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 be 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).
CASPAR Virtualisation Model.
OAIS Representation Information

- Representation Information Registry Repository at [http://registry.dcc.ac.uk](http://registry.dcc.ac.uk).
- API for RepInfo registry at [http://cvsweb.dcc.rl.ac.uk](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
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?