

An XML Structure for IOM Simulation Output

Robert J. Stein, Edward Bachta, and M. Pauline Baker
Visualization and Interactive Spaces Lab
Pervasive Tech Labs at Indiana University
Indiana University-Purdue University Indianapolis

Abstract

XML provides useful ways to structure and describe data of many kinds. In this technical note, we describe an XML structure for capturing the output of the IOM (Inverse Ocean Modeling) graphical user interface.

Overview of XML Implementation

Three primary tags are used in the description of the XML DTD for IOM simulation output. This is a quick overview of what these tags are for and our conventions for using them in this DTD. This should aid in understanding the proposed document structure without going into detail about the syntax of XML DTD's. For more thorough information, a good reference for this material is: [W3 School's DTD Tutorial](#).

- **<!ENTITY>**

An <!ENTITY> tag is used as a kind of XML short hand. Every <!ENTITY> has a name and can be referred to later in the DTD in place of the contents of it's tag. <!ENTITY>'s can be nested meaning that the definition of one <!ENTITY> tag can itself use a previously defined <!ENTITY> tag.

For our purposes in this Document <!ENTITY>'s were used in two ways:

1. DTD's do not enforce any data typing, so we defined <!ENTITY>'s for some standard types like *:Boolean, Byte, Double, etc...* This is just a guide for the developer to know what sort of data type will be expected by the XML parser. It is not enforced in any way by the parser itself
2. We also used <!ENTITY> tags as a way to shorten some of the DTD syntax for commonly used things like referencing data stored in an external file (*i.e. OutputFileAttrs*)

- **<!ELEMENT>**

An **<!ELEMENT>** tag is the main tag used in an XML DTD. It is used to define a correct document structure and syntax for a given output file. **<!ELEMENT>** tags can be *EMPTY* or have other **<!ELEMENT>**'s nested inside of them.

In our document we chose that only items that themselves had attributes or other nested children would be considered an **<!ELEMENT>**. Otherwise they would be considered an attribute of some parent tag. For Example: *PhaseSpeed* is defined as an attribute and not an **<!ELEMENT>** since its representation contains simply a value which is in this case a *Double*. The *RunDescription* tag is considered an **<!ELEMENT>** because it has several attributes that are a part of it (*i.e. Author, Time, Date, etc...*) This is a construct of our particular DTD and is not a part of standard XML practice, but it seemed to be a better way of formalizing the description

- **<!ATTLIST>**

An **<!ATTLIST>** tag is always paired with an **<!ELEMENT>** tag and is used to describe attributes of that **<!ELEMENT>**. In our document we use **<!ATTLIST>**'s to represent any item that can be represented by any single data type. See the description of the **<!ELEMENT>** tag for an example of what is and is not an **<!ATTLIST>**

Overview of IOM DTD

The main document structure for xml output consists of a single root node call *IOM_output* which must contain five child nodes: *RunDescription*, *SimulationModel*, *CovarianceInfo*, *IterationInfo* and *SimulationOutput*.

Most of these are very simple. *RunDescription* has a few fields that give some metadata about the output (*who, what, where, when, etc...*). The *SimulationModel* node tells us which simulation model was used and what the model specific parameters were. Right now there is some question as to whether the information contained in *CovarianceInfo* and *IterationInfo* are model-specific or not. If they are they should really get grouped with the other model specific information.

The *SimulationOutput* node will include references to the computed data. This node currently supports output of: Fields, Residuals, Statistics, Prior and Post Error Covariances, and Array Assesment. These were chosen to closely mirror both the features present in the IOM GUI tool as well as the IOM DSL design document. The output document may contain 0 or 1 of any of these output types. For each output type the document must describe the *OutputDimensions*. This will state the number of dimensions present in the output as well as the grid structure of the Output. Regular and Irregular but structured spacing are supported currently. A description of unstructured grids could be added in the future if needed. The actual data for a particular selected output field (*i.e. Fields:FirstGuessField*) will be stored in some external file of a known

format. The external file will be referenced from the xml document by its URL, and the xml will contain information about the file's type and possibly an starting offset to support multiple fields contained in a single file. It is also required that a particular output field will be given a name from a previously chosen list of names. This name will tell applications reading the xml file what the content of the data file is (*i.e. salinity, temperature, velocity, covariance matrix*). These names will need to be agreed upon by the community so that similar data objects will be referred to in a consistent manner across models. Our current list of names is I'm sure a poor guess at what the final list should look like.

The following section is a detailed description of the contents of the DTD.

Element Index

- [ArrayAssesment](#)
- [ArrayModes](#)
- [BackgroundField](#)
- [BackgroundResidualObs](#)
- [BestEstimateField](#)
- [BestEstimateParams](#)
- [BestEstimateResidualObs](#)
- [BoundaryErrorCov](#)
- [BoundaryResidual](#)
- [ClippedBestEstimateField](#)
- [CONVECTION](#)
- [CovarianceInfo](#)
- [DynamicResidual](#)
- [DynamicsErrorCov](#)
- [ErrorCovParams](#)
- [Extent_T](#)
- [Extent_X](#)
- [Extent_Y](#)
- [Extent_Z](#)
- [Fields](#)
- [FirstGuessErrorCov](#)
- [FirstGuessField](#)
- [FirstGuessResidualObs](#)
- [InitialErrorCov](#)
- [InitialResidual](#)
- [IOM_output](#)
- [IrregularSpacing](#)
- [IterationInfo](#)
- [KDV](#)
- [ObservedErrorCov](#)
- [OutputDimension](#)

- [PenaltyValue](#)
- [PenaltyVariance](#)
- [PostErrorCov](#)
- [PriorErrorCov](#)
- [PriorPenalty](#)
- [ReducedPenalty](#)
- [RegularSpacing](#)
- [RepresenterErrorCov](#)
- [RepresenterMatrix](#)
- [ResidualArrayModes](#)
- [Residuals](#)
- [RunDescription](#)
- [SEOM](#)
- [SimulationModel](#)
- [SimulationOutput](#)
- [StateErrorCov](#)
- [Statistics](#)
- [SVD](#)
- [ValueVarianceParams](#)

DTD Tag Description

- **Element:** IOM_output
 - **Children**
 - [RunDescription](#) (*Required*)
 - [SimulationModel](#) (*Required*)
 - [CovarianceInfo](#) (*Required*)
 - [IterationInfo](#) (*Required*)
 - [SimulationOutput](#) (*Required*)
 - **Attributes**
 - NONE
 - **Description:** This is the top-level node in the data description. It defines the document structure by linking to a number of lower level elements described below.

- **Element:** RunDescription
 - **Children**
 - NONE
 - **Attributes**
 - Author (*Required*)
 - DateTime (*Required*)
 - Platform (*Required*)
 - **Description:** General Meta-data for the run.

- **Element: SimulationModel**
 - **Children**
 - [KDV](#)
 - [SEOM](#)
 - [CONVECTION](#)
 - **Attributes**
 - NONE
 - **Description:** This element describes the simulation model that was used to generate these results. This provides a way to encode model specific parameters that may be of interest into the IOM_output document. Right now we only have information on KDV so SEOM and CONVECTION are just guesses at other models for examples sake.

- **Element: KDV**
 - **Children**
 - NONE
 - **Attributes**
 - PhaseSpeed (*Required*)
 - NonLinearity (*Required*)
 - Dispersion (*Required*)
 - Damping (*Required*)
 - DampingDiffusion (*Required*)
 - HalfWidthCalc (*Required*)
 - PhaseSpeedCalc (*Required*)
 - **Description:** NONE

- **Element: SEOM**
 - **Children**
 - NONE
 - **Attributes**
 - NONE
 - **Description:** NONE

- **Element: CONVECTION**
 - **Children**
 - NONE
 - **Attributes**
 - NONE
 - **Description:** NONE

- **Element: CovarianceInfo**
 - **Children**
 - NONE
 - **Attributes**
 - DiagonalWeight (*Required*)
 - DynamicsAmplitude (*Required*)

- DynamicsDecorrelationScale (*Required*)
 - ICAmplitude (*Required*)
 - ICDecorrelationScale (*Required*)
 - TemporalDecorrelationScale (*Required*)
 - **Description:** NONE

- **Element:** IterationInfo
 - **Children**
 - NONE
 - **Attributes**
 - MaxIter (*Required*)
 - Tolerance (*Required*)
 - **Description:** NONE

- **Element:** OutputDimension
 - **Children**
 - [Extent_X](#)
 - [Extent_Y](#)
 - [Extent_Z](#)
 - [Extent_T](#) (*Required*)
 - **Attributes**
 - NONE
 - **Description:** Describe the dimensions of a dataset by including one or more of the following extents. For example the 1D KdV simulation would have an Extent_X and an Extent_T element defined here but not Extent_Y or Extent_Z

- **Element:** RegularSpacing
 - **Children**
 - NONE
 - **Attributes**
 - min (*Required*)
 - max (*Required*)
 - step (*Required*)
 - number (*Required*)
 - **Description:** Describe a single dimension by giving its min and max value as well as its grid spacing and number of elements

- **Element:** IrregularSpacing
 - **Children**
 - NONE
 - **Attributes**
 - number (*Required*)
 - href (*Required*)

- *OutputType (Required)*
 - *Offset*
 - **Description:** For a single dimension describe the number of grid points in this dimension and reference an external file that gives the point coordinates
- **Element:** Extent_(X | Y | Z | T)
 - **Children**
 - [RegularSpacing](#)
 - [IrregularSpacing](#)
 - **Attributes**
 - NONE
 - **Description:** Describes the extent and spacing of a single dimension of output data. This data is required to have either a regular or irregular but structured grid spacing. Unstructured grid spacing has yet to be implemented
- **Element:** SimulationOutput
 - **Children**
 - [Fields](#)
 - [Residuals](#)
 - [Statistics](#)
 - [PriorErrorCov](#)
 - [PostErrorCov](#)
 - [ArrayAssesment](#)
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the data output files. You may include 0 or 1 of any of the child elements. The child elements correspond to the output section of the IOM GUI.
- **Element:** Fields
 - **Children**
 - *OutputDimension (Required)*
 - *BackgroundField (O11-1)*
 - *FirstGuessField (O11-2)*
 - *BestEstimateField (O11-3)*
 - *ClippedBestEstimateField (O11-4)*
 - *BestEstimateParams (O12)*
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the field output data. The actual data is contained in external files of some defined format (*netCDF, HDF, vtk, ascii, binary, etc...*) You may include 0 or 1 of any of the child tags. The child tags here are tagged against the IOM DSL design document.

- **Element:** Residuals
 - **Children**
 - OutputDimension (*Required*)
 - DynamicResidual (O21)
 - InitialResidual (O22)
 - BoundaryResidual (O23)
 - BackgroundResidualObs (O24-1)
 - FirstGuessResidualObs (O24-2)
 - BestEstimateResidualObs (O24-3)
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the residual output data. The actual data is contained in external files of some defined format (*netCDF, HDF, vtk, ascii, binary, etc...*) You may include 0 or 1 of any of the child tags. The child tags here are tagged against the IOM DSL design document.

- **Element:** Statistics
 - **Children**
 - OutputDimension (*Required*)
 - PriorPenalty (O31)
 - ReducedPenalty (O32)
 - PenaltyValue (O33)
 - PenaltyVariance (O34)
 - ValueVarianceParams (O35)
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the statistics output data. The actual data is contained in external files of some defined format (*netCDF, HDF, vtk, ascii, binary, etc...*) You may include 0 or 1 of any of the child tags. The child tags here are tagged against the IOM DSL design document.

- **Element:** PriorErrorCov
 - **Children**
 - OutputDimension (*Required*)
 - RepresenterErrorCov (O41)
 - FirstGuessErrorCov (O42)
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the PriorErrorCov output data. The actual data is contained in external files of some defined format (*netCDF, HDF, vtk, ascii, binary, etc...*) You may include 0 or 1 of any of the child tags. The child tags here are tagged against the IOM DSL design document.

- **Element:** PostErrorCov

- **Children**
 - OutputDimension (*Required*)
 - StateErrorCov (O51)
 - DynamicsErrorCov (O52)
 - InitialErrorCov (O53)
 - BoundaryErrorCov (O54)
 - ObservedErrorCov (O55)
 - ErrorCovParams (O56)
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the PostErrorCov output data. The actual data is contained in external files of some defined format (*netCDF, HDF, vtk, ascii, binary, etc...*) You may include 0 or 1 of any of the child tags. The child tags here are tagged against the IOM DSL design document.
- **Element:** ArrayAssesment
 - **Children**
 - OutputDimension (*Required*)
 - RepresenterMatrix (O61)
 - SVD (O62)
 - ArrayModes (O63)
 - ResidualArrayModes (O64)
 - **Attributes**
 - NONE
 - **Description:** This is the root node for all the ArrayAssesment output data. The actual data is contained in external files of some defined format (*netCDF, HDF, vtk, ascii, binary, etc...*) You may include 0 or 1 of any of the child tags. The child tags here are tagged against the IOM DSL design document.