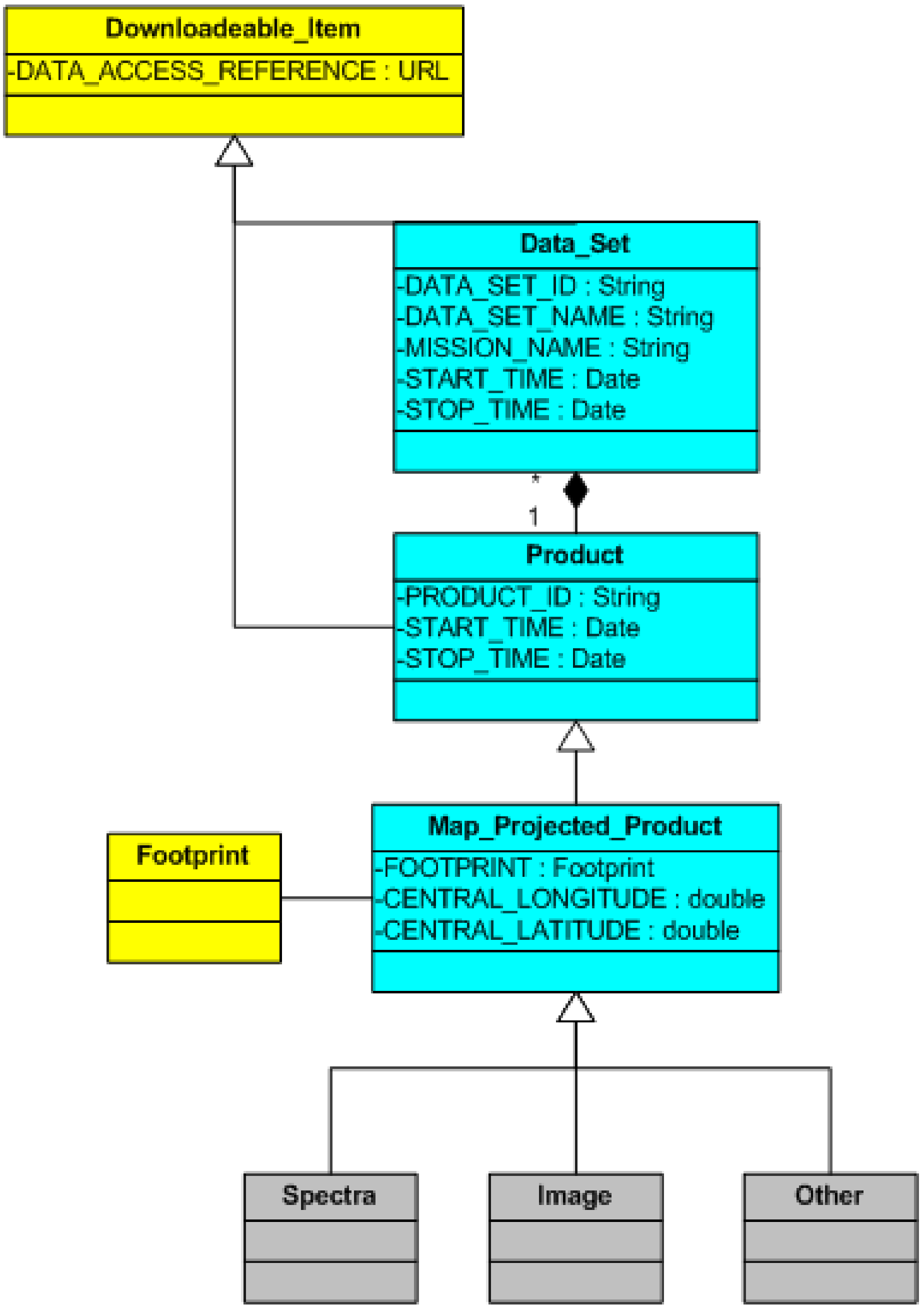
1. **Resource Class**: Only data set, product and map-projected data is described in the specification. It is described how to extend these basic classes and this would be done in separate documents.
2. **Version handling:** Added

3. **Pagination**: Although it was requested, some problems were found to add it in a simple PDAP interface as the default output format, VOTable, does not describe a standard way to incorporate pagination tags inside it. To be done in coordination with IVOA VOTable definition group.
4. **Ellipse** has been removed as possible footprint due to the lack of use and difficult geometrical definition. Only polygon, circle and landmark supported
5. **Registration of services:** External link to registry work in IPDA or other registration efforts. Details out of the scope of PDAP
6. **BNF removed** as considered by some authors as not very useful and confusing . To be extracted as a separate document
7. **Image services called now Map-projected data** services as it defines products with a footprint attached
8. **References** added
9. **Basic UML added** to clarify interrelations between the various resource classes and to clarify where PDAP extensions could be developed

4 PDAP UML MODEL

The following diagram describes the items supported in this PDAP v1.0 that tries to give a simplified view of the planetary data that can be exposed in a PDAP service.

As a summary of the classes described in PDAP v1.0, a data set contains a set of products. Both data sets and products can be downloaded using a data access reference. As one example of product extension, a class map projected product is also included in present specification. This map projected product class has associated a footprint object indirectly described in present specification.



5 FREE QUERY SYNTAX PROPOSAL

There is a need in the PDAP definition of a better granularity for queries, e.g., a similar freedom that the one a SQL could have to specify the specific fields to be returned (SELECT) or complex query conditions (WHERE) that are not allowed in PDAP simple version, based on keyword=value paradigm.

During the project, it was discussed the introduction of RESULT parameter as per Steve Hughes proposal, that would allow clients to specify the SELECT part of a certain query.

Similar work, done in IVOA context, (TAP Table Access Protocol) was also discussed. Two approaches were followed:

- ADQL query: SQL + geometrical conditions approach. Very powerful although difficult to implement. Example:
  http://example.com/tap/sync?REQUEST=doQuery&LANG=ADQL&
  QUERY=SELECT * FROM magnitudes as m where m.r>10 and m.r<16

- PQL approach: Based on the keyword=value approach but leaving more freedom to the client the Parameter Query Language (PQL) is easier to be implemented but it is less powerful and it defines a ad-hoc language. Example:
  http://example.com/tap/sync?REQUEST=doQuery&LANG=PQL&
  FROM=magnitudes&WHERE=r,10/16

Although the selection of a certain language could be part of the PDAP specification, the definition of separate specification for the language was recommended, so the simplified PDAP v1.0 would only content the current keyword=value approach.

6 VERSION HANDLING

As the present project recommends the creation of a preliminary and simplified PDAP specification, leaving room to future extensions, the version handling is required as really needed to allow client applications to select or parse differently PDAP servers.

To make this possible, a PDAP compliant output **should** contain an INFO tag with name PDAP_VERSION and value the PDAP specification version implemented by the service. This INFO tag must be inside the RESOURCE tag due to VOTable compatibility. The tag will be, then, as follows:

<INFO name="**PDAP_VERSION**" value="**0.5**"/>

This information should appear in a RESOURCE_CLASS=METADATA discovery

This tag has two different uses:

1. Provide information to client applications about the version of the PDAP specification implemented by the PDAP server so clients could prevent parsing problems.
2. Allow automatically discover version information by registry servers.

7 PAGINATION HANDLING

Neither IVOA data access protocols nor OpenGIS protocols offer pagination tags. However, this has been requested many times for PDAP services.
The only examples found that could be used as a reference for pagination tags are the IVOA registry and the Google kml services.

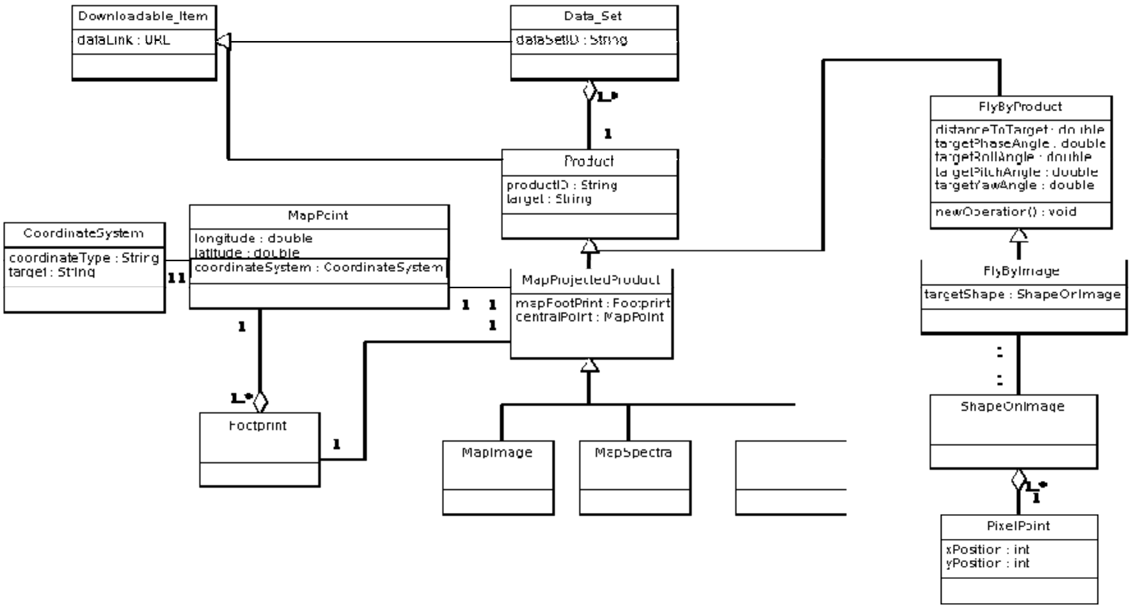
One issue found during the investigation is that VOTable, the standard output format used in PDAP to be close enough to IVOA protocols for a future possible integration, does not define how to include pagination tags in the response. Other output formats (e.g. JSON) have de-facto standards for this (see YUI server paginated table).

In order to include this in a future PDAP version, IVOA VOTable group members were contacted (and also the CDS SAVOT parser, the main VOTable parser in the VO context) asking for a way to incorporate this tags in a future VOTable specification.

Our recommendation is to leave pagination support in PDAP for a future release, unless this is not considered necessary for this PDAP v1.0 by the TCG members.

8 PDAP EXTENSIONS

There are many ways to extend PDAP for specific data types. One of the first possible extensions was the support for fly-by products. In this case, it is not realistic to assign a footprint of the image on the target, but PDAP basic UML could be linked for this new product in the following way:



After some discussions, it was proposed to define a extension mechanism for specific data types. That implies the creation of a simple PDAP spec and separate documents that should describe how the new classes described in this extension link to the basic UML and what are the new fields that should characterize them.

Two separate projects were created as starting point of this extension mechanism to define Fly-by products (lead by Yukio Yamamoto) and Single Files extension (lead by Jesus Salgado)

9 PDAP REGISTRATION

Although preliminary versions of PDAP specification tried to describe how to register PDAP services it was identified as out of the scope of PDAP.

References to PDAP registration were removed from PDAP spec and a different project on registries was created under IPDA umbrella.

10 PROPRIETARY DATA

PDAP services describe only how to handle public data. In order to allow the use of proprietary data, some technical problems should be solved first:

- Secure protocols should be used. http is not encrypted so it could represent a security hole. One possible alternative could be the use of https. Although that implies the use of certificates, it is quite transparent from the server and client sides

- Centralization of users. Users usually have different roles, names and privileges in different planetary data centres. Users could even have different registration details (usernames and passwords) per data centre. A ideal solution could be to homogenize the user registration at IPDA level, but that implies many technical difficulties and political ones.

One possibility could be to add a tag alerting that the data is protected by proprietary rights and request for an http session (using authorization cookies). That was already implemented in the ESA PDAP server but this should be included in PDAP as a standard.

However, the approach is still controversial, so that could be included in a future version.