Scientific Data Management for Visualization Implementation Experience

Mario Valle, Jean Favre [*]
Swiss National Supercomputing Centre

Etienne Parkinson [†]
VA TECH HYDRO

Alexandre Perrig, Mohamed Farhat [⊗]
EPF Lausanne LMH

Abstract

In the validation phase of a commercial CFD code, the importance of scientific data management for visualization and knowledge preservation surfaced. The proposed Scientific Data Bag (SDB) solution offers a lightweight method to record metadata and relationship between related datasets in a file structure preserving access to data files in their native format. An interface layer provides services to post-processing tools enabling uniform access to the project datasets.

1 Motivation

Our Turbine Simulation project [Ale04] aims at the development of an advanced method for flow simulation in Pelton turbines. Water jets impinging the runner buckets ensure the energy transfer in the turbine, leading to complex characteristics of the flow, i.e. 3D turbulent, two phases and unsteady. This project brings together academic (EPF Lausanne Laboratory for Hydraulic Machines and CSCS) and industrial partners (VA TECH HYDRO) that have a continuous development process on the adaptation and validation of a CFD commercial code to perform such a flow simulation.

Figure 1: Real and simulated Pelton turbine

The soundness of the assumptions made during the modeling phase is assessed validating simulation against experimental results. The project is specifically dedicated to the development of an innovative technique aimed to visualize the jet impingement and its complex interaction with the runner buckets and automating the data management and comparison of experiment and simulation outcomes. The result of the project will definitely allow the industrial partner to improve significantly the design process of Pelton turbines.

[*] Swiss National Supercomputing Centre (CSCS) – Via Cantonale Galleria 2 – CH-6928 Manno (Switzerland)
[†] VA TECH HYDRO SA – Rue des Deux Gares 6 – CH-1800 Vevey (Switzerland)
[⊗] Laboratory for Hydraulic Machines - Swiss Federal Institute of Technology - Av. de Cour 33Bis CH-1007 Lausanne (Switzerland)

1.1 Data management for visualization

Visualization is the process of presenting data in a form that allows findings and rapid understanding of relationships that are not readily evident from raw data. However, scientific findings are only as good as the information they are based upon. Data management techniques have the goal of ensuring data quality and secure access to the physical storage of the data itself. Scientific data management applies those techniques to support computational research and increase confidence in the research outcomes. It is becoming increasingly important as formerly disparate subfields in one scientific domain start to integrate, and diverse datasets must be combined for visual exploration and annotation. Moreover, visualization methods can produce additional, derived data (e.g., isosurfaces extracted from a 3D scalar field) that one may want to store, compress, annotate, and manage [Ham03].

Scientists usually tend to consider raw data bytes as the only real scientific data. Nevertheless, knowledge discovery, either by visualization or by data mining, requires contextual information that is seldom considered data: relationship between datasets, simulation parameters, experimental conditions, etc.

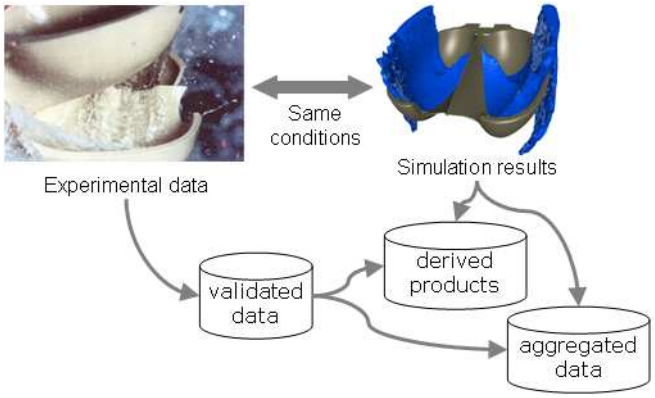

Figure 2: Data management domain for the Turbine Simulation project

1.2 The Turbine Simulation project requirements

In the framework of this project, each simulation run goes together with the corresponding validating experiment. This arrangement involves related and potentially heterogeneous datasets (CFD simulation results, experimental data, images, parameters and researcher annotations) that should be managed as a unit. Other related datasets are created during the analysis phase when derived datasets are produced (e.g. torque computation from CFD data). Those datasets should be stored and managed in the same manner and together with the originating data.

To support future improvements of the turbine design process the knowledge gained during the current project has to be preserved together with the data. This knowledge is mainly in the form of contextual information needed to interpret the simulation and experiment outcomes, like measurement units and experimental conditions. Free form

annotations, related documents or CAD models are another set of unstructured knowledge to be preserved together with the data itself.
To increase effectiveness, researchers want to minimize the number of different kind of interfaces to visualization and post-processing tools they have to master.
From the project point of view, it is critical that any data management proposal provides a smooth transition from the existing working methods without disrupting the current project workflow and without diverting too many resources from the computation and validation project goals.

1.3 Previous work

There is not much work on scientific data management issues related to similar project requirements. Existing studies mostly focus on proposing file formats for scientific data or on classical storage management and access techniques.
Instead, in the area of digital libraries, there are proposals of description methods for collections of heterogeneous digital artifacts, like scanned books with related images and sound recordings. One of them is the METS standard [Met04] for encoding descriptive, administrative, and structural metadata about objects within a digital library expressed using the XML schema language [Xsc01]. The METS standard inspired important ideas: the effectiveness of uniform metadata storage using an accepted standard format, the structural map construct that describes the collection internal composition, the access behaviors stored together with each resource. Unfortunately, METS cannot be applied as-is to the project due to the different kind of artifacts described.
On the software side, the XPackage [Xpa03] proposal tries to describe grouping of software components, like libraries and program files, but it is too limited to describe more complex relationships.
General-purpose file formats, like HDF5 [Hdf04], NetCDF [Net04] and XML-based formats (see [Val04] for a list), provide support for user metadata storage. They are a good choice for storing new data, but require the conversion of the various project dataset types into their own format creating storage and work duplication. To overcome this limitation the idea provided by XDMF [Cla01] has been adopted. This XML-based format distinguishes between heavy (raw) and light (description) data retaining the former in their native formats and storing only a reference to the unmodified native dataset.
The XML standard format [Xml04] provides a good storage format for structural data and metadata. The benefits are a strict syntax, tool availability, the built-in extensibility support and, as in METS, the uniform access to data types specific metadata that otherwise has to be extracted from heterogeneous datasets.
The “Semantic Web” [Swe01] effort introduces a uniform method to describe digital resources and their relationships: the Resource Description Framework (RDF) [Rdf04]. Its only drawback is the complexity compared to the requirements of our Turbine Simulation project. For this reason, only some RDF ideas are kept, delaying a more complete integration when interoperability and external access issues will be of concern.

2 The Scientific Data Bag

The solution proposed to address the project requirements is a lightweight method for adding metadata to a set of scientific datasets, for collecting derived data and for unifying access for data visualization with existing project tools. This collection of methods and tools is dubbed Scientific Data Bag (SDB).

The Scientific Data Bag metaphor recalls the fact that a bag usually contains disparate, but related things. An example of related datasets to be collected in the same Bag could be simulation results, corresponding experimental data, metadata describing the above data, free form annotations, etc. A Bag is a container, not a data format: it only adds semantic information to describe heterogeneous data files, their relationship and related metadata.

The Scientific Data Bag does not offer a complete data management environment, but aims to provide a quick answer to the project requirements (see 1.2), even through this means a more limited scope solution. Moreover, the goal is neither to reinvent existing data formats and tools nor to provide interoperability outside our Turbine Simulation project.

To answer all requirements of the project, the Scientific Data Bag provides: (1) a logical description of items and collections; (2) an implementation layer (API) that provides a uniform access method for post-processing tools; (3) an operational workflow supported by various clients sitting on top of the SDB API.

2.1 The logical structure

To be a useful support for project data management, SDB should provide a logical model that could describe in a single coherent structure the various datasets, operations and relationships involved.

Metadata for this project are data about data used for cataloging purposes, to provide an interpretative context for the data, to record extra information about data, for storage and physical administration of the dataset. Metadata are declarative (atomic data, like author name or a simulation parameter) or relational (record relations between items like the torque dataset *is-derived-from* the pressure simulation result).

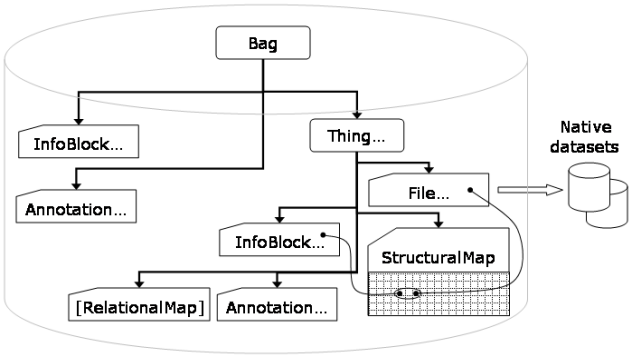

Figure 3: Bag file internal structure

A Bag expresses the fact that there are entities that relates to each other (part of the same simulation for example). Those entities are called Things. Bag and Things contain declarative metadata grouped in various types of InfoBlocks and free form Annotations. Each InfoBlock contains different types of related metadata, but all have the same structure, thereby not forcing the adoption of a specific set of metadata. It is expected that the initial list of metadata will grow based on usage experience as shown below. An external profile records the implemented metadata list to support simple future extensibility of the SDB logical structure.
Only the Bag contains support for general services offered to the contained Things, like relative path resolution or unique identifiers generation. Unique to Things are File blocks, StructuralMap and optional RelationalMap blocks. The File blocks point to the physical datasets that are part of the Thing and provide information specific for file access, like a HDF5 dataset access path or an ASCII file column names list.
What defines the real structure of a Thing is the StructuralMap block. It describes the internal structure of the Thing in term of a hierarchy of parts. The parts are either File or InfoBlocks. The StructuralMap is the basis for building the user interface selector for part visualization. An example is a PhotoAlbum Thing: various File blocks point to the image files; each of them has an associated InfoBlock that records its metadata. The StructuralMap transforms this set of unrelated images and metadata into a coherent collection: it states that various Files compose the Thing and that each File has an associated InfoBlock.
Things may contain also a RelationalMap that records the set of other, non-structural relationships like *derived-from* or *based-on*.

2.2 The operational workflow

Looking to some SDB usage scenarios helps understanding how it can support data management inside projects with requirements similar to those of our Turbine Simulation project. Besides this, the kind of planned software clients requesting SDB services determines its interface structure. The main four usage scenarios are the following.
1. Extraction. Extractor scripts specific to one kind of data format adds datasets to a newly generated Bag. This kind of metadata harvesting [Oai02, How04] automates as much as possible the metadata recording phase and collects in one point format specific processing. Examples from the Turbine Simulation project are the metadata extractor scripts for CFX result files (based on the ANSYS CFX meshexport library) and for images taken with digital cameras that record EXIF [Exi02] format metadata.
2. Bag editing and amendment. Users then add specific metadata to a Bag or contained Thing with a Bag editor. Within the editor, they can merge more than one Bag to add relationships between the Things contained in each of them.
3. Browsing or retrieval phase. The actions performed are: (1) the users select one specific Bag file by browsing a project directory or, in the future, by querying a database. (2) For each contained Thing, and for the Bag itself, they can look at their InfoBlocks and corresponding Annotations to decide if the selected Bag is the correct one and to select Things for visualization or post-processing. (3) A further level of selection is presented for Things that have an internal structure. As an example, for a Bag containing simulation results and the corresponding experimental runs the first selection level selects between

simulation and experiment, the second level selects the timestep, zone and part from the simulation, or the test run and sensor timeseries for the experiment. This selection process simply assembles a "selection packet" for the post-processing tool with all the information needed to access the native datasets. Then the tool itself should provide decoding for the "selection packet" and readers for the specific data type selected.
4. Post-Processing. During analysis, users can generate derived data, like filtered timeseries from an experiment or torque computation from the simulation results. This derived data can be added to the Bag together with a reference to the originating Thing dataset.

2.3 The implementation layer

A Bag file is implemented as a small XML file stored together with the native data files that describes. The Bag file is created, accessed and modified only through the SDB library interface layer (API).
To maintain flexibility in the definition and usage of the SDB layer, two things surfaced during development: from the syntax point of view, the required format structures must be kept at a minimum; and from the semantic point of view, the knowledge of the format semantic and detailed content definition should be moved to an external schema or profile. Initially a very detailed structure was put in the SDB interface, leading to the "fat & arthritic" API syndrome: an interface that exposes one and only one possible structure modeled on the file format forces reality to adapt to the proposed structure and not vice versa.
The SDB API exposes only the required structure of the logical schema as a set of C++ classes. The description of the legal types of those structures and their content is accessed at runtime from the external profile. This profile contains also information on the presentation of the Bag data, like numeric formats and labels, to support external clients. The approach adopted by SDB is more liberal than the one taken by Castor [Cas04] and SchemaWizard [Bal04]; they extract semantic and presentation from the XSchema [Xsc01] that describes the underlying XML file structure.
To facilitate testing and coding of client applications, like browsers and editors, the API is encapsulated also as a Tcl scripting language extension.
Together with the SDB API comes a library of support routines for various common operations, like comparison of values on corresponding meshes and HDF5 file writers for derived data storage.
The readers for the native dataset types are integrated inside AVS/Express [Avs96] and ParaView [Par04] visualization tools; only a module that accepts and transforms the "selection packets" into native visualization command has been added.

3 Discussion

For SDB development, a rapid prototyping process has been followed to help project acceptance. The functionalities have been focused on three kinds of data: the ANSYS CFX simulation result file, the current EPFL-LMH experimental data recording format and JPEG images tagged with EXIF metadata by a digital camera. The parallel

development of extraction scripts for those files types and a simple browser uncovered missing functionalities in the interface layer and tuned the profile definition.
Using SDB on real project datasets highlighted the need of better interfacing with the post-processing tool, for example using an intermediate broker or, like METS does, using an interface similar to Web Services [Wsd01].
Besides implementation, the most important result of the SDB project is the insight gained on the importance of scientific data management in support of visualization and analysis and on the wealth of usable information hidden in metadata and relationships.
The SDB code is available from the authors. It is written in portable C++ and Tcl with clients implemented in Tcl/Tk. The code runs on Linux and Windows platforms.

4 Future work

Once the collection and creation of Bag files for all the project data will be complete, the SDB development will continue along two main lines. (1) Loading of Bag files into a database to allow different kind of unanticipated queries for true data mining support. (2) True extensibility support, to free the researcher from the need to conform to the set of metadata types anticipated by one version of the profile. More long-term goal is to investigate thoroughly the RDF framework to make the Bag file format interoperable with unanticipated applications.
The only serious limitation of the SDB approach so far is the needed distinction between the bag file and corresponding data files. Users expressed the desire to have only one file around without running the risk of loosing positional relationships between files.

Acknowledgements

The authors acknowledge the support of the Swiss Commission for Technology and Innovation via project n. 6139.1. The strong support of the ANSYS team in handling the CFX software is also acknowledged.

References

[Ale04] A. Perrig, M. Farhat, F. Avellan, A. Parkinson, H. Garcin, C. Bissel, M. Valle, J. M. Favre: *Numerical Flow Analysis in a Pelton Turbine Bucket*, Proceedings of the 22nd IAHR Symposium on Hydraulic Machinery and Systems, Stockholm, Sweden, June 29 – July 2, 2004

[Avs96] *AVS/Express Developer Edition Reference Manual*, Advanced Visual Systems Inc., Waltham Mass., June 1996 <http://www.avs.com/>

[Bal04] O. Balsoy, Y. Jin, G. Aydin, M. Pierce, G. Fox: *Automating Metadata Web Service Deployment for Problem Solving Environments.* To be published in FGCS 2004.
<http://ptlportal.communitygrids.iu.edu/schemawizard/paper1.html>

[Cas04] *The Castor Project*, July 2004 <http://www.castor.org/>

[Cla01] J. A. Clarke, R. Namburu: *The eXtensible Data Model and Format for Inter-disciplinary Computing*, September 2001 <http://www.hpcmo.hpc.mil/Htdocs/UGC/UGC01/paper/jerry_clarke_paper.pdf>

[Exi02] *Exchangeable image file format for digital still cameras*: EXIF Version 2.2 – Japan Electronics and Information Technology Industries Association, April 2002 <http://www.exif.org/>

[Ham03] B. Hamann, E. W. Bethel, H. Simon, J. Meza: *NERSC Visualization Greenbook – Future Visualization Needs of the DOE Computational Science Community*. The International Journal of High Performance Computing Applications, Volume 17, Number 2, Summer 2003, pp 97-124. LBNL-51699. <http://vis.lbl.gov/Publications/2002/VisGreenFindings-LBNL-51699.pdf>

[Hdf04] *HDF5 – a general-purpose library and file format for storing scientific data*, September 2004 <http://hdf.ncsa.uiuc.edu/HDF5/>

[How04] B. Howe, K. Tanna, P. Turner, D. Maier: *Emergent Semantics: Towards Self-Organizing Scientific Metadata*, International Conference on Semantics for a Networked World (IC-SFNW 2004).

[Met04] *Metadata Encoding & Transmission Standard (METS)*, October 2004 <http://www.loc.gov/standards/mets/>

[Net04] *NetCDF (network Common Data Form)*, October 2004 <http://my.unidata.ucar.edu/content/software/netcdf/>

[Oai02] *The Open Archives Initiative Protocol for Metadata Harvesting*, June 2002 <http://www.openarchives.org/OAI/openarchivesprotocol.html>

[Par04] *ParaView – Parallel Visualization Application*, October 2004 <http://www.paraview.org/>

[Rdf04] *Resource Description Framework (RDF): Concepts and Abstract Syntax* W3C Recommendation, February 2004 <http://www.w3.org/TR/rdf-concepts>

[Swe01] T. Berners-Lee, J. Hendler, O. Lassila: *The Semantic Web*, Scientific American, May 2001

[Val04] M. Valle: *Scientific Data Management*, June 2004 <http://www.cscs.ch/~mvalle/sdm/scientific-data-management.html>

[Wsd01] *Web Services Description Language (WSDL) 1.1* W3C Note, March 2001 <http://www.w3.org/TR/wsdl/>

[Xml04] *Extensible Markup Language (XML) 1.0 (Third Edition)* W3C Recommendation, February 2004 <http://www.w3.org/TR/REC-xml/>

[Xpa03] *XML Package (XPackage)* 1.0 – Editor's Working Draft, March 2003 <http://www.xpackage.org/specification/>

[Xsc01] *XML Schema Part 1: Structures* – W3C Recommendation, May 2001 <http://www.w3.org/TR/xmlschema-1/>