35th EPS Conference on Plasma Phys. Hersonissos, 9 - 13 June 2008 ECA Vol.32D, P-2.112 (2008)

Data Structure for the European Integrated Tokamak Modelling Task Force

F. Imbeaux[1], J.B. Lister[2], G. T. A. Huysmans[1], L. Appel[3], W. Zwingmann[1], M. Airaj[1], V. Basiuk[1], D. Coster[4], B. Guillerminet[1], D. Kalupin[5], C. Konz[4], G. Manduchi[6], G. Pereverzev[4], Y. Peysson[1], L.-G. Eriksson[1], M. Romanelli[3], P. Strand[7], and contributors to the ITM-TF work programme

1. *Association EURATOM-CEA, CEA/DSM/IRFM, CEA Cadarache, France*
2. *Centre de Recherches en Physique des Plasmas, Association Euratom-Suisse, EPFL, 1015 Lausanne, Switzerland*
3. *UKAEA Euratom Fusion Association, Culham Science Centre, Abingdon, OX14 3DB, United Kingdom*
4. *Max-Planck-Institut für Plasmaphysik, EURATOM-IPP Association, Garching, Germany*
5. *Institut für Energieforschung – 4 (Plasmaphysik), Forschungszentrum Jülich GmbH, EURATOM Association, D-52425 Jülich, Germany, partner in Trilateral Euregio Cluster*
6. *Associazione EURATOM-ENEA sulla Fusione, Consorzio RFX, Corso Stati Uniti 4, 35127 Padova, Italy*
7. *Euratom-VR, Chalmers University of Technology, Göteborg, Sweden*

1. Introduction : broadening the scope and flexibility of integrated modelling

The Integrated Tokamak Modelling (ITM) Task Force aims at providing a suite of validated codes for the support of the European fusion program, including the preparation and analysis of future ITER discharges [1]. These codes will work together under a common framework, resulting in a modular integrated simulator which will allow a large variety of simulation types. The ambition here is to go one important step beyond the present state of the art in terms of i) simulation flexibility and ii) scope of the physics problems that can be addressed. Present integrated tokamak simulators world-wide are organised around a set of transport equations. They already present some nice features for individual model testing in terms of data integration, since they gather in their own internal format most of the data needed for input to a source term or a transport coefficient model. Nevertheless the variety of possible workflows remains quite restricted to their main function : solving time-dependent transport equations. The workflow is defined as the suite of calculations done during the simulation.

The goal of the ITM is to provide an integrated tokamak simulator featuring :

- Maximum flexibility of the workflow : the user chooses the physics modules to be used, defines how they should be linked and executed.
- Complete modularity : the codes are organised by physics problem (e.g. equilibrium solvers, RF wave propagation solvers, turbulence solvers). All modules of a given type are fully interchangeable (standardized code interfaces).
- Comprehensive tokamak modelling : the variety of addressed physics problems could cover the full range from plasma micro-turbulence to engineering issues, including the use and modelling of diagnostic data.

2. A standard physics-oriented format : Consistent Physical Objects

Modularity being a critical requirement, the codes are organised by physics problem. Though different assumptions / models can be used when solving a given physics problem (e.g. calculating anomalous transport coefficients from the plasma profiles), it is possible to define a general set of input / output that should be used / produced by any model solving this problem. A key point is that this definition must be general enough to be relevant for any physics workflow. This non-trivial task results in structuring the ITM data into standardised blocks that become the natural transferable unit between modules. Being standard (identical for all modules solving a given physics problem), they satisfy the modularity requirement. Being defined by the input / output logics of a physics problem, they are the natural transferable unit in a physics workflow. Being produced as a whole by a single module, all data in the block are internally consistent. These physics-oriented transferable data units are named Consistent Physical Objects (CPOs). The physics modules communicate with the others only by exchanging CPOs.

To clarify, let's give here a few examples of CPOs already defined in the present ITM data structure :

- "equilibrium" describes the plasma equilibrium (including 2-D maps of the magnetic field)
- "coreprof" is a set of radial profiles corresponding to the usual core transport equations for poloidal flux, density, temperatures, toroidal velocity
- "coresource" is a general source term to be used by the core transport equations
- "rfwaves" describes the propagation of Radio Frequency Waves, including sub-trees for the various possible methods for solving the physics problem (ray-tracing, beam-tracing, full wave)

In addition to their internal physics consistency, the CPOs structure contains also nodes describing data management information : comments and origin of the data for traceability, method and location of the actual data storage for this CPO (used internally by the Universal Access Library to GET/PUT the CPO). At the top level of the data structure, a general CPO "topinfo" contains informations on the database entry (the full set of CPOs that forms the plasma description) and in particular which workflow has been used to produce it. Therefore the CPOs are fully self-describing objects that provide a natural guarantee of consistency and traceability.

3. CPOs and workflows

Designing the CPO structure requires to think of the interaction between modules in a physics workflow. For instance an MHD linear solver requires information on the plasma equilibrium, which is provided by an equilibrium solver. Thus an "equilibrium" CPO is an output of an equilibrium solver and an input to the MHD solver. When assembling modules in a workflow, their input / output must be compatible – otherwise the link between modules has no physical meaning. Outside of this requirement, the workflow is fully flexible and designed by the user. For instance, a transport equation solver must use as input a source term CPO. How this source term is produced from the combination of the various heating sources is fully flexible and chosen by editing the workflow.

Editing the workflow is done using the KEPLER framework [2]. The user defines links between physics modules. Those links represent both the workflow (the order in which the modules should be executed) and the dataflow : each link represents a CPO which is transferred from one module to the other. The KEPLER system knows what kind of CPOs each physics module needs for input / output and controls the compatibility of the links created by the user.

4. Experimental CPOs, machine configuration

In order to validate physics model against present experiments, the ITM codes must be able to use experimental data. Two main usages can be considered. First, when comparing the result of a simulation to an experiment, it is better to use synthesised diagnostic signal from the simulation results than to compare to a pre-fitted profile (the method used to fit/invert profiles from experimental data may induce some bias in the comparison). Second, many physics predictions require the knowledge of the tokamak configuration (e.g. to model a Radio Frequency heat source requires some knowledge of the antenna). Therefore the data structure and the CPO concept are extended to describe i) diagnostics and their measured data and ii) machine configuration.

The principles of the “experimental” and of the “simulation” CPOs are the same : they are transferable units in the ITM workflow, and they are internally consistent. To enforce this latter point, experimental CPOs contain both the machine configuration and the related pulse-based time-dependent data. For instance, the interferometer CPO “interfdiag” gathers the time-dependent measurement of the line integrated density AND the description of the line-of-sight geometry.

While in existing codes the machine configuration is usually hard-coded, which is a problem in view of multi-tokamak modelling and modularity of the codes, the ITM data structure describes it in a generic and standard format. Therefore ITM modules can be completely machine-independent, which is a significant progress and a requirement for multi-device validation exercises.

Usually, each diagnostic has its own acquisition strategy (sampling time). Similarly, each CPO has its own time base. This property is also important for “simulation” CPOs, since the different physics problems solved during the simulation may not be solved at the same time slices.

5. CPOs and the physics user

The CPO approach provides the physicist with a sophisticated and powerful method for coding the input / output of his physics model. Depending on the kind of physics problem he solves, his module has a certain list of input and output CPOs. Libraries describing the internal structure of each CPO are provided by the ITM support team for various languages (Fortran 90, C++, Java). Thus integrating an existing physics module in the ITM framework requires a quite minimal effort. As an example, we indicate below the generic format for an ITM physics module (Fortran 90 example) :

```
Subroutine Physics_module(CPOin1,….,CPOinN, CPOout1,…, CPOoutM)
  use euitm_schemas    ! provided library containing the type definitions of all CPOs
  ! Declaration of the CPO variables must be as :
  type (type_equilibrium) :: CPOin1    ! for one time slice,  OR
  type (type_equilibrium), pointer :: CPOin1(:) ! if the code handles several time slices
  ! Extraction of the CPO variables from the input requires the knowledge of the CPO
internal structure (see ITM documentation online):
  My_plasma_current = CPOin1%global_param%i_plasma
  ! Begin the physics calculation
  … (this is the physics calculation, which can be used without modification in the
ITM framework)
  ! Fill the output CPOs from the internal physics code variables
  CPOout1%global_param%i_plasma = my_calculated_plasma_current
End subroutine
```

This adaptation is the only one the physicist has to do. The physics module will then receive the proper input CPOs and send back the calculated output CPOs to the next module in the workflow without knowing the sophisticated machinery that schedule the run and makes the actual data transfer between modules (which may be written in different languages). As soon as one has understood the logics of the CPOs, the integration of the physics code in the ITM framework is straightforward.

6. What makes it happen

The purpose of this paper is to describe the general logic of the ITM data structure for the physicist. However in this paragraph we address briefly the internal software that allows dealing with this sophisticated data structure. We stress once more that the technologies presented here are invisible to the physics user (if he does not want to know about them).

Since it aims at a full description of a tokamak (plasma physics quantities + subsystems characteristics) the ITM data structure has rapidly become quite detailed and large. Several tools (CPO type definitions and access libraries in various languages, HTML documentation …) must be derived from this huge data structure without errors. In order to make this feasible, and following the pioneering ideas reported in [3], we define the ITM data structure using XML schemas. The XML language is a world-wide used standard supported by the World Wide Web Consortium [4]. It allows coding easily the hierarchy of the nodes, their type and properties, as well as an integrated documentation for the physics user (physical definition, units, dimensionality in case of arrays). This XML schema description of the data structure is the source from which all tools and documentations are derived. Translation scripts in XSL language allow generating dynamically the ITM tools related to the handling of the data structure. As quoted above, key tools like the CPO type definition and CPO access libraries in various languages (called the Universal Access Layer [5]) are generated with these methods.

7. Conclusion

The need to broaden the scope and flexibility of integrated tokamak modelling has led to the definition of sophisticated, comprehensive, physics-oriented data structures. The Consistent Physical Objects define the common standard for representing and exchanging the properties of a physics problem. Equilibrium, linear MHD stability, and core transport were the first three topics addressed. Data structures have already been finalised for these and are being expanded to address non-linear MHD stability, edge physics, turbulence, and heating and current drive. This is an unprecedented collective effort to which the European fusion physics community should be more widely associated.

In parallel to the development of the physics concept, powerful tools are provided in order to manipulate the data structure. Most of these tools are used internally by the ITM framework and the physics user can freely ignore them. The ITM TF tokamak simulator is a significant step beyond existing integrated modelling codes in terms of workflow flexibility. The ITM data structure and its associated tools are key elements for allowing this step and an easy and powerful way to design integrated simulations for the physicist.

More information can be found on the Expert Area of the ITM TF website : http://www.efda-taskforce-itm.org/. The complete HTML documentation of the ITM data structure is posted in this password-protected area.

8. References

[1] A. Becoulet, P. Strand, H. Wilson, M. Romanelli and L.G. Eriksson, Comput. Phys. Commun. 177 (2007) 55-59

[2] B. Guillerminet et al, Fusion Engineering and Design 83 (2008) 442–447

[3] J.B. Lister et al, *Creating an XML driven data bus between fusion experiments,* Proceedings ICALEPCS 2005, Geneva, P02.093

[4] World Wide Web Consortium : http://www.w3.org

[5] G. Manduchi et al, Fusion Engineering and Design 83 (2008) 462-466