New XML I/O in Quantum Espresso.

Pietro Delugas

SISSA

Outline

- Need for standardized I/O formats
- Brief presentation of XML schema for pw.x
- Implementation
- Ongoing work and conclusions

Necessity of standardized I/O formats

Robust communication/storage with text files

- Defined standard for format description
- Schema publication shares efficiently any format changes.
- Automatic reader/writer synchronization.

Development and maintenance easiness.

- Possibility to use standard libraries for I/O.
- Automatic generation of source code.

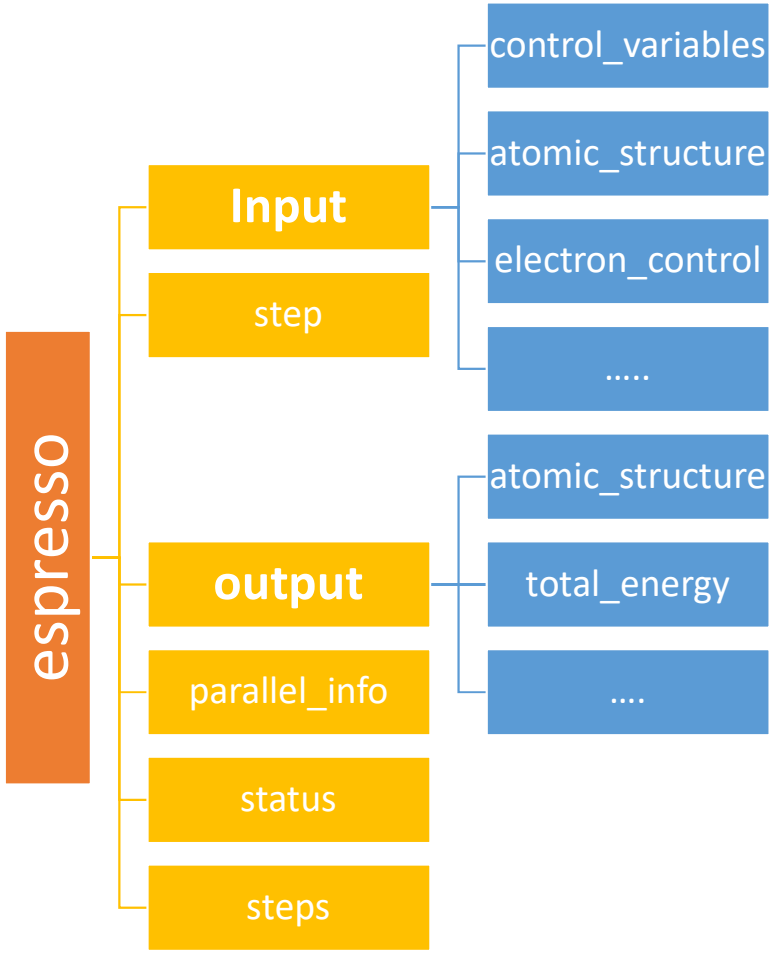
XML schema

input element (the only mandatory): contains all information for the calculation

step elements: if the case summarizes the results for each step of dynamics/optimization

output element: contains all small data yielded from the computation

parallel_info and status report information on how calculation was run



XML I/O in qe-6.0

Optional compilation

- Configure option --enable-xml
- `pw_restart_new.f90` is compiled instead of `pw_restart.f90`
- This same option enables other feature on new I/O in pw

Behavior

- Saves in <prefix>.save data-file-schema.xml
- Same file is saved in the outdir as <prefix>.xml
- If an XML input file is provided the input element is copied verbatim to the final

Implementation: basic modules

Basic libraries directly generated from the XML via a python script

qes_types_module

- Contains fortran data structure describing each complex element of the schema

qes_libs_module

- contains library functions for the creation, writing out and destructions of each fortran data type defined in qes_types

Implementation: qexsd modules for I/O

qexsd modules use qes_libs for writing and reading XML files.

qexsd_module

- Contains subroutines which are used in pw_restart_new for filling the output element at the end of the run.

qexsd_input

- Contains subroutines which are used at the beginning of the run to fill the input element at the beginning of the run.

qexsd_reader

- Contains subroutines for reading data from an XML file filling input and output elements.
- Used in pw_restart_new.f90 and read_file.f90 to extract data from XML files.

Implementation: where and when

read_file

- Is used at the beginning of a run to recover needed information from previous computations.
- Should be used by any code which is starting on top of a pw computation.

qexsd_input_init

- Is used at the beginning of a pw run to fill an input_type variable taking information from input_parameters global variables.
- In case

pw_write_schema (in pw_restart_new)

- Fills the output element
- Writes the XML file at the end of the run.

Implementation: where and when

step elements

- they are initialized at the end of each optimization/relax step;
- are allocated dynamically in a list and written out only at the end of the run
- maybe it would be better not to memorize them at each step but at each iprint step

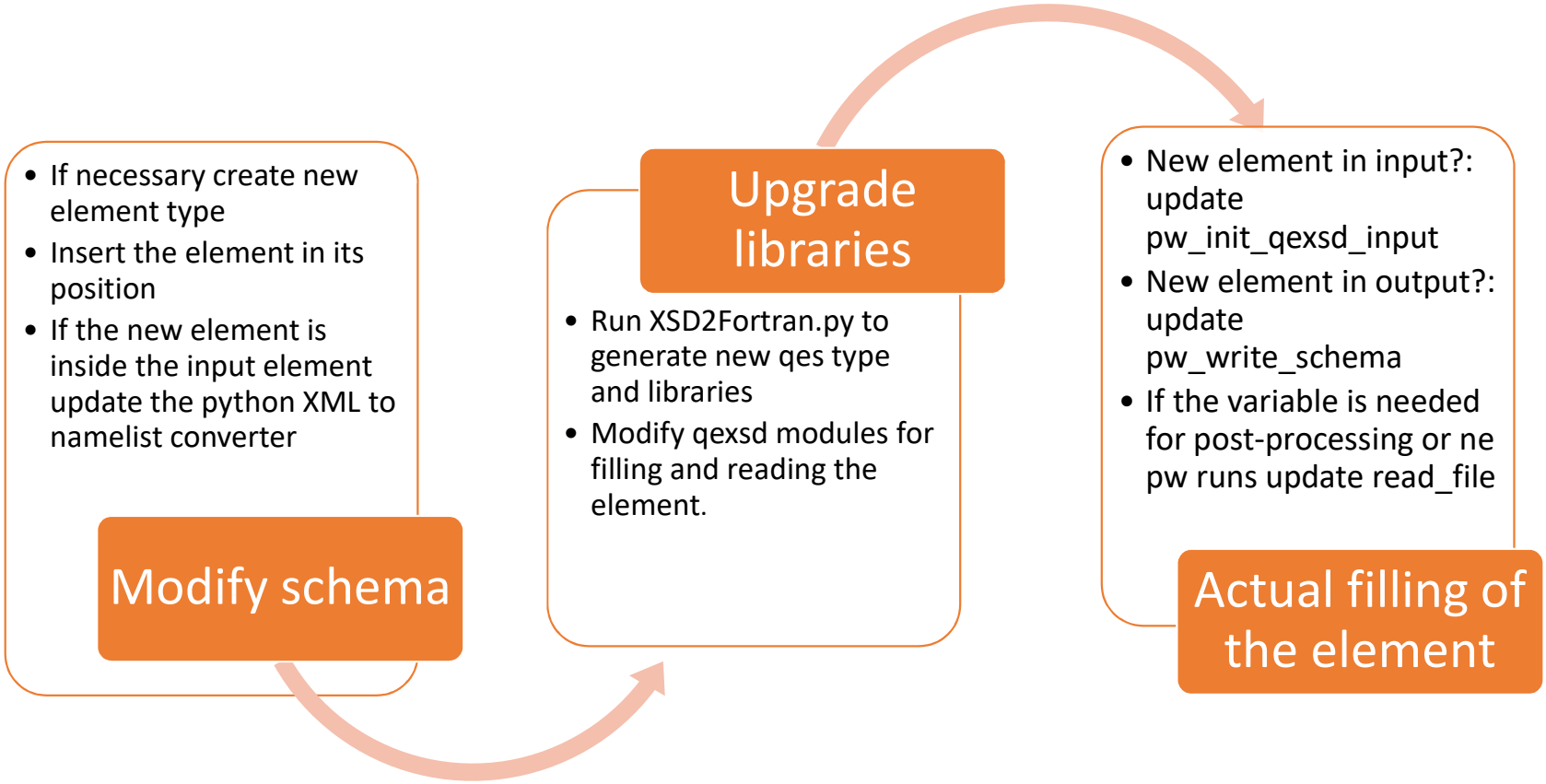
parallel_info and status

- They are filled in pw_write_schema at the end of the run

Warnings and error messages

- Warnings should be filled and memorized dynamically as for step elements.
- The errore subroutine should be modified in order to print out an XML file containg an errore element.
- These elements are not yet in the XML schema

How to modify the schema: new elements



Ongoing work

Adoption of Fox libraries for XML I/O

- Adapt XSD2fortran,py to Fox writing instructions
- Automatic generation of qes_read routines to be inserted in qes_lib_module
- Replacement of all other use of IOTK inside the code

Implementation of XML I/O for the other QE codes

- Schemes for phonon and neb and cp
- Scheme for pseudopotential files

People

Schemes

- Antonio Zambon
- Paolo Giannozzi
- Mauro Palumbo
- Davide Brunato

Python

- Giovanni Borghi
- Andrea Ferretti
- Davide Brunato

Fortran

- Andrea Ferretti, Giovanni Borghi
- Simone Ziraldo
- Paolo Giannozzi