Theory session

Moscow INTEROP
2006.09.18 14:00

Agenda

- Theory Semantics (UCD, standard vocabulary)
- Simple Numerical Access Protocol (SNAP)

- Working drafts downloadable at: http://www.ivoa.net/twiki/bin/view/IVOA/InterOpSep2006Theory
- Issues list and discussions: http://www.ivoa.net/twiki/bin/view/IVOA/IVOATheorySNAP

From Victoria to Moscow: actions near completion

- **Vocabulary/semantics**
  - Preliminary vocabulary list to describe simulation code/dataset metadata compiled in Victoria
  - Updates/improvements by email -> July
  - WD of technical note v0.1 (L. Shaw et al.)
  - released on ‘theory’ list on Sep 11 + feedback→update Sep 14
  - **Action near completion**
  - Need to converge on a few open issues (e.g. physics/algorithm description…)
- **Simple Numerical Access Protocol**
  - WD technical note v0.1: first draft (C. Gheller et al.) + telecons on Sep 8 and 15 to refined
  - **Action near completion** (need data model for query services)
  - Presentation + feedback (e.g. data staging)

From Victoria to Moscow: actions in progress/delayed

- **Data modeling**
  - action in progress (needed for SNAP). G. Lemson et al.
  - Should be tied in with SNAP → SNAP v0.5 next INTEROP
- **Data formats**
  - Decided in Victoria to stick to VOTable where appropriate
  - Translators from various usual formats to be done
  - Poll theorists about the data format they use in their simulations
  - Actions delayed until next INTEROP

Theory Semantics Vocabulary/UCDs

- Attempt to find all words used to describe simulation codes and/or datasets
- Breakdown in categories:
  - (d) Name of Dataset
  - (d,c) Name of the developer/team/contact
  - (d,c) Name of Code
  - (c) Version of Code
  - (d,c) Description of the code (text)
  - (d,c) Physical Objective
  - (d,c) Physical Process
  - (d,c) Subject(s)
  - (d,c) Algorithm
  - (d,c) Time evolution
  - (c) Protocol
  - (c,d) Result format
  - (d) Results Parameters

Theory Semantics Vocabulary/UCDs

- Basic attributes:
  - (d) **Name of Dataset** as for obs. data
  - (d,c) **Name of the developer/team/contact** as for obs. data

  - **(d,c) Name of Code**
    UCD: meta.id;comp.code.main or comp.code.secondary
    pb with code of codes
    pb with no-named codes

  - **(c) Version of Code**
    should be leave to author's responsibility ?
    Use standard versioning (e.g. 2.1.0) or date (2006-09-17)?
    pb when name of code includes version (e.g. Gadget2)

  - (d,c) **Description of the code** (text)
    UCD: meta.note

Theory Semantics Vocabulary/UCDs

- Physics of the code:
  - (d,c) **Physical Objective**
    Purpose is to give a general indication of overall phenomenon that is being simulated, or the physical objective. What are we trying to simulate? Aim is to provide an umbrella term for the collection of physical processes that are modeled or simulated.
    UCD: phys.process;comp.simulation;meta.main
    examples: formation, evolution, accretion, merging, stellar population synthesis, explosion, etc.
  - (d,c) **Physical Process**
    used to describe the level of detail in which the Physical Objective is investigated. Allows a user to determine the level of physical detail the simulation achieves. Does it account for this/that/etc? Consists of a list of words chosen from the standard vocabulary **but IVOA std. vocab. is incomplete [v0.8 20060505]**
  - (d,c) **Subject(s)**
    UCD: obj(ect);meta.main
    Words from the std vocab to describe all the (astrophysical) objects in the simulation [not the physical objective: e.g. for stellar evolution, subject is 'star']

Theory Semantics Vocabulary/UCDs

- Code operation:
  - (d,c) **Algorithm**
    this is purely to describe the numerical procedure being used to evaluate the physical processes being simulated. Difficult to list all algorithms. Which ones? which level of refinement ?
    UCD: comp.alg
  - (d,c) **Time evolution**
    UCD : comp.sim.timeEvolution
    yes/no
  - (c) **Protocol**
    if a code is parallel, gives the protocol used. Describes type of parallelization – useful for grid and webservices group?
    UCD: comp.protocol
    examples: OpenMP, MPI

Theory Semantics Vocabulary/UCDs

- Results metadata:
  - (c,d) **Type of Result**
    give the type of result produced by simulation
    examples: snapshot, animation, table, FITS, catalogue of objects, statistics of objects
  - (c,d) **Result format**
    Gives an indication of the type of output produced by a simulation – is it raw unprocessed particle data, or mock images/spectra or a halo catalogue from FoF etc.
    UCD: meta.id
    Examples : spectra, object catalogue, statistics, fit parameters, raw particle/grid data
  - (c,d) **algorithm Parameters**
    UCD: comp.sim.params;comp.alg
    Examples: number of particles, box size
  - (c,d) **physical parameters**
    Same as above, but for input physical parameters to simulation. Should really be UCDs for all this, e.g. phys.cosmology.omega (matter/energy density of universe)

SNAP

- Simple numerical access protocol
- SNAP specification 0.1
- Main goal:
  - protocol for retrieving data coming from numerical simulations from a variety of data repositories through a uniform interface
  - should be reasonably simple to be implemented by service providers.
- Issues list and discussions:
  http://www.ivoa.net/twiki/bin/view/IVOA/IVOATheorySNAP

SNAP main features

1. A query defining the interesting physical models is used for searching for candidate simulations and related data.
2. The service returns a list of the candidate simulations.
3. The service can be further queried in order to get information on data associated to interesting simulations.
4. data can be further selected, choosing specific quantities and extracting rectangular or spherical sub-samples from the simulated volumes.
5. Data are be returned in VOTable simulation specific format, with support of external binary file management and data staging.

SNAP current limitations

- simulations of moving objects in 3D real space (x,y,z)
- => simulation outputs organized as snapshots (at various times)
- physical quantities are sampled on a 3D (x,y,z) space (irregularly like particles, adaptive meshes or regularly like cartesian meshes)
- direct output of the simulation code or post-processed data with same structure (particle/mesh)

SNAP working draft

1. **Search for available simulations and data** (Simulation Discovery, section 3)
   The query is on metadata. The result is an XML document (maybe VOTable) with matching result metadata. [NOT YET WRITTEN]

2. **Identification of subset of interest** (Subset Selection, Section 4)
   The user identifies a subset of the full simulation data which is of interest, which can used to select easily the region to focus on. This subset is defined both in time and in space. [NOT YET WRITTEN]

3. **Snap request** (Section 5)
   Send to the server the selection parameters for the Snap action

4. **Data staging and delivery** (Section 6 and 7)
   Metadata are immediately delivered to the client as a VOTable
   Data are delivered (possibly after some time, needed for extraction) via HTTP, FTP as binary files.
   Delivery of VOTable and binary data files can be in two separated stages

SNAP compliance chart

1. **The SNAP service MUST be registered.**
   Registration allows clients to use a central registry service to locate compliant simulation access services and select an optimal subset of services to query, based on the characteristics of each service and the simulation data collections it serves.

2. **The service MUST support a Simulation Discovery service. [TBD, depends strongly on datamodel]**
   Available data can be retuned as the result of a query based on a series of physical and technical parameters which specify the requirement of the user. These parameters can be general or specific of the discipline or research field of interest. The research can be further refined or used to select the dataset of interest and proceed with following steps of the SNAP procedure.

SNAP compliance

- **Sub-Volume/Sub-Set determination methods MAY/ SHOULD be supported [TBD]**
  Contrary to real-sky-oriented protocols, there is in most simulations no natural sub volume that a user might be *a priori* interested in. The SNAP service SHOULD provide methods to perform the selection of a simulate volume.
  Services which has only download capabilities MAY provide this fuctionality.
  Services which support data cutout MUST provide such function.

- **Sub-Volume Extraction method MAY be supported**
  This method allows clients to retrieve data from a spatially defined sub-volume of the simulation box. The client determines the rectangular or spherical region within the simulation, the bounds and scale (i.e. units) of which are specified in the simulation metadata, and the service returns the simulation data contained within this region.

- **The Snapshot Retrieval (getSnap) web method MUST be supported.**
  This method allows clients to retrieve single simulation snapshots and cutouts

SNAP compliance

- **The service MAY use a *staging* method to return the particle file** extracting a sub-sample of particles or grid points from a larger simulation box is likely to be a time-consuming process and would thus require some kind of caching.

- **A job status request MAY be supported.**
  This method allows users to inquire about the status of a staged request. It MAY allow a user to cancel a request.

SNAP request

- input parameters:
  - region of interest (in physical units or fraction of the box)
    - POS=0.3,0.2,0.1 [NULL == 0.5,0.5,0.5]
    - SIZE=0.2,0.3,0.2 or SIZE=0.2 [NULL == whole box]
    - [SHAPE=BOX or SPHERE]
    - BOUNDARY=TRUNC or PERIODIC
  - field of interest
    - FIELDS [by default return all fields, i.e. all physical quantities]
  - file format
    - FORMAT=data/votable (default), data/hdf5, data/fits etc.
  - [table verbosity
    - VERB==integer] inherited from SIAP but could be removed to be closer to SSA
  - service defined parameter MAY be supported by the service

SNAP successful output

```
<RESOURCE name=myParticles type="results">
  <DESCRIPTION>Velocity-Position from N-Body run</DESCRIPTION>
  <INFO name="QUERY_STATUS" value="OK"/>
  <TABLE name="Particles" ID="NBody" >
    <FIELD name="x" ID="x1" ucd="pos.cartesian;pos.cartesian.x“ datatype="float" unit="Mpc" />
    <FIELD name="y" ID="y1" ucd="pos.cartesian;pos.cartesian.y“ datatype="float" unit="Mpc" />
    <FIELD name="z" ID="z1" ucd="pos.cartesian;pos.cartesian.z“ datatype="float" unit="Mpc" />
    <FIELD name="vx" ID="vx1" ucd="phys.veloc;pos.cartesian.x“ datatype="float" unit="km/s“ />
    <FIELD name="vy" ID="vy1" ucd="phys.veloc;pos.cartesian.y“ datatype="float"  unit="km/s" />
    <FIELD name="vz" ID="vz1" ucd="phys.veloc;pos.cartesian.z“ datatype="float"  unit="km/s" />
    <DATA>
      <BINARY>
        <STREAM href="file:///scratch/myhome/test.bin"/>
      </BINARY>
    </DATA>
  </TABLE>
 </RESOURCE>
</VOTABLE>
```

SNAP successful output

VOTable for the velocity field of a fluid on a fixed 3D mesh

```
<RESOURCE name="myVectorField" type="results" >
  <DESCRIPTION>Velocity Field from N-Body run</DESCRIPTION>
  <INFO name="QUERY_STATUS" value="OK"/>
  <TABLE name="VelocityField" ID="Vel" order="sequential">
    <FIELD name="vx" ID="vx1" ucd="phys.veloc;pos.cartesian.x" datatype="float" arraysize="41x41x41"
     unit="km/s" geometry="mesh" />
    <FIELD name="vy" ID="vy1" ucd="phys.veloc;pos.cartesian.y" datatype="float" arraysize="41x41x41"
     unit="km/s" geometry="mesh" />
    <FIELD name="vz" ID="vz1" ucd="phys.veloc;pos.cartesian.z" datatype="float" arraysize="41x41x41"
     unit="km/s" geometry="mesh" />
    <DATA>
     <BINARY>
      <STREAM href="file:///scratch/myhome/test.bin"/>
     </BINARY>
    </DATA>
  </TABLE>
 </RESOURCE>
</VOTABLE>
```

- We still need a proper way of indicating what the spatial dimensions are for a representation like this. FITS has its WCS system for implicitly specifying the spatial coordinates of a multidimensional array. Is something like WCS (FITS) for VOTable ?

SNAP data staging

- Data staging == the process the server performs to retrieve or generate the requested simulation (sub-)volume from a similar box and cache them in online storage for retrieval by a client.
- necessary for:
  - large archives which must retrieve simulation data from hierarchical storage,
  - services which can dynamically extract subvolumes, where it may take a substantial time (e.g., minutes or hours) to retrieve the particles in the relevant region of the simulation box.
  - Issuing a staging request for a set of simulation subvolumes (e.g. for a set of small cubes randomly placed in a simulation box) also permits large servers to optimize subvolume extraction, for example to take advantage of parallelization for large requests.

SNAP data staging

- data is staged on the server for later retrieval by the client
- the data is only staged for a period of time and is eventually deleted by the service. This therefore permits the getSnap method to be identical whether or not staging is used. The service can proceed to generate the simulation sub-volume regardless of the state or accessibility of the client.
- The service provider can decide how to deal with multiple requests from the same user or large number of requests at the same time from different users.
  - Queue, batches, limits are defined by the provider.
  - The provider is only requested to publish and notify such features to the registry service (details to be defined GWS working group ?).

SNAP data staging

- As soon as staged data are available at the given URL, the user can start the download procedure. The user can be informed of the availability of the data following two different approaches:
  - The client searches for information on the service. Requires protocol. the client reload the data URL to check if data are available. Preferred solution up to now.
  - The service searches for the client and, if present, sends information to it. Requires authentication (unique id by user even in different session). Messaging capability.

SNAP data delivery

- The snapshot retrieval request (getSnap web method) allows a client to retrieve a single raw simulation file given an access reference or "acref" as returned by a prior simulation query.
- All the metainformation about the content and the structure of the data file are stored in the associated VOTable.
- The retrieved data file is in a binary format (hdf5, fits, raw tabulated data etc.)
- Input
  - The input to the getSnap web method is the simulation acref for the indicated raw simulation data or extract subvolume. The acref for a particular file is obtained through a prior call to the Simulation Query web method.
- Successful Output
  - The output of getSnap MUST be a single data file returned with a MIME-type identifying the file format. The primary type of the MIME code should be "data/". Other MIME types are not expected.
- Error Response
  - If a condition is encountered that prevents the requested image from being downloaded, the output MUST be a VOTable with a single RESOURCE element containing an INFO child with name="QUERY_STATUS". The allowed values for this INFO are the same as those defined for the Image Query; in addition, the following additional attribute MAY be supported: **DEFERRED**
    This indicates that the requested image is not yet available for some reason but that it will be at some time in the future. Clients that receive this type of message can periodically try (poll) the given acref URL until the image becomes available.