

Minutes of the DFDL-WG meeting at GGF9  
2003-10-07

About 20 people attended.

Four presentations were given:

- Alan Chappell on the repository of examples, GridForge and the interim meeting;
- Martin Westhead on the current drafts of the group documents;
- Jim Myers on "DFDL as hierarchy": presenting an alternate approach, based on transformations, to the current drafts;
- Mike Beckerle on "DFDL proposal based on commercial data-processing requirements".

The repository of examples does not exist yet. Volunteers were sought to host the repository and to supply examples. Reagan Moore volunteered to host the repository at SDSC and this offer was accepted.

The interim meeting is needed to allow focused discussions to continue longer than one GGF time-slot. The meeting has been arranged to spread over several days during Supercomputing 2003, in Phoenix, in November.

Free discussion followed and the following points were raised. The responses to the questions represent approximate consensus, judged without any formal voting.

Is there a distinction between format and technical metadata? Can DFDL carry the metadata needed by a user community. The consensus was that DFDL already does this via the ontologies, and a community can add its own ontology.

What kind of data should DFDL try to describe as examples? Scientific, commercial and generic relation data were all proposed.

Has there been any effort to extend DFDL to data exchange? Yes, DFDL does this by solving the format problem at the receiving end, c.f. prior solutions that prescribe an interchange format at the sending end of the exchange. In any case, an accurate DFDL description is universal. A canonical form of the data might be obtained from the DFDL description.

In the long run we must allow for DFDL descriptions outliving the implementing technology. Therefore, the DFDL transformations must be lossless and we should record their semantics. E.g., we need to record the algorithm of gzip instead of just prescribing a gzip compression.

Can we describe trees and graphs in DFDL? This could be done with XLink, although that solution is not good for large data-sets. This is important for some communities, e.g. for grid scheduling. The capability may be deferred to a later version of DFDL.

Can we handle "implicit" representations, e.g. geospatial data with latitude and longitude stored in a separate file to the main data-structure? Other data-representation systems do this. It was felt to be important that DFDL could describe structures held in more than one file.