

The SED-ML Processing Service and the SED-ML Running Service

Andrew Miller <ak.miller@auckland.ac.nz>



**AUCKLAND
BIOENGINEERING INSTITUTE**

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Te Whare Wānanga o Tāmaki Makaurau

- XPSEDAPI is an effort to produce a standard programmatic interface for working with SED-ML models.
- It is designed to be independent of any particular SED-ML implementation, programming language, or modelling language, so new and existing software tools and SED-ML libraries can all communicate via common interfaces.
- It is defined in Interface Definition Language (IDL) as a series of interfaces, on which attributes can be read and written to, and operations performed.
- IDL can be mapped to nearly any imperative programming language, and fits particularly naturally with object-orientated programming.

- XPSEDAPI is divided into two parts: the SED-ML Processing Service (SProS), and the SED-ML Running Service (SRuS).
- The SED-ML Processing Service allows access to and manipulation of SED-ML documents at a higher level than direct manipulation of the XML.
- All types of SED-ML element have an interface allowing them to be manipulated.
- These interfaces are arranged in an inheritance hierarchy, with all element types ultimately deriving from Base, providing the attributes common to all SED-ML elements. Many elements derive Base via NamedElement, which additionally provides the name.
- Interfaces representing sets of elements, and iterators of those sets as also provided.

- The SED-ML Running Service (SRuS) is an interface for running simulation experiments described in SED-ML.
- It refers to SProS - for example, the simulation experiment is provided as a SProS SEDMLElement interface implementation.
- A SEDMLProcess interface implementation can be queried about what modelling languages are supported, and can apply the transformations described in the SED-ML to those models.
- The generateData operation on the SEDMLElement gives the instruction to actually run the simulation experiment.
- Interfaces to be implemented by the application are provided to asynchronously stream back data from the simulator as it is computed.

- An implementation of XPSEDAPI is included with the CellML API implementation. The inclusion with the CellML API implementation is purely for infrastructural convenience, and it will probably be separated out later.
- The CellML API implementation provides a standard mapping from IDL to C++, a standard reference counting based memory model, and additionally provides automatically generated bridges from the C++ implementation to Python and Java.
- The SProS implementation is completely independent of the core CellML API, but it does use the W3C DOM implementation included with the CellML API.

- The SRuS implementation currently only supports CellML, through use of the CellML Integration Service, but has been designed so it could be extended to support other modelling languages.
- Internally, it can stream back data using the XPSEDAPI interfaces - but only if no aggregate operators are used in the MathML in the model.
- If aggregate operators are used, all data is stored, aggregate values are computed, and data is returned.
- As with the CellML Integration Service, data is sent back in batches for performance reasons, with the trigger for a new batch being either real time elapsed or number of samples collected.
- This approach means that where possible, applications can plot data as it comes back, without sacrificing efficiency too much.



Questions?