Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Object-Oriented Modelling & Simulation of Power Plants with Modelica

Francesco Casella* and Alberto Leva*

Abstract— **This paper presents and discusses the application of the object-oriented modelling paradigm to thermo-hydraulic systems, with particular reference to fossil-fired and nuclear power plants. As a result, a particular modelling approach is proposed and motivated. The paper also presents the ThermoPower Modelica library, developed at the Politecnico di Milano along the proposed approach, and made available to the scientific and professional community within the terms of the Modelica license. Some application cases involving ThermoPower are briefly reported.**

I. INTRODUCTION

Dynamic simulation is a very important tool in the design of power plant control systems, particularly when innovative plants, or innovative control strategies are considered. Simulation can play a role from the initial design stages, when the control strategies and the required instrumentation are evaluated, to the validation of the controller tuning, up to the plant commissioning phase, not to mention personnel training. Dynamic modelling and transient analysis efforts can be very cost-effective, both during design, when it can catch flaws which would result in later costly interventions, and during the commissioning phase, when the savings in terms of reduced down-time can be huge.

Recent advances in object-oriented modelling of dynamical systems, and in particular the development of the Modelica language [1], bring new possibilities in this field, allowing the fast development of system simulators which can be tailored to the different needs of the design process, while maximising the re-use of existing information and knowledge. The flexibility of the object-oriented approach is particularly well-suited to support model-based control system design methodologies, such as Model Predictive Control. Currently there is a wide gap between full-scale modular plant models used for simulation, and reduced-order hand-coded models used for controller design; as available computing power increases, and Modelica compilers become more efficient, this gap will presumably become narrower in the near future, until modular (albeit simplified) first-principle models could possibly be used directly for controller design.

The paper is structured as follows: Section 2 reviews the key features of object-oriented modelling of dynamical systems; in Section 3, the advantages of using object-oriented modelling in power plant simulation are discussed; Section 4 introduces the ThermoPower Modelica library; example applications are reviewed in Section 5.

*Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy. casella@elet.polimi.it, leva@elet.polimi.it

II. OBJECT-ORIENTED PHYSICAL SYSTEM MODELLING WITH MODELICA

The Modelica language was introduced in 1997 [1], as the product of an international cooperative effort to define an object-oriented language for the modelling of generic physical models, described by algebraic and differential equations. The features of the language which are relevant in the context of this paper are summarised here.

A. A-causal, declarative modelling

The model of each physical component (e.g. a pipe, a pump, a valve, or a turbine) is described by a set of algebraic, differential, and event-triggered difference equations; these describe how the modelled object behaves, rather than how the equations are to be numerically solved. The boundary conditions (pressures, temperatures, flow rates) are not necessarily declared a-priori as input or outputs: this is essential to achieve truly object-oriented modelling of physical systems, since the model of a physical is always the same, irrespective of what is connected to it. This marks a fundamental difference with conventional block-diagram-oriented simulation languages, such as Simulink, in which each model must have definite input and output signals.

B. Code transparency

The declarative approach allows to write the model code in a way that tightly matches the way equations are written on the paper, without bothering how the equations will eventually be solved. This greatly eases the model development, documentation, modification and reuse, thus providing a significant advantage over specialised power plant simulation packages. These in fact usually provide "closed" models, which can be very difficult or even impossible to inspect, in order to understand what's inside, and even harder to modify, in order to adapt them to one's specific needs.

C. Encapsulation

The models of system components are connected through rigorously defined interfaces or connectors (e.g. fluid connectors with pressure, flow rate, and enthalpy, or heat connectors with temperatures and heat fluxes). Any two components with compatible connectors can be bound together, regardless of their internal details. This feature is essential to re-use models, and to easily replace subsystems with more or less detailed counterparts, without affecting the rest of the system.

D. Inheritance

Model libraries can be given a hierarchical structure, in which more complex models are obtained from basic models by adding specific variables, equations or even models. It is then possible to factor out the common behaviour of a family of components (e.g. valves, or pumps) in a parent model, and then to define child model which add their specific variables and equations. It is also possible, e.g., to model objects with replaceable fluid models, by separating the component model equations from the fluid model equations.

E. Multi-physics modelling

The Modelica language allows modelling of generic dynamical systems. It is then straightforward to combine physical models belonging to different engineering domains with continuous- or discrete-time control systems models.

F. Reusability

A-causal modelling, encapsulation, and inheritance are strong incentives toward reuse of modelling knowledge inside simulation projects. At the component level, it is often possible to re-use models provided by standard libraries, while developing a few specific components with ad-hoc (and possibly proprietary) modelling, wherever needed. At the system level, it is possible to easily manage a family of models with different accuracy and simulation speeds; in most cases, this can be obtained by slight variations of a "reference" system model. This greatly helps to maintain the consistency of models as they evolve throughout the project life cycle, as modifications and improvements applied to the reference model are automatically inherited by simplified ones.

III. OBJECT-ORIENTED MODELLING OF POWER PLANTS

A. Flexibility

Models of power plant components can have a widely varying complexity, depending on the desired degree of detail. On the other hand, their boundary connections essentially fall under three categories: fluid flange connections (like the inlet and outlet flanges of a pump), thermal transfer between zero- or one-dimensional objects (such as heat exchangers), and mechanical flanges (such as a turbine or a compressor shaft). It is therefore possible to define standard connectors for these types of interfaces, or even to re-use interfaces which are pre-defined in the Modelica standard library.

The declarative approach leaves almost complete freedom as to what equations should be used to describe a specific component; on the other hand, any two components can be connected, as long as the standard connectors are used.

B. Modularity

The object-oriented approach is highly modular. This means, first of all, that it is possible to build the model of a plant unit by connecting the models of its physical components in any way which makes physical sense; secondly, it allows to build a plant model by connecting the unit model, with an arbitrary number of hierarchical levels.

Besides that, the advanced features of the Modelica language allow to define *replaceable* components, which can be substituted by more or less detailed counterparts, as long as they have the same interface, i.e. the same connectors and parameters. It becomes then much easier to manage a whole family of simulators of the same plant, each one characterised by a level of detail which is appropriate to a specific simulation task. For example, it is possible to substitute a very accurate water property model with a much simpler one, when the system is going to be simulated around a certain operating point; or, it is possible to substitute the model of whole plants sub-units (e.g., the feedwater systems) with more idealised counterparts (e.g. an ideal flow source) in a systematic way.

IV. THE THERMOPOWER MODELICA LIBRARY

The ThermoPower library has been developed to provide basic components for the modelling of power plants. The scope of the library is thus narrower than that of other Modelica libraries for generic thermo-fluid systems, such as ThermoFluid [2] or the forthcoming Modelica.Fluid standard library [3]. On one hand, this allows to make some basic simplifying assumptions on the nature of the fluids and their phenomena: e.g., turbulent flow is always assumed to compute pressure drops, which is a good choice in typical power plant components handling water and gases, but it's not in petrochemical plants, where oil is heavily involved. On the other hand, it makes it possible to put more detail and sophistication where it is really needed: for example, pipe models use finer discretisation grids for enthalpies than for pressures and flow rates, since wave propagation phenomena can usually be neglected when dealing with control-relevant power plant dynamics.

The library, described in more detail in [4], [5], is an open source project: the source code is freely available online [6].

A. Library Structure

The library is structured into 5 packages. The package `Water` provides models of components using water/steam as working fluid, while the package `Gas` provides models of components using ideal gas mixtures. The default water/steam model, based on the IAPWS-IF97 formulation, and the default ideal gas mixture model, based on a NASA property database, are provided by the `Modelica.Media` library [3], which is a part of the standard `Modelica` library. Those models can be replaced by any other fluid model, provided it is equipped with a Modelica.Media-compliant interface.

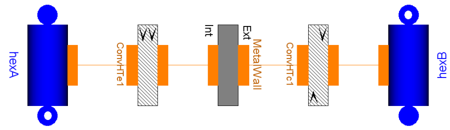

Fig. 1. Model of a counter-current heat exchanger

The `Thermal` package provides basic building blocks to describe heat transfer phenomena. The `Electrical` package provides highly idealised models of electric generators and power grid, to describe the plant boundary conditions; note that more accurate models from specialised electrical machinery libraries could be used as well, in case higher-frequency electro-mechanical dynamics becomes of interest.

B. Elementary components

The `Water` and `Gas` library contain models of elementary components such as prescribed pressure reservoirs, prescribed flow generators, valves, pumps, compressors, turbines, mixers, headers, drum boilers, three-way connections, and generic pressure drops. Those models are based on basic mass and energy balances, and on standard pressure vs. flow rate characteristic curves.

C. Components for heat exchanger modelling

Heat exchangers can have widely different configurations (co-current, cross-flow, counter-flow), and involve two or more fluid flows, (water/steam and/or flue gases). Basic building blocks are provided to build those models.

`Flow1D` models a 1-dimensional fluid flow with heat transfer from the boundary; the model is based on distributed-parameter mass, momentum and energy conservation equations, discretised by a finite-volume method. The `Flow1Dfem` model is obtained from the same basic balance equations, using a finite element method instead. The `Flow1D2ph` model is a finite volume model considering averaged densities in the neighbourhood of phase changes, to avoid non-physical simulation artifacts due to phase change discontinuities at the model nodes. All of these models can be extended through inheritance, by adding further equations to compute heat transfer coefficient according to a number of well-established empirical correlations (like Dittus-Boelter's correlation for one-phase flow, or Chen's correlation for boiling heat transfer).

Basic models for co-current and counter-current heat transfer configurations are provided, as well as simple models of metal wall, accounting for thermal resistance and thermal capacitance. These can be used together with the 1-D flow models to build heat exchanger models such as the one in Fig. 1. More complex heat transfer configuration can be easily modelled, by writing ad-hoc heat transfer modules, relating temperatures and heat flows between the different objects, and re-using the existing fluid flow models.

V. APPLICATIONS

Once a Modelica plant model has been built, it can be turned into a simulator by a Modelica tool, such as Dymola [7], and used for many different purposes. First of all, it is possible to run it open-loop against experimental data, to assess the validity of the model itself. The model of a control system (either continuous-time or digital) can then be connected to it, and closed-loop simulations can be run, to analyze different plant control strategies, and eventually to tune the controller parameters. In case a large number of simulations must be run, such as in sensitivity studies, it is possible to replace some parts of the model with simpler counterparts, to change the fluid models, or to change the number of nodes of distributed-parameter components. Besides simulating a model, it is also possible to obtain the A,B,C,D matrices of its equations, linearised around one or more operating points; this can be very useful to design model-based controllers based on linear (or piecewise-linear) models. It is also possible to automatically obtain real-time, fixed-time-step simulators from the open-loop plant model, to be used for hardware-in-the-loop testing of the control system or for operator training.

Three applications of the library are now briefly presented, and will be illustrated in the interactive session, along with the library itself.

A. Detailed boiler model validation

All the library models have been tested in very simple configurations, to check their correctness. Besides that, the ThermoPower library was validated against experimental data coming from the physical model of the evaporating section of a heat-recovery boiler, with a power scaling factor of 1:600. Details on this validation are given in [5]. Suffice here to say that the simulation model represents the circulation loop in detail, while non-equilibrium phenomena are accounted for in the model of the drum. The model has 50 state variables and 288 nontrivial equations, showing that the proposed approach allows to treat efficiently cases of realistic complexity. The diagram of the Modelica model obtained by connecting ThermoPower models is shown in Fig. 2.

A sample of the validation tests is reported here, namely a negative throttling valve step at low load. Figures 3 and 4 show the drum pressure and level transients: notice the good agreement between the model output and data.

In particular, the non-equilibrium phenomena represented in the drum model allow to reproduce both low- and mid-frequency dynamics in the pressure responses correctly. Also the effects of thermal exchanges between the fluid in the drum and the drum metal wall were investigated, showing that the corresponding heat transfer coefficient has a significant influence on the superheated steam temperature. This phenomenon is often neglected in the simulation models proposed in the literature.

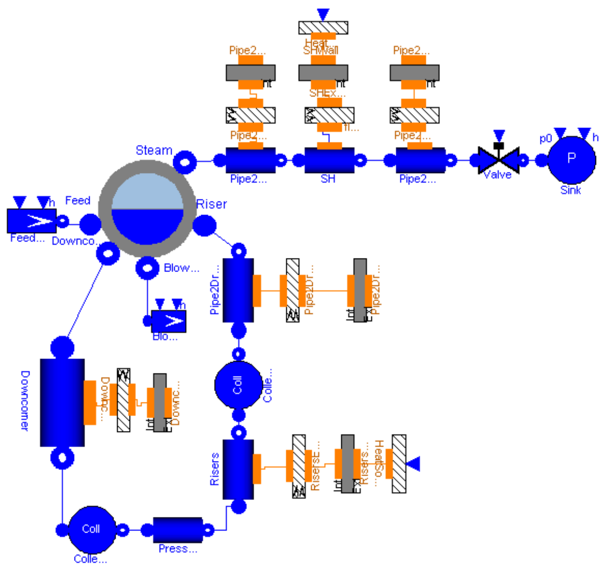

Fig. 2. Modelica diagram of the boiler model.

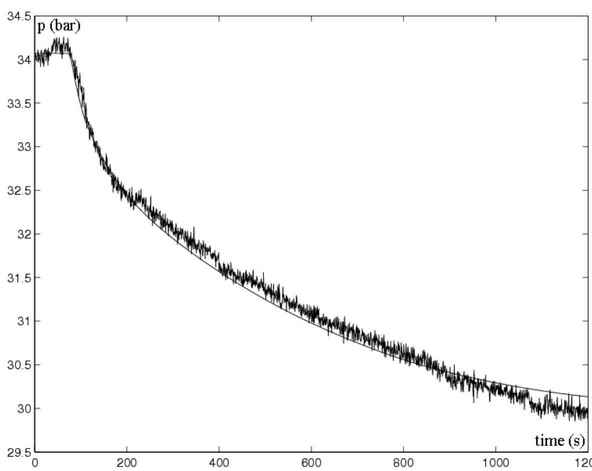

Fig. 3. Drum pressure transient for a throttling valve step leading to a 13% pressure reduction at low load (simulated vs. experimental data).

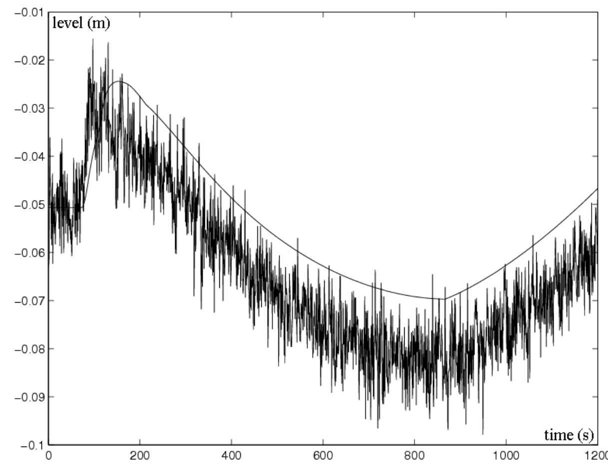

Fig. 4. Drum level transient for a throttling valve step leading to a 13% pressure reduction at low load (simulated vs. experimental data).

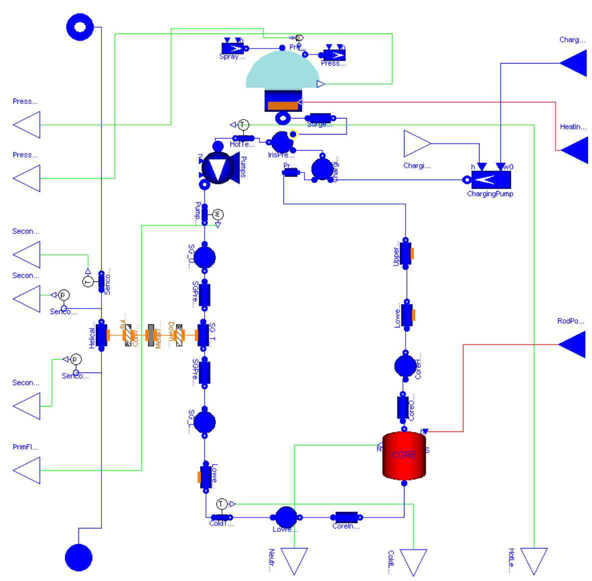

Fig. 5. NSSS model diagram

B. Nuclear power plant simulation

The IRIS plant is an innovative pressurised-water nuclear reactor, with an innovative integral design: all the components of the steam generator (nuclear core, control rods, primary circuit, pressuriser, circulation pumps, steam generator) are included in the primary containment vessel.

Very detailed dynamic models of the plant are available, using certified simulation codes such as RELAP. These codes are mainly targeted at the simulation of extreme transients for safety analysis; on the other hand, they are way too slow to be used for control system design, and very difficult to integrate with realistic models of the control system. The ThermoPower library, coupled with additional components modelling the neutronic kinetics and the fuel rod thermal dynamics, has then been used to implement system-level simulators of this plant, to support the preliminary study of the plant control strategies [8].

Fig. 5 represents the diagram of the so-called Nuclear Steam Supply System (NSSS). Hot water is pumped through the primary side of the steam generator, passes through a lower headers, flows through the core, where it gets heated, and then through the risers up to the pressuriser, and then back to the pumps. The secondary side is connected to the TGFWS model, containing the turbine, electric generator, and feedwater system models.

The top-level model (shown in Fig. 6) contains the two main plant units, together with the control system and supervisory system models; the latter two are connected to the plant via bus signal connectors. Note that this model is an abstract (or *partial* in Modelica terms) model, describing the high-level structure of the simulator. A specific simulation

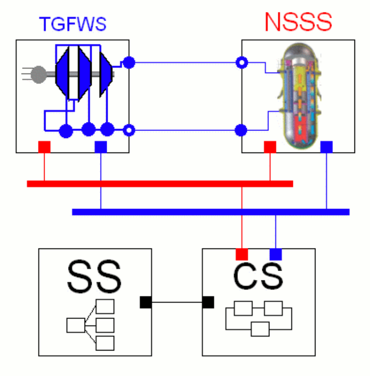

Fig. 6. Top-level simulator structure

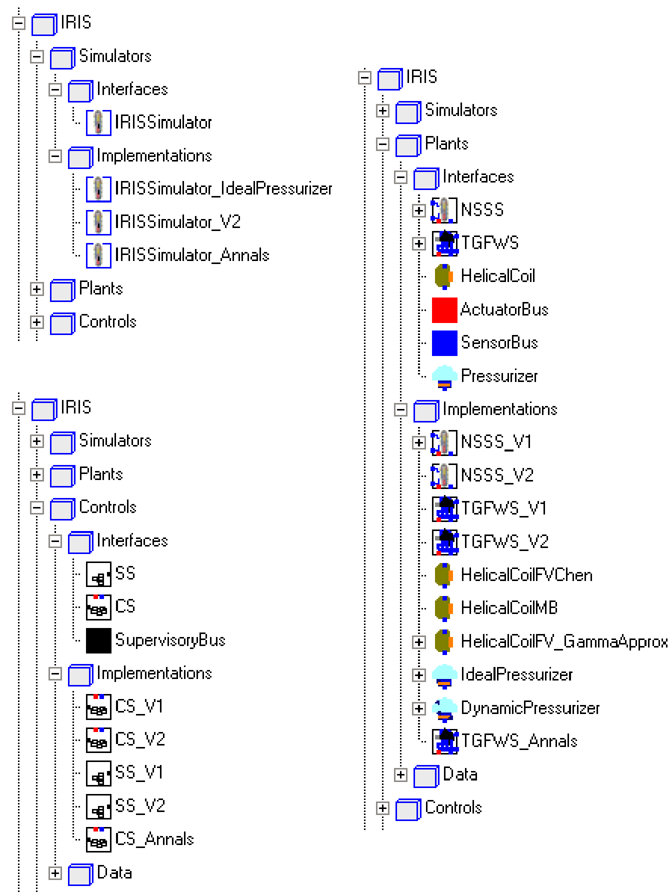

Fig. 7. The IRIS Simulator Library

model is then obtained by instantiating the partial model and replacing each abstract unit with a specific implementation, taken from a *simulator library* (Fig. 7). The NSSS model implementation can be further customised by specifying the fluid model used for the primary and secondary loops, as well as the number of nodes in the core and once-through steam generator models, and the steam generator model itself. Note that, if an incompressible fluid model is chosen for the primary loop, the fast states related to the pressure dynamics automatically disappear from the model, thanks to symbolic manipulation by the Modelica compiler. Dimensional and operarting data are put in data sheets contained in the Data sub-package; those data sheets can then be referenced when instantiating a specific simulation model.

Using this architecture, it is possible to maintain a very large set of potential simulation models in a consistent state throughout the entire project lifetime, by avoiding any unnecessary duplication of data and models.

Fig. 8 shows the results of the simulation of 10% step load change requests.

C. Combined-cycle plant startup simulation

In the new scenario originated by the deregulation and restructuring of the power market, the fast startup of power plants, possibly assisted by advanced, model-based controllers, is becoming an increasingly attractive topic (see, e.g., [9], which has been recently implemented on full-scale, operating plants). A research project is currently being carried out at Politecnico to evaluate the potential reduction in start-up times for typical large combined cycle units (250 MWe + 130 MWe) installed on the Italian network. The start-up time of combined cycle plants is essentially limited by the thermal stresses of the most critical points of the steam units, i.e. the superheated steam collectors and the sections of the turbine shafts corresponding to the first blade row.

A detailed first-principle model of the plant, including models of thermal stresses in those parts, has been developed and is being calibrated and tested at the time of this writing. The model (see Fig. 9 for a high-level diagram) includes a very simple description of the gas turbine unit (which is much faster than the steam unit), a heat recovery boiler including high and intermediate pressure superheaters, reheaters, evaporators and economisers, a model of the high and intermediate pressure drums, and a model of the steam turbine generator, complete with bypass valves. The low-pressure part of the steam generator has been modelled as a simplified boundary condition; although it is essential to achieve high energy conversion efficiency, its influence on the thermal stress of high-pressure, high-temperature components is in fact negligible. The reference model has currently over 2000 variables and 130 states and is able to simulate a warm start-up, i.e. from the reignition of the gas turbine after a nightly stop, through turbine startup and synchronisation, up to full load operation. Faster, more simplified versions will then be developed and tested against this reference model, to assess if they are accurate enough to predict the stress peaks during the whole startup transient, while requiring less computational effort. All models will be eventually released as a part of the ThermoPower library.

The resulting plant models, combined with numerical optimisation algorithms, will be used to compute the time-optimal transients, under the constraint that the peak level of

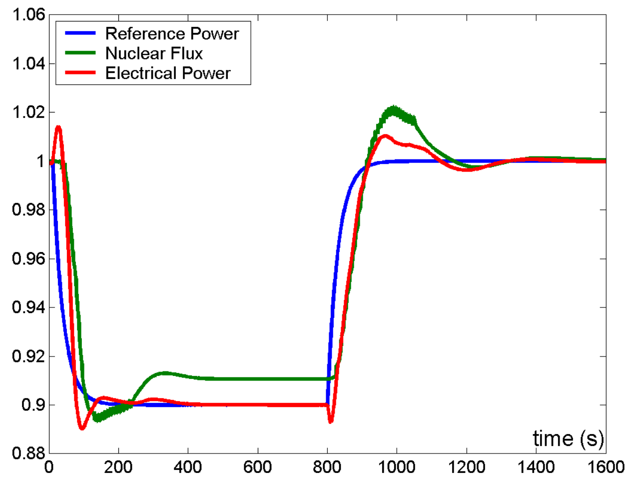

Fig. 8. Controlled variable response

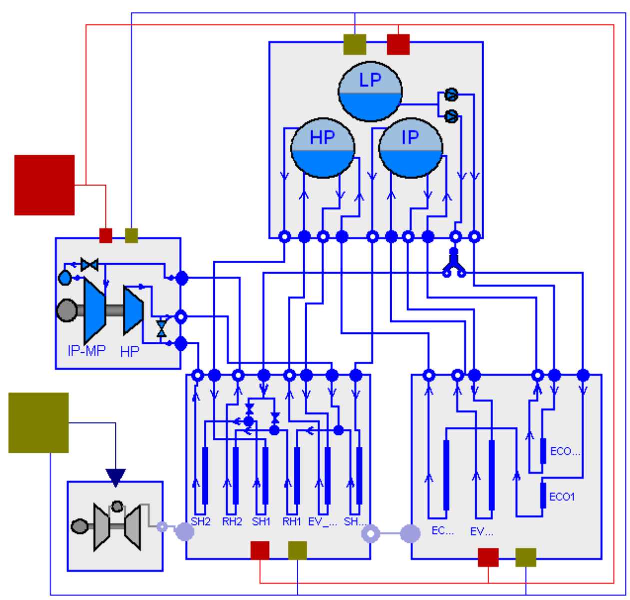

Fig. 9. Diagram of the combined-cycle plant model

stress which is currently attained during start-up maneuvers is not exceeded. This open-loop result, obtained assuming perfect knowledge of the model and of all its variables, will represent a theoretical upper limit to the performance of real-life, real-time controllers; significant results in terms of achievable time reduction would then motivate further research toward the design and implementation of such controllers.

VI. CONCLUSIONS

The application of object-oriented methods and tools to power plant simulation have been presented in this paper, especially focusing on the ThermoPower Modelica library. The library has been conceived in order to emphasise model readability and extensibility; at present it still contains a limited number of components, which nevertheless allow modelling a wide range of different physical components. It should be stressed that the Modelica language allowed translating sophisticated modelling concepts into working code with remarkable ease. The library is available to the public, and is open to contribution from other research groups [6].

Some applications involving the ThermoPower library have been reviewed, to show the potential of object-oriented modelling in this field.

Further research work is underway to investigate the proposed modelling principles, and their application to control system design, in particular in the field of model-based methodologies such as non-linear model predictive control. As a consequence, work is in progress to extend ThermoPower, to address new application cases, and to introduce reasonably accurate and reliable descriptions of the control systems employed in real-world applications. Future applications could also possibly involve less conventional power generation units, such as micro-turbines, waste incineration plants, gas turbines with external combustion, and low-temperature organic Rankine cycles, which will become more and more attractive as the oil and gas prices continue increasing.

VII. ACKNOWLEDGMENTS

First of all, the authors would like to remember the late professor Claudio Maffezzoni, whose inspiration and research was fundamental in starting this work. Many thanks go to Antonio Cammi, Marco E. Ricotti, and Francesco Schiavo, who also worked on the IRIS plant application. The combined-cycle plant startup project is supported by CESI under a ‘Ricerca di Sistema’ contract.

REFERENCES

[1] S. Mattsson, H. Elmqvist, and M. Otter, “Physical system modeling with modelica,” *Control Engineering Practice*, vol. 6, pp. 265–271, 1998.

[2] H. Tummescheit, J. Eborn, and F. Wagner, “Development of a modelica base library for modeling of thermo-hydraulic systems,” in *Proc. Modelica Workshop 2000*, Lund, Sweden, 2000, pp. 41–50.

[3] H. Elmqvist, H. Tummescheit, and M. Otter, “Object-oriented modeling of thermo-fluid systems,” in *Proc. 3rd Modelica Conference*, Linköping, Sweden, 2003, pp. 269–286.

[4] F. Casella and A. Leva, “Modelling of distributed thermo-hydraulic processes using modelica,” in *Proc. MathMod Conference*, Vienna, Austria, 2003, pp. 514–521 (to appear in *Mathematical and Computer Modelling of Dynamical Systems*).

[5] ——, “Modelica open library for power plant simulation: design and experimental validation,” in *Proc. 3rd Modelica Conference*, Linköping, Sweden, 2003, pp. 41–50. [Online]. Available: http://www.modelica.org/Conference2003/papers/h08_Leva.pdf

[6] The ThermoPower library system home page. [Online]. Available: http://www.elet.polimi.it/upload/casella/thermopower/

[7] “Dymola, dynamic modelling laboratory v. 5.3b,” Dynasim AB, Sweden, 2003.

[8] A. Cammi, F. Casella, M. Ricotti, F. Schiavo, and G. Storrick, “Object-oriented simulation for the control of the iris nuclear power plant,” in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005, pp. We–A02–TP/25.

[9] K. Krueger, R. Franke, and M. Rode, “Optimization of boiler start-up using a nonlinear boiler model and hard constraints,” *Energy*, vol. 29, pp. 2239–2251, 2004.