Virtualisation of Simple Scientific Data Objects

Stephen Rankin
CCLRC – CASPAR- DCC
s.e.rankin@rl.ac.uk
David Giaretta, Steve Crothers, Brian McIlwrath, Matt Dunckley

iPRES 2006

Contents

- Introduction to CASPAR.
- Aim
- OAIS Representation Information.
- Data Description Languages (EAST).
- Data Semantic Descriptions (DEDSL).
- Simple Objects (Table Example).
- Virtualisation of Simple Scientific Data Objects.
- Demonstration Example.
- Conclusions and Future Work.

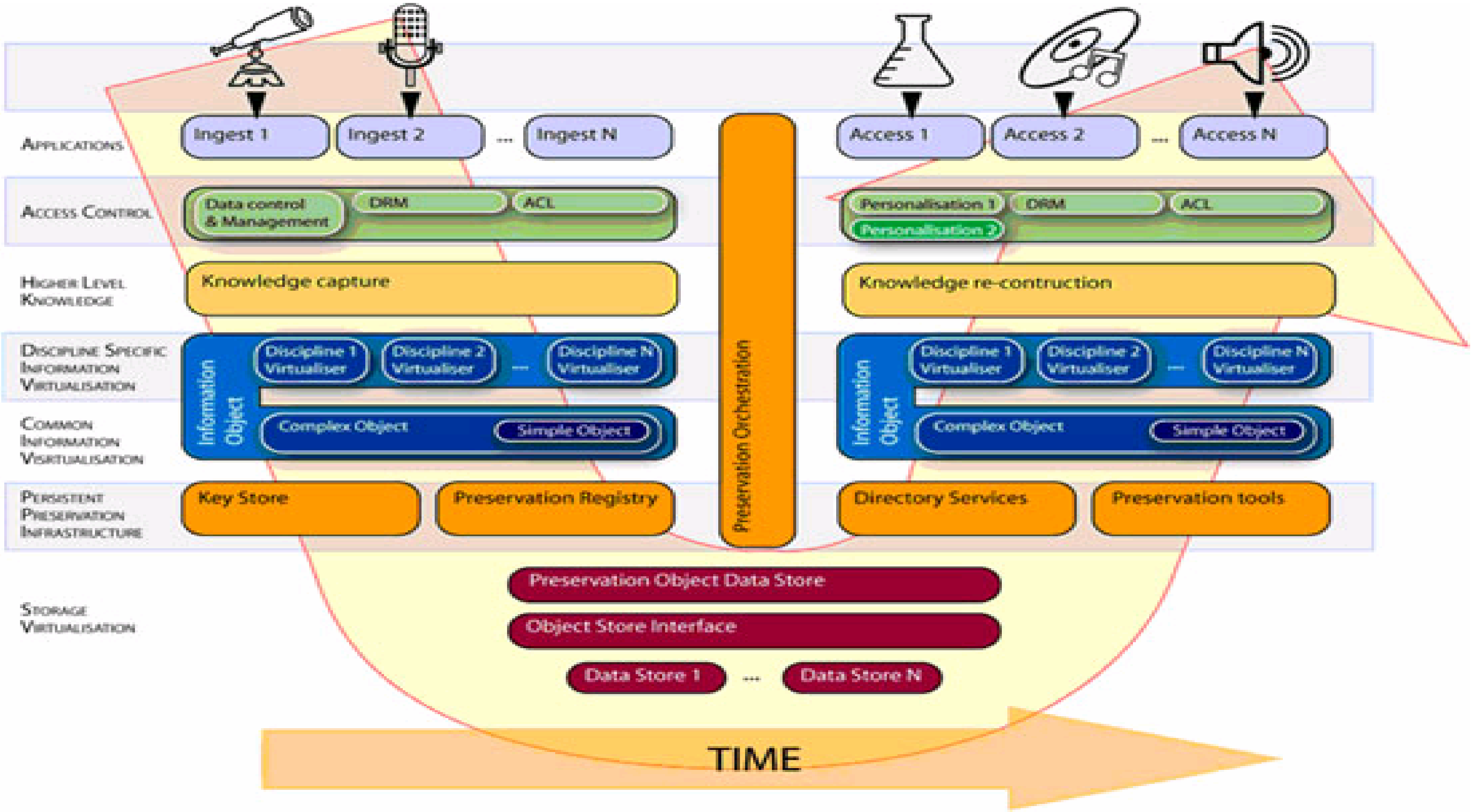
Aim

- To show that OAIS representation information can used to automate the reading and rendering of scientific data and help a future scientist to reuse data.
- To validate the concept of OAIS representation information.

CASPAR – Cultural, Artistic and Scientific Knowledge for Preservation, Access and Retrieval

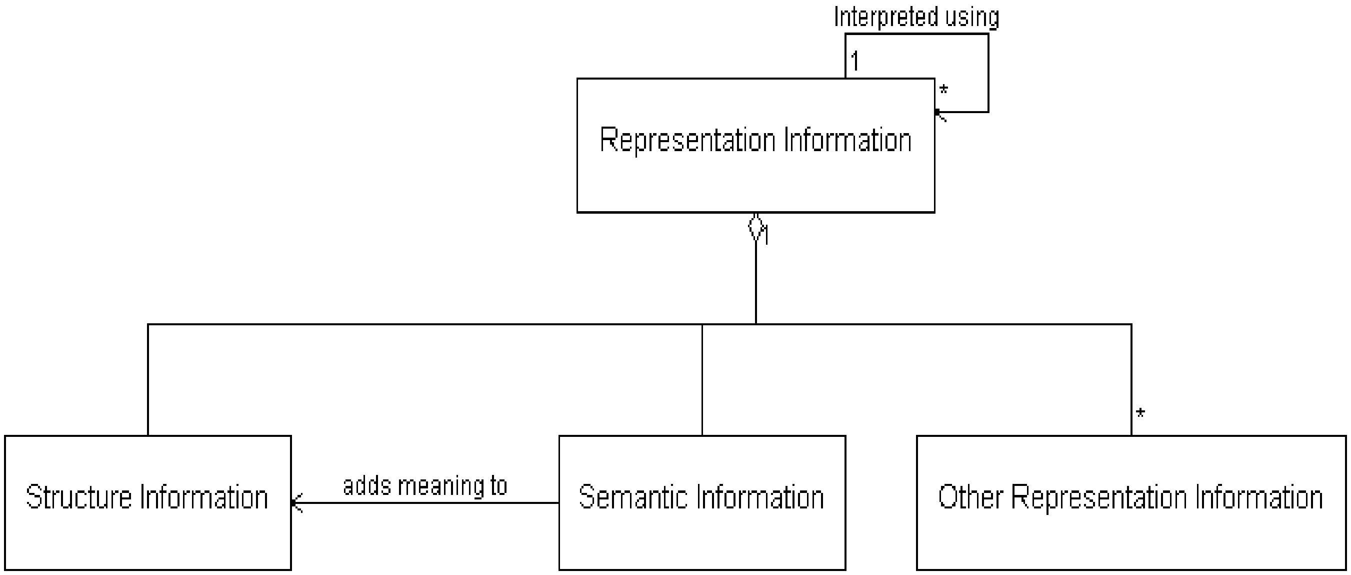
- **CASPAR, a new EU FP6 Integrated Project.**
- The ambitious goal is to build up a common preservation framework for heterogeneous data, along with a variety of innovative applications.
- The Reference Model for an Open Archival Information System (OAIS, ISO 14721) forms the basis of CASPAR.
- http://www.casparpreserves.eu/
- CASPAR consortium (CCLRC - the lead partner and ESA), cultural (UNESCO) and creative expertise (INA, CNRS, University of Leeds, IRCAM and CIANT). Commercial partners (ACS, ASemantics, MetaWare, Engineering, and IBM/Haifa), experts in knowledge engineering (CNR and FORTH) and other leaders in the field of information preservation (University of Glasgow and University of Urbino).
- Publication - http://www.ercim.org/publication/Ercim_News/enw66/giaretta.html

CASPAR Virtualisation Model.



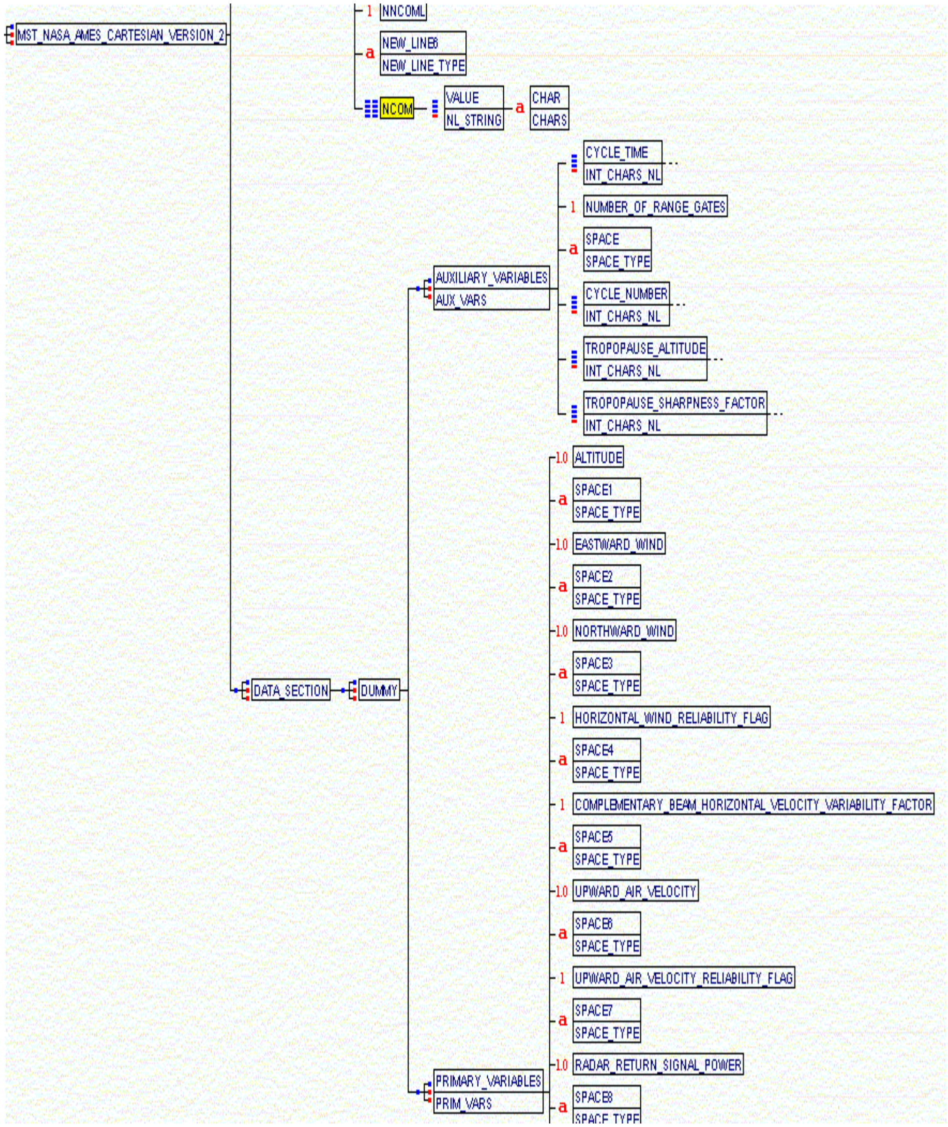
OAIS Representation Information

- Representation Information Registry Repository at http://registry.dcc.ac.uk.
- API for RepInfo registry at http://cvsweb.dcc.rl.ac.uk (registry)
- Tools being developed.



EAST Logical Structure

- Hierarchical Data Description.
- Structure Types – Record, Array, Repeat Variable (List) and Enumeration.
- Value Types – Integer, Real, String, Character.
- Access Paths - MST_NASA_AMES_CARTESIAN_VERSION_2.DATA_SECTION.DUMMY.PRIMARY_VARIABLES.ALTITUDE



EAST Physical Structure (the bits)

- EAST can describe data structures at the bit level in a very general way.
- Allows you to define the bit structure of a Real, Integer, Enumeration – this includes octet order (byte order) and array storage.
- Can do conditional structures and restrictions – potential for data validation and identification - authenticity?
- Can not do everything that I need – no pointers or simple expressions that are required for more complex file formats.

Data Entity Dictionary Specification Language (DEDSL)

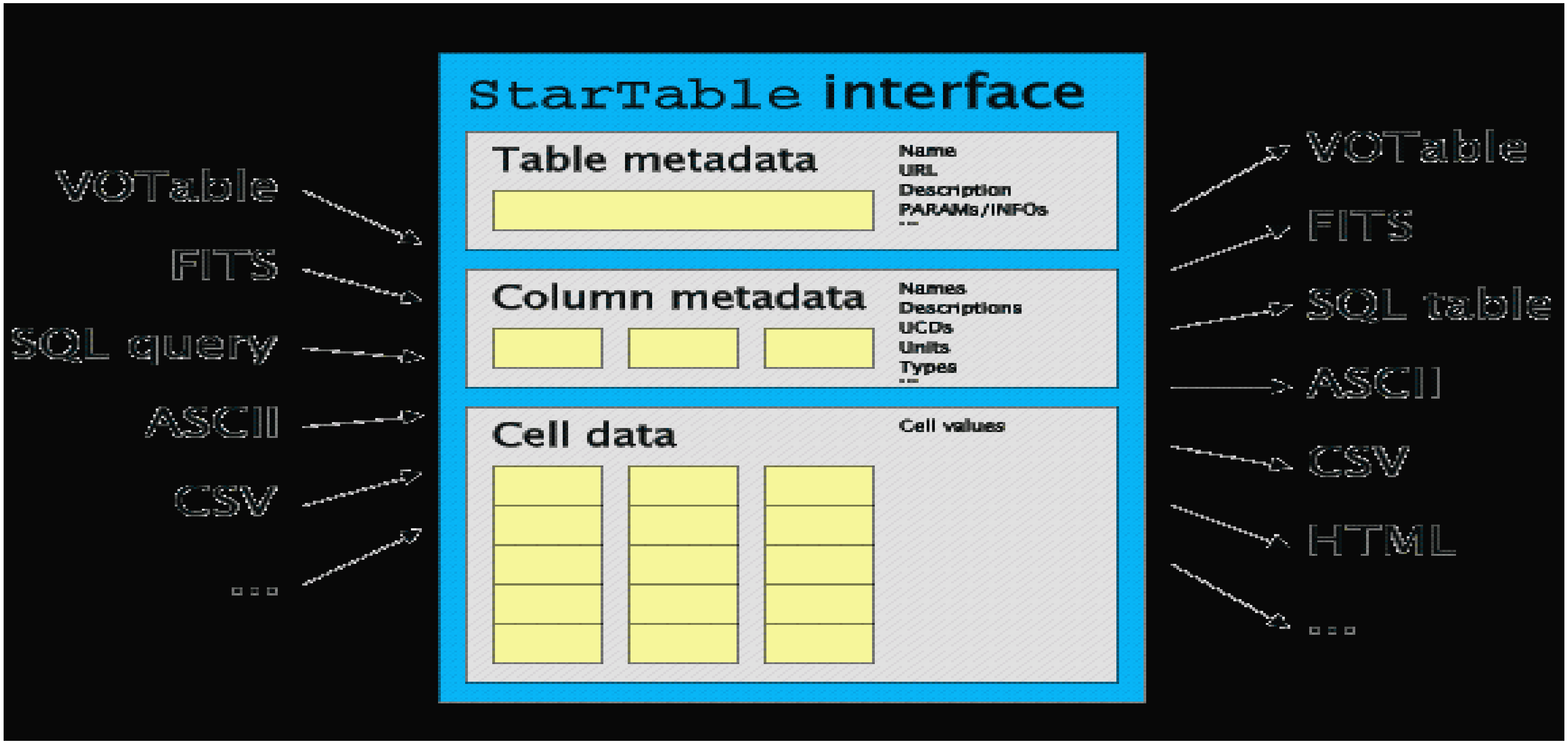
- Abstract, PVL, and XML(DTD) syntax for defining some simple data semantics.
- Only a small number of required attributes for a given data structure, NAME, DEFINITION, UNITS (conditional), *ENTITY*_TYPE (conditional), ENUMERATION_VALUES (conditional), TEXT_SIZE (conditional).
- You can define your own attributes.
- You can reuse definitions from other dictionaries.
- **Link the data structures to the semantics via the EAST access path, i.e. define a new attribute – EAST_PATH (OASIS tool does this).**

EAST and DEDSL Tools

- CNES EAST tools (http://east.cnes.f), OASIS, EAST C Library (reference implementation).
- Also DEBAT (BEST Tools) http://debat.c-s.fr/
- JNI Wrapper for EAST C Library in our CVS repository (jnieast) http://cvsweb.dcc.rl.ac.uk.
- Interfaces for a more general data description language and semantics API in our CVS (DSSIL).

A Simple Object Example (Table Data)

- Using an existing table object definition (STIL, http://www.starlink.ac.uk/stil) – Mark Taylor)



Additional Metadata Required for Virtualisation (Access)

- The structures and the semantics are linked via the access path (pointer to structure and metadata).
- Add additional attributes to say if a structure is a TABLE, COLUMN, ROW or an individual VALUE.
- Currently I take the NAME, DEFINITION, UNITS and *ENTITY*_TYPE in the DEDSL description to populate the table object metadata.
- There are only so many possible ways of describing a table, column or row within an EAST description:
- COLUMN = 1D ARRAY
- COLUMN = List of VALUES
- ROW = RECORD
- TABLE = nD ARRAY
- TABLE = List of RECORDS
- TABLE = ARRAY of RECORDS
- Etc…

Additional Metadata Required for Virtualisation (Rendering)

- Knowing which table columns are useful to plot against one another is important information.
- Many types of plot – scatter, line etc – domain specific.
- An ontology for plots?
- The plot metadata needs to be kept with the other semantic information?

Demonstration Application

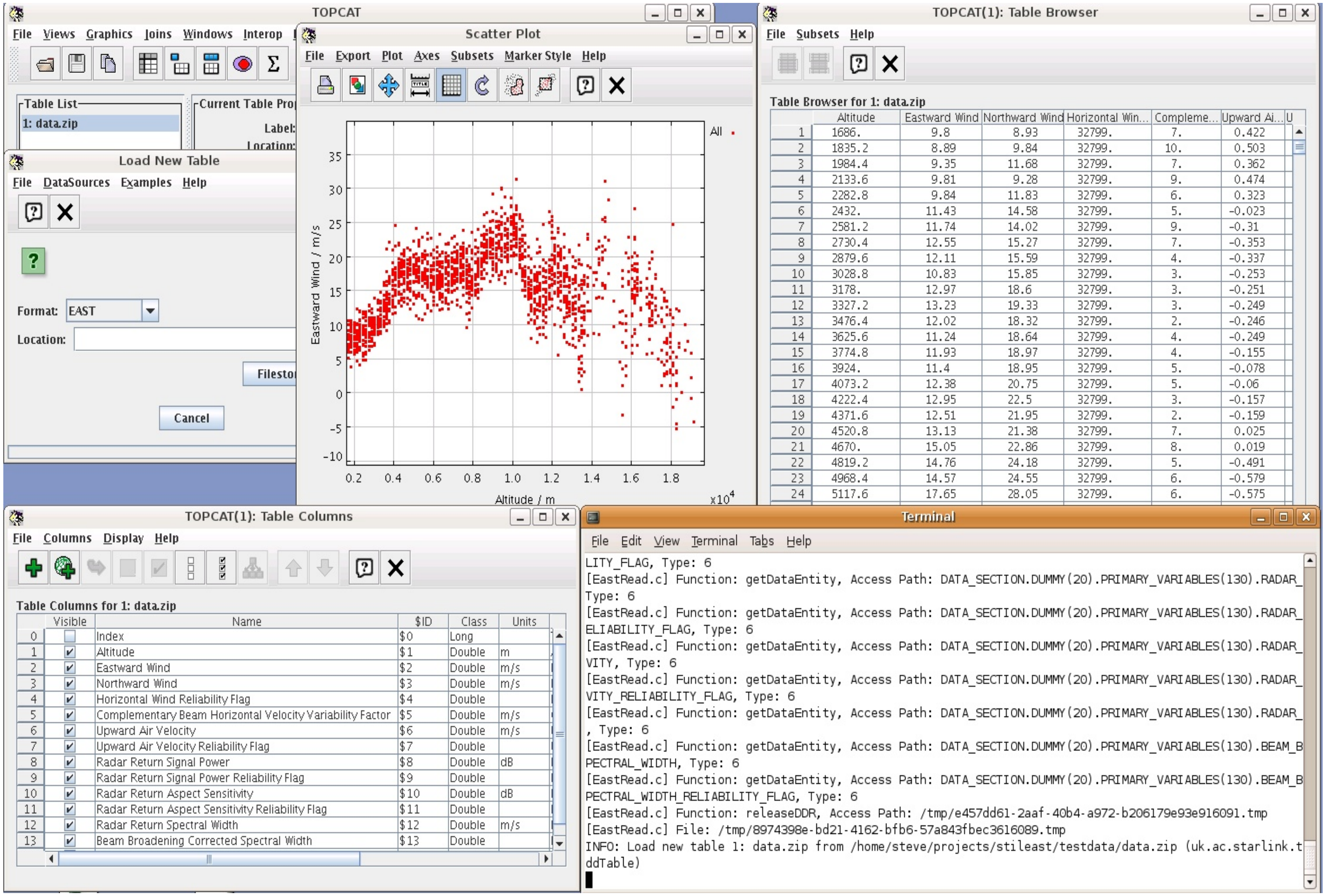


Table List: 1: data.zip

Label:

Location:

Format: EAST

Location:

Table Browser for 1: data.zip

	Altitude	Eastward Wind	Northward Wind	Horizontal Win...	Compleme...	Upward Ai...	U
1	1686.	9.8	8.93	32799.	7.	0.422	
2	1835.2	8.89	9.84	32799.	10.	0.503	
3	1984.4	9.35	11.68	32799.	7.	0.362	
4	2133.6	9.81	9.28	32799.	9.	0.474	
5	2282.8	9.84	11.83	32799.	6.	0.323	
6	2432.	11.43	14.58	32799.	5.	-0.023	
7	2581.2	11.74	14.02	32799.	9.	-0.31	
8	2730.4	12.55	15.27	32799.	7.	-0.353	
9	2879.6	12.11	15.59	32799.	4.	-0.337	
10	3028.8	10.83	15.85	32799.	3.	-0.253	
11	3178.	12.97	18.6	32799.	3.	-0.251	
12	3327.2	13.23	19.33	32799.	3.	-0.249	
13	3476.4	12.02	18.32	32799.	2.	-0.246	
14	3625.6	11.24	18.64	32799.	4.	-0.249	
15	3774.8	11.93	18.97	32799.	4.	-0.155	
16	3924.	11.4	18.95	32799.	5.	-0.078	
17	4073.2	12.38	20.75	32799.	5.	-0.06	
18	4222.4	12.95	22.5	32799.	3.	-0.157	
19	4371.6	12.51	21.95	32799.	2.	-0.159	
20	4520.8	13.13	21.38	32799.	7.	0.025	
21	4670.	15.05	22.86	32799.	8.	0.019	
22	4819.2	14.76	24.18	32799.	5.	-0.491	
23	4968.4	14.57	24.55	32799.	6.	-0.579	
24	5117.6	17.65	28.05	32799.	6.	-0.575	

Table Columns for 1: data.zip

	Visible	Name	$ID	Class	Units
0	☐	Index	$0	Long	
1	☑	Altitude	$1	Double	m
2	☑	Eastward Wind	$2	Double	m/s
3	☑	Northward Wind	$3	Double	m/s
4	☑	Horizontal Wind Reliability Flag	$4	Double	
5	☑	Complementary Beam Horizontal Velocity Variability Factor	$5	Double	m/s
6	☑	Upward Air Velocity	$6	Double	m/s
7	☑	Upward Air Velocity Reliability Flag	$7	Double	
8	☑	Radar Return Signal Power	$8	Double	dB
9	☑	Radar Return Signal Power Reliability Flag	$9	Double	
10	☑	Radar Return Aspect Sensitivity	$10	Double	dB
11	☑	Radar Return Aspect Sensitivity Reliability Flag	$11	Double	
12	☑	Radar Return Spectral Width	$12	Double	m/s
13	☑	Beam Broadening Corrected Spectral Width	$13	Double	

```
LITY_FLAG, Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).RADAR_
Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).RADAR_
ELIABILITY_FLAG, Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).RADAR_
VITY, Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).RADAR_
VITY_RELIABILITY_FLAG, Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).RADAR_
, Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).BEAM_B
PECTRAL_WIDTH, Type: 6
[EastRead.c] Function: getDataEntity, Access Path: DATA_SECTION.DUMMY(20).PRIMARY_VARIABLES(130).BEAM_B
PECTRAL_WIDTH_RELIABILITY_FLAG, Type: 6
[EastRead.c] Function: releaseDDR, Access Path: /tmp/e457dd61-2aaf-40b4-a972-b206179e93e916091.tmp
[EastRead.c] File: /tmp/8974398e-bd21-4162-bfb6-57a843fbec3616089.tmp
INFO: Load new table 1: data.zip from /home/steve/projects/stileast/testdata/data.zip (uk.ac.starlink.t
ddTable)
```

Conclusions Future Work

- It does look possible to use structure and semantic descriptions to virtualise simple scientific data objects, if the correct metadata is defined for the object.
- Need more object descriptions and to create a simple object API.
- Need to think about plots, and define them.
- Future – extend EAST to include pointers etc.
- Future – Other structure and semantic descriptions, DFDL?