ILDG Metadata

Working Groups

Hands-on Workshop

June 15, 2023

Overview

1. Overview and Requirements
2. Digression: XML Technologies
3. Ensemble Metadata
4. Configuration Metadata
5. Concluding Remarks

Content

Metadata Content

1. **Physics**
   Description of action and simulation parameters
2. **Algorithm**
   Information about the used algorithm and algorithmic parameters
3. **Source code**
   Used code, compile time parameters, compilation software
4. **Machine**
   Machine used to produce the configuration
5. **Data management**
   Information related to data provenance and checksums

Metadata Requirements

- **Standardised** description of the data (=**metadata**)
  - Solution strategy: **XML documents** which conform to an **XML schema**
- **Unique** description of the data
  - Avoid situation that action is described in two different ways,

    e.g. Iwasaki gauge action

    $$S = \beta \ (c_0 \,\mathrm{plaquette} + c_1 \,\mathrm{rectangular})$$

    or

    $$S = \tilde{c}_0 \,\mathrm{plaquette} + \tilde{c}_1 \,\mathrm{rectangular}$$
- Forward compatible **extensibility** of the standard
  - A metadata document conforming to the current version of the standard should also conform to its future versions
- Broad **coverage** of data that can be described
  - In parts achieved by foreseeing flexibility

Example (1/2)

- SU(3) Wilson plaquette action with gluon fields U and coupling β:

$$S_{\mathrm{G}} = \beta \sum_{\text{plaquette}} \frac{1}{3} \mathrm{Re}\ \mathrm{Tr}\,(1 - U_{\text{plaquette}})$$

Example (2/2)

- (Almost) ILDG-compliant mark-up:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <plaquetteGluonAction>
  ...
    <gluonField>
      <gaugeGroup>SU(3)</gaugeGroup>
      ...
    </gluonField>
    <couplings>
      <elem>
        <beta>5.2</beta>
      </elem>
    </couplings>
  </plaquetteGluonAction>
  ...
</markovChain>
```

Ensembles and Configurations

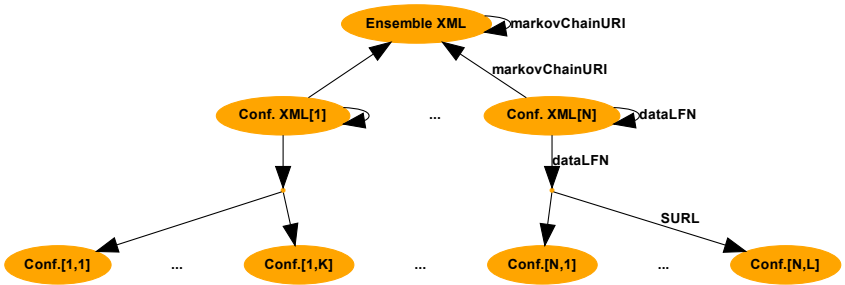
- **Aim:** Avoid replication of complicated data structures
- Markov chains are used to generate gauge field configurations

  Markov chain $\leftrightarrow$ Ensemble XML

  Markov step $\leftrightarrow$ Configuration XML

Linking Metadata and Data

Objects	Links
Ensemble XML document	`markovChainURI`
Configuration XML document	`dataLFN`
Binary data file	



Content

Why XML?

- Both human-readable and machine-readable
- Standardized by the World Wide Web Consortium's (W3C) XML 1.0 Specification
- Availability of the XML schema (XSD) technology that allows to define the necessary metadata for interpreting and validating XML documents
- Availability of standardised technologies to query XML documents
  - W3C's XPath (current version: XPath 3.1)
  - W3C's XQuery (current version: XQuery 3.1)
- Many tools for processing XML documents
  - For parsing: `libxml2` with interfaces for C, C++, Python, Perl
  - For formatting and validation: `xmllint`

Background on XML

- Simple example:

```
<?xml version="1.0" encoding="UTF-8"?>
<dataEntry>
  <key>7</key>
  <value>Hello world!</value>
</dataEntry>
```

- Key elements:
  - Tag: Markup construct that begins with $<$ and ends with $>$
    - Start tag: $<$elem$>$
    - End tag: $<$ /elem$>$
    - Empty-element tag: $<$elem/ $>$
  - Element: Logical document component starting/ending with a tag (or empty-element tag)
  - Attribute

XML Schema

- Technology to formally describe the elements in an XML document
- Simple example:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="dataEntry">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="key" type="xs:decimal"/>
        <xs:element name="value" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Validating XML Files (1/2)

- Document validation using the `xmllint` tool:

```
% xmllint --schema ./tst-schema.xml ./tst-doc-ok.xml
<?xml version="1.0" encoding="UTF-8"?>
<dataEntry>
  <key>7</key>
  <value>Hello world!</value>
</dataEntry>
./tst-doc-ok.xml validates
```

Validating XML Files (2/2)

- Try document with incorrect tag:

```
% xmllint --schema ./tst-schema.xml ./tst-doc-bad-1.xml
<?xml version="1.0" encoding="UTF-8"?>
<dataEntry>
  <idx>7</idx>
  <value>Hello world!</value>
</dataEntry>
./tst-doc-bad-1.xml:3: element idx: Schemas validity error : Element 'idx': This element is not expected. Expected is ( key ).
./tst-doc-bad-1.xml fails to validate
```

- Try document with incorrect element content:

```
% xmllint --schema ./tst-schema.xml ./tst-doc-bad-2.xml
<?xml version="1.0" encoding="UTF-8"?>
<dataEntry>
  <key>seven</key>
  <value>Hello world!</value>
</dataEntry>
./tst-doc-bad-2.xml:3: element key: Schemas validity error : Element 'key': 'seven' is not a valid value of the atomic type 'xs:decimal'.
./tst-doc-bad-2.xml fails to validate
```

Content

Ensemble XML Overview

Top-level elements of the ensemble XML file:

- `markovChainURI`
  - A globally unique identifier that identifies
    - An Ensemble XML document
    - A set of configurations (metadata + data)
- `management`
  - Data relevant for tracing and recording the origins of the data (data provenance)
- `physics`
  - Area where providers must provide all information needed to define the physical parameters of the data set (ensemble)
- `algorithm`
  - Area where providers can provide all information needed to describe the used algorithms (except for configuration-specific information)

Ensemble XML: markovChainURI

- Schema mandates an Uniform Resource Identifier (URI)
- ILDG convention not enforced by schema to ensure uniqueness: `mc://<reg>/<col>/<prj>/...`
  - Regional grid (`reg`)
  - Collaboration (`col`)
  - Project (`prj`)
- Warning: ILDG currently does not have mechanisms to check uniqueness of the `markovChainURI`

Example:

```
1 <?xml version="1.0"?>
2 <markovChain>
3   <markovChainURI>
4     mc://ldg/qcdsf/clover_nf2p1/b7p20kp12450kp12450-16x32
5   </markovChainURI>
6   ...
7 </markovChain>
```

Ensemble XML: `management`

- Collaboration
  - String unconstrained by the schema, but should be consistent and unique
- Project name
  - String unconstrained by the schema, but should be consistent and unique at collaboration level
- Archive history
  - List of entries for tracking data provenance
  - Each entry comprises
    - Action: `add`, `replace`, `remove`
    - Participant identified by name and institution
    - Date

Example:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <management>
    <collaboration>qcdsf</collaboration>
    <projectName>clover_nf2p1</projectName>
    <archiveHistory>
      <elem>
        <revisionAction>add</revisionAction>
        <participant>
          <name>Yoshifumi Nakamura</name>
          <institution>NIC/DESY</institution>
        </participant>
        <date>2007-03-30T14:34:29+02:00</date>
      </elem>
    </archiveHistory>
    <replicate>deny</replicate>
  </management>
  ...
</markovChain>
```

Ensemble XML: `physics` / Overview

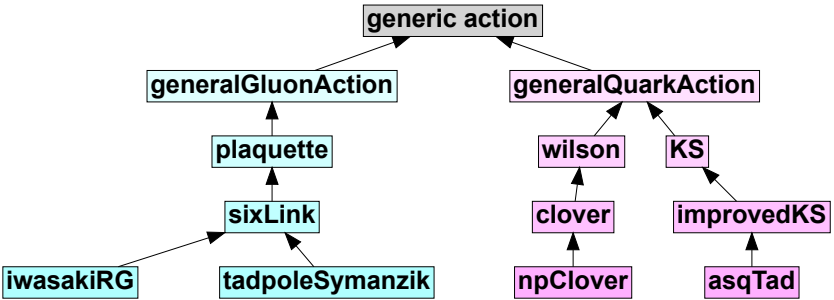
- `size`: Lattice size
  - Unbounded list of elements
  - Each element comprises of `name` and `length`
- `action/gluon`: Gluonic part of the action
- `action/quark`: Fermionic part of the action

Example:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <physics>
    <size>
      <elem>
        <name>X</name>
        <length>16</length>
      </elem>
      <elem>
        <name>Y</name>
        <length>16</length>
      </elem>
      <elem>
        <name>Z</name>
        <length>16</length>
      </elem>
      <elem>
        <name>T</name>
        <length>32</length>
      </elem>
    </size>
    ...
  </physics>
  ...
</markovChain>
```

Ensemble XML: `physics` / Action

- The description of action parameters is most critical to ensure metadata being unique and extensible
- Strategy: Hierarchy of actions



Ensemble XML: physics / Gluonic Part (1/2)

- The element gluon contains an unbounded list of gluon actions
- Various gluon actions have been defined since the schema has been established
  - plaquetteGluonAction
  - iwasakiRGGluonAction
  - tpLuescherWeiszGluonAction
  - anisotropicTpWilsonGluonActionType
  - ...

Example:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <physics>
    ...
    <action>
      <gluon>
        <plaquetteGluonAction>
          <glossary>http://...</glossary>
          <gluonField>
            <gaugeGroup>SU(3)</gaugeGroup>
            <representation>fundamental</representation>
            <boundaryCondition>
              <elem>periodic</elem>
              <elem>periodic</elem>
              <elem>periodic</elem>
              <elem>periodic</elem>
            </boundaryCondition>
          </gluonField>
          <beta>7.2</beta>
        </plaquetteGluonAction>
      </gluon>
      ...
    </action>
    ...
  </physics>
  ...
</markovChain>
```

Ensemble XML: physics / Gluonic Part (2/2)

- All gluon actions must contain
  - glossary: URI to documentation
  - gluonField
  - topologyFixing (optional)
- The element gluonField must contain the following sub-elements
  - gaugeGroup
    - Allowed values: SU(3), SU(2), ...
  - representation:
    - Allowed values: fundamental, adjoint
  - boundaryCondition
    - List of elements with the following values: periodic, antiperiodic, dirichlet
- Other elements (e.g., beta) are action specific

Example:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <physics>
    ...
    <action>
      <gluon>
        <plaquetteGluonAction>
          <glossary>http://...</glossary>
          <gluonField>
            <gaugeGroup>SU(3)</gaugeGroup>
            <representation>fundamental</representation>
            <boundaryCondition>
              <elem>periodic</elem>
              <elem>periodic</elem>
              <elem>periodic</elem>
              <elem>periodic</elem>
            </boundaryCondition>
          </gluonField>
          <beta>7.2</beta>
        </plaquetteGluonAction>
      </gluon>
      ...
    </action>
    ...
  </physics>
  ...
</markovChain>
```

Ensemble XML: physics / Fermionic Part (1/3)

- The element quark contains an unbounded list of quark actions
- Various quark actions have been defined since the schema has been established
  - npCloverQuarkAction
  - overlapQuarkAction
  - ...
- All quark actions must contain
  - glossary: URI to documentation
  - quarkField
  - numberOfFlavours
  - linkSmearing (optional)
- Other elements (e.g., kappa) are action specific

Example:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <physics>
    ...
    <action>
      ...
      <quark>
        <npCloverQuarkAction>
          <glossary>...</glossary>
          <quarkField>...</quarkField>
          <numberOfFlavours>2</numberOfFlavours>
          <linkSmearing>...</linkSmearing>
          <kappa>0.12450</kappa>
          <cSW>1.0</cSW>
        </npCloverQuarkAction>
        <npCloverQuarkAction>
          <glossary>...</glossary>
          <quarkField>...</quarkField>
          <numberOfFlavours>1</numberOfFlavours>
          <linkSmearing>...</linkSmearing>
          <kappa>0.12450</kappa>
          <cSW>1.0</cSW>
        </npCloverQuarkAction>
        ...
    </action>
    ...
  </physics>
  ...
```

Ensemble XML: physics / Fermionic Part (2/3)

- The element quarkField must contain the following sub-elements
  - normalisation
    - Allowed values are not prescribed
  - boundaryCondition
    - List of elements with the following values: periodic, antiperiodic, dirichlet

Example for quarkField:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <physics>
    ...
    <action>
      ...
      <quark>
        <npCloverQuarkAction>
          ...
          <quarkField>
            <normalisation>sqrt2kappa</normalisation>
            <boundaryCondition>
              <elem>periodic</elem>
              <elem>periodic</elem>
              <elem>periodic</elem>
              <elem>antiperiodic</elem>
            </boundaryCondition>
          </quarkField>
          ...
        </npCloverQuarkAction>
        <npCloverQuarkAction>
          ...
        </npCloverQuarkAction>
        ...
      </action>
    ...
  </physics>
  ...
```

Ensemble XML: physics / Fermionic Part (3/3)

- The element linkSmearing must contain the following sub-elements
  - An element defining the link blocking, e.g.
    - apeLinkBlocking
    - anisotropicApeLinkBlocking
  - An element defining the link unitarization, e.g.
    - stoutLinkUnitarization
    - invSqRootLinkUnitarization
  - numSmear: integer value

Example for linkSmearing:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <physics>
    ...
    <action>
      ...
      <quark>
        <npCloverQuarkAction>
          ...
          <linkSmearing>
            <apeLinkBlocking>
              <rho0>0.1</rho0>
              <rho1>0.1</rho1>
            </apeLinkBlocking>
            <stoutLinkUnitarization/>
            <numSmear>1</numSmear>
          </linkSmearing>
          ...
        </npCloverQuarkAction>
        <npCloverQuarkAction>
          ...
        </npCloverQuarkAction>
        ...
      </action>
    ...
  </physics>
  ...
</markovChain>
```

Ensemble XML: `algorithm`

- The element `algorithm` must contain the following sub-elements
  - `name`: A string restricted to legal XML 1.0 names
  - `glossary`: URI to documentation
  - `reference`: String containing, e.g., a publication
  - `exact`: Boolean value
  - `parameters` (optional)
  - Option to insert other XML elements using a different namespace
- `parameters` is a list of pairs
  - `name`: A string restricted to legal XML 1.0 names
  - `value`: Any simple type

Example:

```
<?xml version="1.0"?>
<markovChain>
  ...
  <algorithm>
    <name>qcdsfAcceleratedHMC</name>
    <glossary>http://www-zeuthen.desy.de/latfor/ldg/algorithmGloss
    <reference/>
    <exact>true</exact>
    <parameters>
      <name>timeScaleRatio</name>
      <value>3</value>
    </parameters>
  </algorithm>
</markovChain>
```

Content

Configuration XML Overview

Top-level elements of the configuration XML file:

- `management`
- `implementation`
- `algorithm`
- `precision`
- `markovStep`

Example:

```
<?xml version="1.0"?>
<gaugeConfiguration>
  <management>
    ...
  </management>
  <implementation>
    ...
  <algorithm>
    ...
  </algorithm>
  <precision>...</precision>
  <markovStep>
    ...
  </markovStep>
</gaugeConfiguration>
```

Configuration XML: `management`

- CRC32 checksum of the `ildg-binary-data` part of an ILDG configuration based on the LIME format
- Archive history
  - List of entries for tracking data provenance
  - Each entry comprises
    - Action: `generate`, `add`, `replace`, `remove`
    - Participant identified by name and institution
    - Date

Example:

```
<?xml version="1.0"?>
<gaugeConfiguration>
  <management>
    <crcCheckSum>2266949024</crcCheckSum>
    <archiveHistory>
      <elem>
        <revisionAction>generate</revisionAction>
        <participant>
          <name>Hinnerk Stueben</name>
          <institution>ZIB</institution>
        </participant>
        <date>2004-11-15T19:33:55+01:00</date>
      </elem>
    </archiveHistory>
  </management>
  <implementation>
    ...
  </implementation>
  <algorithm>
    ...
  </algorithm>
  <precision>...</precision>
  <markovStep>
    ...
  </markovStep>
</gaugeConfiguration>
```

Configuration XML: `implementation`

- `machine`
  - `name`: String referring to the name of the computer used to generate the configuration
  - `institution`: String with the name of the organisation hosting that computer
  - `machineType`: String showing the computer type
  - `comment`: String with optional additional information
- `code`
  - `name`: String with the name of the code
  - `version`: String with the code veroisn
  - `date`: Date of the code release
  - `comment`: String with optional additional information

Example:

```
<?xml version="1.0"?>
<gaugeConfiguration>
  <management>
    ...
  </management>
  <implementation>
    <machine>
      <name>sr8000</name>
      <institution>LRZ Munich</institution>
      <machineType>Hitachi SR8000-F1</machineType>
    </machine>
    <code>
      <name>BQCD</name>
      <version>3.0.1</version>
      <date>2003-11-28T13:00:00+01:00</date>
    </code>
  </implementation>
  <algorithm>
    ...
  </algorithm>
  <precision>...</precision>
  <markovStep>
    ...
  </markovStep>
</gaugeConfiguration>
```

Configuration XML: `algorithm`

- The optional element `parameters` comprises a list of pairs
  - `name`: A string restricted to legal XML 1.0 names
  - `value`: Any simple type
- It is allowed to insert other XML elements using a different namespace

Example:

```
<?xml version="1.0"?>
<gaugeConfiguration>
  <management>
    ...
  </management>
  <implementation>
    ...
  </implementation>
  <algorithm>
    <parameters>
      <name>targetResidue</name>
      <value>1e-07</value>
    </parameters>
  </algorithm>
  <precision>...</precision>
  <markovStep>
    ...
  </markovStep>
</gaugeConfiguration>
```

Configuration XML: `precision`

- Allowed values:
  - `single`: Single precision calculations
  - `double`: Double precision calculations
  - `mixed`: Mix of single and double precision calculations
- Note that configurations may be stored in different precision

Example:

```
<?xml version="1.0"?>
<gaugeConfiguration>
  <management>
    ...
  </management>
  <implementation>
    ...
  </implementation>
  <algorithm>
    ...
  </algorithm>
  <precision>double</precision>
  <markovStep>
    ...
  </markovStep>
</gaugeConfiguration>
```

Configuration XML: markovStep

- markovChainURI: Reference to the ensemble
- series: Whitespace-replaced and collapsed string identifying a run
- update: Any simple type that references an update within a run
- avePlaquette: Average plaquette value (double)
- dataLFN: URI that is a globally unique reference to a configuration

Example:

```
<?xml version="1.0"?>
<gaugeConfiguration>
  <management>
    ...
  </management>
  <implementation>
    ...
  </implementation>
  <algorithm>
    ...
  </algorithm>
  <precision>...</precision>
  <markovStep>
    <markovChainURI>mc://...</markovChainURI>
    <series>563</series>
    <update>1310</update>
    <avePlaquette>0.5610635491</avePlaquette>
    <dataLFN>lfn://reg/col/prj/...</dataLFN>
  </markovStep>
</gaugeConfiguration>
```

Content

Concluding Remarks

- The ILDG metadata standards have been a robust framework for describing LQCD configurations data
  - Forward compatibility has been important
  - But: Evolution of the standard stagnated during the last 10 years
- The complexity of the underlying metadata schema is in practice not exposed to end-users
  - Most metadata documents are derived from existing documents with small modifications
- The restart with ILDG 2.0 is an opportunity to identify gaps in terms of
  - Training and tools for generating metadata
  - Coverage of data that can be described by ILDG-compliant metadata