Metadata Working Group report on the QCDml version 2.0

ILDG Metadata Working Groups*

November 24 2024

Abstract

This is a note on the update of the QCDml schemata towards version 2.0.

Contents

[*] Members as of November 2024: G. Andronico, J. Hettrick, G. von Hippel, G. Koutsou, H. Matsufuru, Y. Nakamura, D. Pleiter, H. Simma, J. Simone, C. Urbach, and T. Yoshie.

1 Introduction

Recently activity of the International Lattice Data Grid (ILDG) has been rebooted [1, 2]. For an overview of previous ILDG activity (called ILDG1 hereafter), see *e.g.* [3].

At the Lattice conference 2022, a parallel session on Lattice data was held [4] where a number of collaborations expressed their interest to share gauge configurations through ILDG. The ILDG online Hands-on workshop was held on 14-16 June 2023 [5]. In this hands-on workshop, more detailed requests from the participated collaborations were raised. To satisfy these requirements, we need to update the QCDml schemata. This includes not only extensions, but also modifications which are incompatible with the already uploaded (legacy) metadata. The necessary conversion of the legacy metadata to the new schema is possible with limited effort before the start of massive new uploads for ILDG2.

In this document, we summarize the requests from the collaborations to enable markup their data, the proposed update of the QCDml schemata, and related discussions in the Metadata Working Group.

The latest draft for the updated XSD files, together with the previous released versions and corresponding sets of example XML (which validate against these schemata), are available in the public gitlab repository [6] (see branch "`next`"). Related extensions are also taken into account in the latest draft an update of the "ILDG Binary File Format", which can be found in the public gitlab repository [7] (see branch "`next`").

The first public version of QCDml was made available in 2004 [8]. The latest released versions of the QCDml schemata are

- QCDmlConfig1.3.1.xsd (2010/01/17)
- QCDmlEnsemble1.4.8.xsd (2013/06/08)

General guiding principles for the design of the QCDml metadata schema were stated at Lattice 2007 [9] and in [8]:

- The schema has to be *extendable* as parameters of future simulations cannot be anticipated.
- The mark-up of simulation parameters has to be *unique* to avoid *e.g.* the same action being described in two different ways.
- The schema has been kept *general* enough to allow the description of data other than gauge configuration (propagators, correlators, etc.) in the future.

This report is organized as follows. In the next section, we list the requests for the update of the QCDml schemata and summarize the status of their implementation. Considerations on the schema design and implementation details of the proposed updates are given in Sect. 3.

2 Requests

2.1 Requests from collaborations

In the summary discussion at the Hands-on workshop in June 2023 [5], the participating collaborations raised requests for extensions of the markup. All collaborations with interest to start upload of new configurations were asked to formulate their requests in more detail and the following documents were received:

collaboration	main requests
BSM [10]	markup of the Beyond Standard Model physics
CLS [11]	simulations with open BC and reweighting
HotQCD [12]	markup and upload of multiple configurations in single ConfigXML and binary file
JLQCD [13]	simulations with Möbius domain-wall fermions
QCDSF-UKQCD-CSSM [14]	QCD+QED simulations
FASTSUM [15]	support for ORCID and general observables
openLAT [16]	simulations with exponential Clover term, open/Schrödinger functional BC, reweighting

In order to enable upload of new configurations, these requests have been used by the MDWG as main basis for drafting the current update of the metadata schemata. For the markup of QCD+QED simulations, also the possibility to extend the schema towards other setups, *e.g.* compact QED with C^* boundary conditions [18], has been taken into account.

In addition, several suggestions for changes came up and were incorporated during the discussions in the MDWG in order to improve structure, coherence, and practical usability (e.g. for Xpath searches) of the schemata, *e.g.*

- Restructuring of the XML tree (e.g. for `<parameters>`, `<size>`, and `<boundaryCondition>`)
- Renaming of elements, or removal of un-used/obsolete elements

See Sect. 3.2.5 for a complete list and a more detailed discussion.

2.2 List of requests

A detailed list of specific requests has been extracted from the above documents and is summarized in the following. Requests without reference have been included during the discussion in the working group. For each request we indicate the status of its implementation in the current draft of the new schemata, using the following super-scripts:

$*$ if the implementation is not compatible with legacy XML documents. (The necessary conversion of these documents is easily possible when restoring them in ILDG2 by scripts that are already available for the proposed schema changes.)

† if the solution needs further discussion and/or might be postponed

∅ if the request is only a matter of better documentation

For resolved requests, also corresponding example XML documents are mentioned, which can be found in the sub-directories `config/2.0.0/ref-valid` and `ensemble/2.0.0/ref-valid` of the branch "`next`" in the gitlab repository [6].

ConfigXML

(a-01) Markup and packaging of multiple configurations [12]

Simulations at finite temperature or step scaling often have a large number of small configurations. Thus, markup and packaging as single files becomes not economical.

Status*: Resolved by new `<markovSequence>` in QCDmlConfig and corresponding extension of the ILDG Binary File Format [7]
Details: see Sect. 3.3.1
Examples: `c-multi1.xml` and `c-multi2.xml`

(a-02) Specification of plaquette [11]

For simulations with open or Dirichlet boundary condition for the gauge field, the action and the plaquette sum is defined with a weight factor $w(p) \neq 1$ for plaquettes which involve links on the boundary.

Status$^{\emptyset}$: To be specified in schema documentation

(a-03) Open boundary condition [11]

Status: not relevant for markup of configs, see (b-03),

(a-04) Reweighting factor [11, 16]

Status*: Resolved by adding mandatory `<reweightingNeeded>` in QCDmlEnsemble, together with new `<additionalInfo>`, see (a-07)
Example: `c-free-top-level.xml` and `e-free-top-level.xml`

(a-05) Markup and storage for Dirichlet boundary conditions [11]

Status: Resolved, see (b-03) and (b-10)

(a-06) Additional comments [11]

Status*: Resolved by adding optional `<annotation>` elements in both schemata
Details: see Sect. 3.2.1
Examples: `c-annotation-*.xml` and `e-annotation-*.xml`

(a-07) Any kind of observables [15]

Status: Resolved by adding optional top-level element `<additionalInfo>` to include a free subtree in ensemble and configuration schema
Details: see Sect. 3.2.2
Example: `c-free-top-level.xml` and `e-free-top-level.xml`

(a-08) Addition of ORCID in `<participant>` section [15]

Status: Resolved
Details: see Sect. 3.2.3
Example: `e-participant.xml`

(a-09) Structure and annotation of parameter elements [11]

Status*: Resolved by grouping `<name>` and `<value>` into new `<parameter>` element (with optional annotation) in ensemble and configuration schema
Details: see Sect. 3.2.4
Examples: `c-param.xml` and `e-param.xml`

EnsembleXML

(b-01) Support for BSM gauge groups [10]

Status: Resolved by extension of `gaugeGroupType` in QCDmlEnsemble and ILDG Binary File Format specification [7]
Details: see Sect. 3.4.1
Example: `e-gauge-groups.xml`

(b-02) Higher fermion representation [10]

Status: Solved by introducing 3 new actions
Details: see Sect. 3.4.2
Example: `hirep.xml`

(b-03) Open/OpenSF boundary condition [11, 16] (and [18])

Status: Resolved by adding `open`, `openSF`, and `cstar` in enumeration for `boundaryConditionType` and adding specific actions *e.g.* `treelevelSymanzikOpenBCGluonAction` and `npCloverOpenBCQuarkAction`.
Details: see Sect. 3.4.3
Examples: `e-open.xml` and `e-openSF.xml`

(b-04) Möbius domain-wall fermion [13]

Status: Resolved
Details: see Sect. 3.4.4
Example: `e-moebiusDW.xml` and `e-shamirDW.xml`

(b-05)† QED gauge action [14] (and [18])
Requires an additional (optional) `<photon>` sub-tree under `<action>`

Status†: Resolved (thanks to helpful discussions with A. Patella)
Details: see Sect. 3.4.5
Examples: `slinc-qed.xml` and `rcstar1.xml`

(b-06) Fermion coupled to electromagnetic field [14] (and [18])

Status†: Resolved by additional instances of "`QuarkActionType`".
Details: see Sect. 3.4.6
Examples: `slinc-qed.xml` and `rcstar1.xml`

(b-07) SLiNC fermion [14]

Special case of (b-06)
Status†: See (b-05)
Example: `slinc-qed.xml`

(b-08) License and/or use conditions
Requested by [11] and others (also a requirement for FAIR data [19])

Status*: Resolved by new mandatory top-level element `<license>` (supporting standard or custom license, and optional embargo period)
Details: see Sect. 3.4.8.
Examples: `e-license*.xml`

(b-09) Acknowledgment to funding or computer-time grants [11, 14]

Status: Resolved by including subset of `<fundingReferences>` from DataCite [17] as optional top-level element, as well as `<acknowledgment>` in (b-08)
Details: see Sect. 3.4.9
Example: `e-funding.xml`

(b-10) Dirichlet boundary condition [11]

Requires specification of convention `<size>` element
Status$^\emptyset$: See last remark in Sect. 2.3 of draft for ILDG Binary File Format specifications [7], still to be included also in schema documentation

(b-11) Additional comments [11]

Status*: Resolved, see (a-06)

(b-12) Exponential clover fermion action [16]

Status: Resolved by including `<npExpCloverQuarkAction>` as instance of "`cloverQuarkActionType`"
Example: `e-npExpClover.xml`

(b-13) Same as (a-09)

(b-14) Free top-level element [11]

Status: Resolved, analogous to (a-07)

(b-15) Structure of the elements `<size>` and `<boundaryCondition>`

Status*: Resolved by introducing pre-defined names instead of `<elem>`
Details: see Sect. 3.4.10
Examples: all ensemble XML documents

3 Details and implementation of the proposed schema updates

3.1 Implementation conventions

We strictly preserve (as in all former versions of QCDml) the following restrictions of the generic XML syntax allowed by the schema:

- No use of attributes (except for root element of XML)
- No mixed elements (i.e. no elements with children AND text)
- Order of child elements is always fixed and specified by the schema
- All element names are defined by the schema (except in free sub-trees)

Moreover, the proposed schema update was guided by the aim to respect and realize, as far as possible, the following design rules:

(A) Introduce specific names for action sub-trees, which have specific properties or values of parameters (e.g. versions with tree-level, non-perturbative, and no improvement), even if they have (or derive from) the same generic structure

(B) Avoid use of optional elements in specification of actions or algorithm (to enforce well-defined information and possibly simplify Xpath searches)

(C) Avoid leaves without value. Leaf elements which are mandatory should also be required to have a non-empty value (either defined by an enumeration or with a documented recommendation of a dummy value, like `UNKNOWN`, as fall-back)

(D) Lists of repeated elements follow the naming strategy of dataCite [17]

```
<P> <E>...</E> <E>...</E> ...  </P>
```

where ”`P`” is the plural of the element name ”`E`”

3.2 Updates common to QCDmlConfig and QCDmlEnsemble

3.2.1 Additional comments*

QCDmlConfig already supports some optional `<comment>` elements. The possibility to provide such optional information in the has been added in several places of the configuration and ensemble XML. The proposed name of these elements (with free text content) is `<annotation>` their location has been chosen to be always the **first** child of the parent element.

These `<annotation>` elements also replace the former `<comment>` elements in QCDmlConfig. Therefore, legacy configuration XML documents which contain such elements need to be converted (by renaming and moving them to become the first child element).

3.2.2 Free sub-tree as optional top-level element

In the previous QCDml, free sub-trees are allowed only as last child of the `<algorithm>` element (in configuration and ensemble XML).

To allow free markup of properties not explicitly specified in the QCDml schema (and logically not directly related to the algorithm specifications), an optional element `<additionalInfo>` has been introduced. If present, it must be the last of the top-level elements, and can hold a free (but non-empty) XML sub-tree and/or an `<annotation>` element (with default `xmlns`). This sub-tree is expected to be used *e.g.* for providing estimators of reweighting factors (a-04) or measurements of observables without standardized schema (a-07).

In the XSD, the free sub-tree is implemented as

```
<xs:any namespace="##other" maxOccurs="unbounded" processContents="lax"/>
```

i.e. requiring a different namespace (by `namespace="##other"`) and with validation errors suppressed (by `processContents="lax"`). This enables free markup of additional information, like for example

```
<additionalInfo>
   <someElement xmlns="some.url://example">
      This is a completely free sub-tree, to be used e.g. for
      * info on reweighting factors
      * observables without standardized schema
   </someElement>
</additionalInfo>
```

3.2.3 ORCID

In previous QCDml versions, the `<participant>` in the `<management>` sub-tree had two mandatory children, `<name>` and `<institution>` (with possibly empty values).

ORCID is meanwhile widely used in the scientific community as a unique, persistent. and public identifier of a researchers. Also the fact that name and institution (*i.e.* the affiliation at the time of the `<revisionAction>`) are **personal data** is a concern with respect to GDPR which still requires further consideration and clarification. Specification of the `<participant>` by an ORCID is therefore a desirable (and preferred) option, which has been requested (a-08) and may help to achieve GDPR compliance.

The proposed schema modification specifies two alternative forms for the children of the `<participant>` element:

- `<orcid>` (mandatory), followed by optional `<name>` and/or `<institution>`
- `<name>` and `<institution>` (both mandatory)

The first form is recommended one, while the latter should only be used when converting legacy metadata with has a participant without ORCID.

Example: see `e-participant.xml`

3.2.4 Parameter element*

Algorithm parameters are currently marked up as a consecutive combination of `<name>` and `<value>` elements as children of `<parameters>`. This lacks a clear structure (e.g. for Xpath queries) and does not allow to add optional annotations. We propose to introduce an additional hierarchy level using a new `<parameter>` element for grouping. Thus, `<parameters>` becomes a list of `<parameter>` elements (also in line with the DataCite naming style, c.f. rule (D) of Sect. 3.1). Each `<parameter>` element has mandatory child elements

- `<name>`
- `<value>`

An optional (first) `<annotation>` element allows to provide additional explanations on the parameter (e.g. its meaning or definition, and/or the choice of the specific value), see also (a-06).

Examples: see `c-param.xml` and `e-param.xml`

3.2.5 Other changes*

Further changes in the (logical and technical) implementation of the schemata were suggested and incorporated during the discussions in the MDWG in order to improve structure, coherence (with the guiding rules of Sect. 3.1), and practical usability (e.g. for Xpath searches) of the schemata:

- Removed optional `<replicate>` element in `<management>` of ensemble XML (un-used in legacy metadata and obsolete by `license>`)
- Removed obsolete elements (e.g. `<data>` and `<array>`)

- Renamed element `<elem>` to `<archiveEvent>` in `<archiveHistroy>` to improve readability (and w.r.t. rule (C) of Sect. 3.1)
- Stronger restrictions on some leaf element values (c.f. rule (D) of Sect. 3.1). Further cases are still to be considered in future revisions.
- Replaced (in QCDmlEnsemble) the empty leaf elements `<stoutLinkUnitarization>` and `<invSqRootLinkUnitarizationType>` by a new element `<linkUnitarization>` with enumeration values "`stout`", "`invsqrt`, and "`none`" (c.f. rule (C) of Sect. 3.1)
- Restructured XSD implementation to avoid auxiliary top-level elements (which would allow incorrect validation of XML documents with only such an element) and `substitutionGroup` attributes (which require common base types and thus obstruct implementation of harder consistency conditions, e.g. on particular values of elements)

Some of these changes render existing XML documents incompatible with the new schemata, and thus require conversion of all legacy metadata. It is therefore desirable to include all these changes already in the 2.0.0 release of the schemata to avoid extra conversion efforts (which would become necessary if these changes were postponed to future schema releases). Scripts to automatically perform the conversions needed for all the proposed (non-backward compatible) changes have been prepared and tested. They allows to quickly convert (the moderate number of) all existing XML documents from ILDG1 before restoring them in the new Metadata Catalogue instances of ILDG2.

3.3 Updates to QCDmlConfig

3.3.1 Multiple configurations in single XML and binary file*

Markup (and packaging) of multiple configurations in a single config XML (and binary file) has been requested for finite-temperature projects. This possibility might also be convenient in other simulations, *e.g.* for step scaling, with a large number of relatively small configurations. Markup and storage of individual configurations would then become inconvenient and inefficient (*e.g.* due to the large number of access operations to the Metadata Catalogue and Storage Elements).

The proposed solution closely follows the suggestion [12] from request (a-01). The (mandatory) top-level element `<markovStep>` is replaced by a (mandatory) `<markovSequence>` element, which in turn can have one or more `<markovStep>` children.

The elements `<dataLFN>` and `<markovChainURI>` define unique identifiers in ILDG: the former identifies the config XML document and the corresponding binary data file, the latter specifies the unique Markov chain to which the configuration(s) belong. Thus, `<dataLFN>` is moved to become the first top-level element, while `<markovChainURI>` and `<series>` (to identify e.g. replica) are moved from the (original) `<markovStep>` to the (new) `<markovSequence>`.

Moreover, in case of simulations with multiple gauge groups, like QCD+QED, a single `<markovStep>` can refer to multiple `ildg-binary-data` records of the data file which all belong to the same update. Therefore, the new `<markovStep>` has as children an `<update>` element folowed by one or more `<record>` elements. The latter correspond to the individual messages of the binary file which hold an `ildg-binary-data` record and have the child elements

- `<field>` holding the same value as the `<field>` element of the `ildg-format` record
- `<crcCheckSum>` holding the ILDG checksum (which previously was stored in under the `<management>` sub-tree)
- `<avePlaquette>` holding the average plaquette of that gauge field

The following XML fragment illustrates a `<markovSequence>` with multiple `<markovStep>` elements (see also `c-multi1.xml` and `c-multi2.xml`):

```
<markovSequence>
   <markovChainURI>mc://x/y/e1</markovChainURI>
   <series>0</series>
   <markovStep>
```

```
      <update>991</update>
      <record>
        <field>su3gauge</field>
        <crcCheckSum>538770653</crcCheckSum>
        <avePlaquette>0.61529211</avePlaquette>
      </record>
   </markovStep>
   ...
   <markovStep>
      <update>992</update>
      <record>
        <field>su3gauge</field>
        <crcCheckSum>UNKNOWN</crcCheckSum>
        <avePlaquette>0.5435737</avePlaquette>
      </record>
   </markovStep>
</markovSequence>
```

3.4 Updates to QCDmlEnsemble

3.4.1 Additional gauge groups

The “gaugeGroupType” in QCDmlEnsemble (and the values supported in the `ildg-format` record of the `ildg-binary-format` specifications) are now implemented (by regular expressions) to support any of the following gauge groups:

- $SU(n)$ and $SO(n)$ with $n \geq 2$
- $Sp(n)$ with $n \geq 4$ even
- $U(n)$ with $n \geq 1$

Example: `e-gauge-groups.xml`

3.4.2 Higher fermion representation

Since quark actions using a higher fermion representation have a different functional form, it is preferable (*c.f.* rules (A) and (B) of Sect. 3.1) to introduce corresponding new quark-actions, instead of just adding an optional `<quarkRepresentation>` element in existing actions.

As a solution for request (b-02), three new actions

- `<wilsonAdjointQuarkAction>` (`adjoint`)
- `<wilsonTwoIndexSymmetricQuarkAction>` (`2-index-symmetric`)
- `<wilsonTwoIndexAntisymmetricQuarkAction>` (`2-index-antisymmetric`)

are added. Their `<quarkField>` element contains a mandatory `<representation>` element which is restricted to a corresponding unique value (indicated in parenthesis above). Although this element is redundant (due to the use of 3 different action names), this element is guaranteed to be consistent and is introduced to keep the metadata normalized and searchable.

3.4.3 Open/OpenSF boundary condition

While the open boundary condition or open Scrödinger functional boundary condition can be specified in `boundaryCondition`, the actions have additional parameters for boundary improvement coefficients. Based on the rule (A) of Sect. 3.1), new actions `treelevelSymanzikOpenBCGluonAction` and `npCloverOpenBCQuarkAction` with additional elements for `cG` and `cF`), respectively.

3.4.4 Möbius domain-wall fermion

A new `<moebiusDomainWallQuarkAction>` with additional mandatory child elements `<b5>` and `<c5>` is defined (in addition to the existing `<domainWallQuarkAction>` which implicitly represents Shamir's form with fixed parameters $b_5 = 1$ and $c_5 = 0$).

Although Shamir's form can be viewed as a special case of the Möbius form (and no legacy ensembles of the former exist in ILDG), introducing two distinct actions is preferable (*c.f.* rule (A) of Sect. 3.1).

Examples: `e-moebiusDW.xml` and `e-shamirDW.xml`

3.4.5 QED gauge action

In order to describe the photon action in QCD+QCD, a new optional element `<photon>` is introduced between (and at the same level as) `<gluon>` and `<quark>`. The `<photon>` can have one or more child elements from the set of supported photon actions (with distinct names, c.f. rule (A) of 3.1. The initially supported set includes

- `<nonCompactQedSPhotonAction>` (used in [20])
- `<compactPlaquetteCstarPhotonAction>` (used in [18, 21, 22])
- `<compactTreelevelSymanzikCstarPhotonAction>` (used in [18, 21, 22])

This set can be easily extended later on, when needed, e.g. by `<nonCompactQedLPhotonAction>` and `<nonCompactMassivePhotonAction>`.

Analogous to gluon and quark actions, each photon action has a mandatory child with name `<photonField>`, and further elements as required to specify the functional form of that action. The element `<photonField>` should provide a complete specification of the integration variables used in the path integral. Therefore, depending on the action in which the `<photonField>` element occurs, it has further mandatory child elements (after the mandatory `boundaryCondition>` list). For instance, non-compact QED formulations have a mandatory `<gaugeFixing>` (with values `landau`, `coulomb`, etc.) and `<additionalConstraint>` (with values `qedS`, `qedL`, `qedTL`, etc.).

Examples: `slinc-qed.xml` and `rcstar.xml`

3.4.6 Fermion coupled to electromagnetic field

To describe the coupling of the quarks to the photon, distinct quark action names are introduced (following rule (A) of Sect. 3.1, as for the photon action). The initially supported set of actions includes

- `<fatLinkDerivNpNiCloverChargedQuarkAction>` (used in [20])
- `<npCloverCstarChargedQuarkAction>` (used in [18, 21, 22])

Each of these actions has an additional mandatory child element with name `<couplingToPhoton>`. This element has always a (first) child element `<charge>` to specify the charge of the quark and possibly further mandatory child elements (depending on the quark action in which it occurs), *e.g.* to specify the coefficient of an electromagnetic Clover term. The charge is specified not by value but with either `positive` or `negative`, since the actual value depends on the gauge action, compact or non-compact, and the boundary condition in the latter case.

Example: QCDSF (see `slinc-qed.xml`)

```
<fatLinkDerivNpNiCloverChargedQuarkAction>
   <glossary>...</glossary>
   <quarkField>...</quarkField>
   <numberOfFlavours>1</numberOfFlavours>
   <linkSmearing>...</linkSmearing>
   <kappa>0.124362</kappa>
   <cSW>2.65</cSW>
   <couplingToPhoton>
      <charge>positive</charge>
      <cSW>0</cSW>
   </couplingToPhoton>
</fatLinkDerivNpNiCloverChargedQuarkAction>
```

3.4.7 Exponential clover term

Quark actions with an exponential clover term are considered as distinct actions. Thus, new quark-action names should be specified for them (c.f. rule (A) of Sect. 3.1). Introducing `<npExpCloverQuarkAction>` (with the same structure as `<npCloverQuarkAction>`) resolves request (b-12).

Example: `e-npExpClover.xml`

3.4.8 License*

A "clear and accessible data usage license" is one of the FAIR principles, *c.f.* R1.2 of [19], and solves request (b-08). Therefore, `<license>` is introduced as **mandatory** sub-tree at top level and directly after `<management>`.

The proposed schema implementation allows to specify either a "standard" or a "custom" license:

- a standard license is specified by an (optional) `<licenseName>` followed by a mandatory `<licenseURI>`
- a custom license can be specified by either a `<customLicenseText>` or a `<customLicenseReference>` or both.

After the above child elements, an optional `<embargoEndDate>` can be added to indicate when the embargo period (*i.e.* "All rights reserved") ends and the specified license applies.

Moreover, as (last) child of the `<license>` sub-tree, an optional `<acknowledgment>` element is allowed. It has as children an optional `<annotation>` element and either a `<citation>` (to specify possibly requested or desired citations) or a `<templateText>` element, or both.

When converting legacy QCDmlEnsemble documents, a `<license>` element with default child `<customlicenseText>` and value "All rights reserved" is inserted, unless the collaboration specifies (or later on changes to) a less restrictive license.

Examples: `e-license*.xml`

3.4.9 Funding references

To enable markup and book-keeping of funding and computer-time grants, as requested in (b-09), an optional sub-tree `<fundingReferences>` is introduced at the top level and directly after `<license>`. The allowed forms of this sub-tree are a sub-set of those allowed by the DataCite schema [17]. Thus, this sub-tree can be directly used in an XML document which validates against the latter.

The children of the `<fundingReferences>` element are a (non-empty) list of `<fundingReference>` elements, which in turn can have the following children

- `<funderName>` (mandatory)
- `<awardTitle>` (optional)
- `<awardNumber>` (optional)

Examples: see `e-funding.xml`

3.4.10 Structure of `size` and `boundaryCondition` elements*

In previous QCDmlEnsemble versions, the lattice geometry was specified with arbitrary names and order of the directions, e.g.

```
<size>
  <elem>
     <name>my_direction_name</name>
     <length>32</length>
  <elem>
  ...
</size>
```

and the `<elem>` children of `<boundaryCondition>` were expected to be in the **corresponding** order. This had a high risk of mistakes in the ordering of the children of `<boundaryCondition>` and also made the `<size>` sub-tree somewhat lengthy.

In the proposed new implementation the geometry is specified in a more compact form as

```
<size>
  <x>32</x>
  <y>32</y>
  <z>32</z>
  <t>32</t>
</size>
```

where the ordering must be either `x y z t` or `t x y z`, and the elements `<z>` (or `<y>` and `<z>`) are optional to possibly allow only one or two spatial dimensions. Moreover, the children of `<boundaryCondition` must have corresponding element names (to reduce the risk wrong direction assignment), like

```
<boundaryCondition>
  <x>periodic</x>
  ...
  <t>antiperiodic</t>
</boundaryCondition>
```

with one of the two allowed orderings as in `<size>`.

4 Outlook

Several (backward-/forward-compatible) improvements of the schema have not yet been implemented and should be considered in future minor releases:

- improved and more detailed documentation
- timely adding of further actions following rule (A) when required to enable new uploads
- a fully consistent implementation of rule (A) should go along with a corresponding hardening of the schema, e.g. through further constraints on specific values of elements. (Example: `<treelevelSymanzikGluonAction>` has as parameters the four coefficients `c0` ... `c3`, but the schema does not enforce their unique values).
- possibly introducing a custom action as a free sub-tree. This should clearly be deprecated (because it destroys searchability and reduces the value of the metadata), but might serve as a stopgap solution in extreme cases.
- improving searchability for the action names that contain specific implementation. For example any kind of quark action derived from the clover fermion should be caught by search with "clover" or "Wilson". Several action names contains abbreviated keywords, for which some kind of search mechanism is desired.

5 Acknowledgment

We thank Agostino Patella, Ed Bennett, Paul Rakow, and Roger Horsley for helpful discussion on the markup for QCD+QED and BSM simulations, and Issaku Kanamori on the markup of the Möbius domain-wall fermion action.

References

[1] F. Karsch, H. Simma and T. Yoshie, "The International Lattice Data Grid – towards FAIR data," PoS **LATTICE2022** (2023), 244 doi:10.22323/1.430.0244 [arXiv:2212.08392 [hep-lat]].

[2] F. Di Renzo, "The International Lattice Data Grid (ILDG 2.0)," [arXiv:2401.14752 [hep-lat]].

[3] M. G. Beckett, B. Joo, C. M. Maynard, D. Pleiter, O. Tatebe and T. Yoshie, "Building the International Lattice Data Grid," Comput. Phys. Commun. **182** (2011), 1208-1214 doi:10.1016/j.cpc.2011.01.027 [arXiv:0910.1692 [hep-lat]].

[4] G. Bali, R. Bignell, A. Francis, S. Gottlieb, R. Gupta, I. Kanamori, B. Kostrzewa, A. Y. Kotov, Y. Kuramashi and R. Mawhinney, *et al.* "Lattice gauge ensembles and data management," PoS **LATTICE2022** (2022), 203 doi:10.22323/1.430.0203 [arXiv:2212.10138 [hep-lat]].

[5] ILDG Hands-on workshop, 14-16 June 2023, online, https://indico.desy.de/event/39311/

[6] Public gitlab repository of ILDG MDWG https://gitlab.desy.de/ildg/mdwg/qcdml

[7] Public gitlab repository of ILDG MDWG https://gitlab.desy.de/ildg/mdwg/file-format

[8] C. M. Maynard and D. Pleiter, "QCDml: First milestone for building an International Lattice Data Grid," Nucl. Phys. B Proc. Suppl. **140** (2005), 213-221 doi:10.1016/j.nuclphysbps.2004.11.116 [arXiv:hep-lat/0409055 [hep-lat]].

[9] P. Coddington *et al.* [ILDG Metadata Working Group], "Marking up lattice QCD configurations and ensembles," PoS **LATTICE2007** (2007), 048 doi:10.22323/1.042.0048 [arXiv:0710.0230 [hep-lat]].

[10] E. Bennett, "Proposed extension to the ILDG metadata and binary specifications to enable adoption for BSM physics"

[11] G.Bali, G. von Hippel, E.Scholz, H.Simma, "Request for extensions of the ILDG metadata and binary specifications to enable upload of CLS configurations"

[12] O. Kaczmarek and C. Schmidt for the HotQCD collaboration, "HotQCD Requests for Extension of ILDG metadata Schema and Binary File Format"

[13] JLQCD, Hands-on wrap-up I.kanamori and H.Matsufuru, additional document (added 2023.11.24)

[14] QCDSF-UKQCD-CSSM, Hands-on wrap-up

[15] Ryan Bignell, Hands-on wrap-up

[16] A.Rago for openLAT initiative, "Request for Extensions of ILDG Metadata Specifications" (added 2023.10.23)

[17] DataCite Metadata Schema 4.5, https://schema.datacite.org/meta/kernel-4.5/metadata.xsd

[18] B. Lucini, A. Patella, A. Ramos and N. Tantalo, "Charged hadrons in local finite-volume QED+QCD with C* boundary conditions," JHEP **02** (2016), 076 doi:10.1007/JHEP02(2016)076 [arXiv:1509.01636 [hep-th]].

[19] M. Wilkinson, M. Dumontier, I. Aalbersberg, *et al.* "The FAIR Guiding Principles for scientific data management and stewardship", Scientific Data. **3** (1), 160018 doi:10.1038/SDATA.2016.18

[20] R. Horsley, Y. Nakamura, H. Perlt, D. Pleiter, P. E. L. Rakow, G. Schierholz, A. Schiller, R. Stokes, H. Stüben and R. D. Young, *et al.* "Isospin splittings of meson and baryon masses from three-flavor lattice QCD + QED," J. Phys. G **43** (2016) no.10, 10LT02 doi:10.1088/0954-3899/43/10/10LT02 [arXiv:1508.06401 [hep-lat]].

[21] A. Patella, "QED Corrections to Hadronic Observables," PoS **LATTICE2016** (2017), 020 doi:10.22323/1.256.0020 [arXiv:1702.03857 [hep-lat]].

[22] I. Campos *et al.* [RC*], "openQ*D code: a versatile tool for QCD+QED simulations," Eur. Phys. J. C **80** (2020) no.3, 195 doi:10.1140/epjc/s10052-020-7617-3 [arXiv:1908.11673 [hep-lat]].