A Modelica Library for Scalable Modelling of Aircraft Environmental Control Systems

Philip Jordan Gerhard Schmitz
Hamburg University of Technology, Institute of Thermo-Fluid Dynamics
Denickestr. 17, 21075 Hamburg

Abstract

In the design process of complex technical systems such as environmental control and cooling systems for modern commercial aircraft, modelling and simulation is widely used. Simulation can provide viable insight during all phases of the system design cycle, but the questions to be answered by a simulation activity vary along the advancement of the design process. In this paper, the Modelica library developed in the Clean Sky project TEMPO is presented. The aim of the library is to support scalable system models which can be modified in detail and characteristic to be used during different phases of the design cycle without the need of rebuilding the system model or switching to another software tool. The library structure and the approach to integrate different detail levels is outlined and demonstrated at the example of a generic aircraft environmental control system architecture.

Keywords: Environmental control system, scalable detail, object-oriented modelling, library

1 Introduction

The competitive environment and the challenge to limit environmental impact while air traffic demand is increasing, aircraft manufacturers and suppliers are under continuous pressure to improve the performance of their products. Since it becomes increasingly difficult to improve the global system performance by independent optimization of subsystems, the need to instead aim for an optimized global system performance during development has been identified [1]. New and unconventional aircraft systems architectures such as the "More Electric Aircraft" (MEA) are the subject of past and ongoing European research programmes [2, 3].

The design process of complex technical systems such as aircraft environmental control and cooling systems can be broken down into characteristic design phases. During all phases, modelling and simulation can be used to support the design process. However, each phase may pose different questions with regard to the physical phenomena of interest, the accuracy, or the simulation speed for example. Specialized tools can be used to meet the requirements for each simulation task. The library presented in this paper aims at using the object-oriented features of the Modelica language to integrate multiple layers of models. Each layer corresponds to a set of models designed for a specific simulation task and is subsequently referred to as "Detail Level". The Detail Levels are integrated such that a single system model that has been assembled from available library component models can be "scaled", i.e. reused, for a simulation activity of a different phase of the system design cycle.

The ability to create polymorphic models is a core feature of the Modelica language and it is used to create flexible and configurable models in many published Modelica libraries. An example are the variable dynamics settings and exchangeable heat transfer models in the `DynamicPipe` Model of the Modelica Standard Library. Several libraries which pick up the concept and focus the design of the library more on scalable models have been presented.

In [4] structuring a Modelica library with scalable models for building simulation with the need for different models along the temporal evolution of a project in mind is discussed. A library for power plant simulation currently under development uses the idea of detail levels as group of models with similar complexity and incorporates the concept to organize the models within the library structure [5].

A somewhat different approach towards the concept of a "Detail Level" is taken in the library subject to this paper. Rather than a specific parametrization of a polymorphic model, a "Detail Level" represents the required characteristics of a model designed for a specific simulation task. This task is to be carried out at a certain point or in the system development process.

DOI
10.3384/ECP14096599

Switching a system model between two “Detail Levels” can therefore be seen as switching between different use cases where the information about the existing system structure is kept. The advantage of this is that the user does not have to rebuild the same system with another set of specialized models even though the actual model equations for each detail level may not have much in common.

The library discussed in this paper is developed within the scope of the TEMPO project (Thermal Exchange Modelling and Power Optimization). TEMPO is a research project in the Systems for Green Operations (SGO) technology demonstrator of the Clean Sky Joint Technology Initiative. The focus of the SGO demonstrator is to mature technologies which were developed during the MOET (More Open Electrical Technologies) project [2], and demonstrate architectural integration of thermal management technologies such as the electrically driven air system which was developed in MOET. In order to optimize and validate such architectures, extensive modelling studies are carried out. The presented library supplies the models for the environmental control system for the SGO demonstrator.

2 Models for Different Stages in the System Design Process

The process of designing and implementing a technical system is characterized by distinctive phases. In each phase design decisions have to be made and predefined milestones have to be reached before the development can proceed to the next phase [6]. The V-Diagram shown in figure 1 is a graphical representation of the system development process.

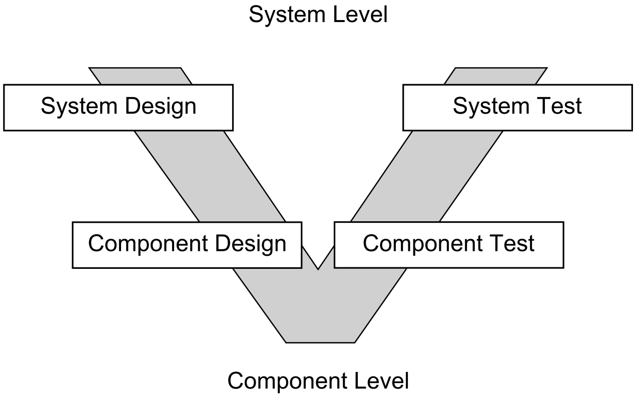


Figure 1: System design process displayed in the form of a V-Diagram

In this case, the process has been broken down into four characteristic phases, System Design, Component Design, Component Test and System Test. Based on the tasks associated to the design phase, requirements for the models in the respective phase have to be formulated. The following section describes the characteristics of the Detail Levels of the TEMPO library. Table 1 gives a brief overview over characteristic differences of the main components for different Detail Level settings.

2.1 Phase 1: System Design

The system development process begins with the conceptual design of the system followed by preliminary system design. Based on customer need, system requirements have to be developed. From the overall system-level requirements, design requirements for the subsystem level are deducted. The phase covers feasibility assessments and leads to the decision on a system architecture. Simulation during this phase is used to support decisions on a system architecture and to deduct performance requirements for the components of the system.

The models designated for use during this phase are collected in the Detail Level 1 of the library. The Detail Level 1 models do not include dynamic effects. Performance characteristics are modelled by parameter setting. Turbomachinery components for example allow user defined efficiency settings, heat exchanger models are effectiveness based where the effectiveness can be a user defined parameter. The simple modelling assumptions aim for very fast and robust models for initial requirements estimation. The effects of a parameter variation on the system behaviour in this setting is limited by the constraints imposed on the remaining components.

2.2 Phases 2 and 3: Component Design and Component Testing

In the detailed design and development phase of the system development cycle, subcomponents and subsystems are brought into realization. As the design of the component progresses, more reliable performance predictions become available. These component performance predictions can be fed back into the system model, for example to check for compliance with overall system performance requirements. In parallel to the development of the actual components and their thermodynamic performance validation, the development of a control strategy is started. Models for control

DOI 10.3384/ECP14096599

Table 1: Overview of main component model characteristics for different Detail Level settings

Component	Detail Level 1	Detail Level 2	Detail Level 3	Detail Level 4
Turbine	Fixed operating point (parameter)	Map based, input filtering	Map based, input filtering	Simplified flow model, forward computation
Compressor	Fixed operating point (paramter)	Map based	Map based	Map based
Heat Exchangers	Static balance eqns, fixed effectiveness, optional local over/under determination	Static balance eqns, map based heat transfer, optional local over/under determination	Dynamic balance eqns, dynamic wall, geometric parametrization, discretization	Surrogate function for outlet temperatures calculation
Valves	Optional local under determination	Optional local under determination, input filtering, flow reversal supported	Input filtering	Input filtering, forward computation
Reservoir (tank) models	Pressure boundary, optional local over/under determination	Map based, optional local over/under determination	Map based, dynamic balances equations, global mass control	Map based, dynamic balances equations, global mass control

developments focus on the dynamic behaviour of the components.

When component prototypes have been produced or acquired, they are tested and additional performance data becomes available. In the form of performance maps, this information can be fed back into the system model, both for thermodynamic performance models and models for control development.

The Detail Level 2 Models of the library are designed to be used for thermodynamic performance simulations during these phases of the development process. The components are static models, performance characteristics are typically calculated from performance maps. Turbomachinery components use efficiency and flow characteristic maps, heat exchangers use effectiveness maps.

In the library, the models for control development activities during these phases are collected in Detail Level 3. These models use a more physical modelling approach. The heat exchanger models now include dynamic mass and energy balances on the fluid side and dynamic energy balances for the core material. Heat exchangers with connection to the incompressible liquid fluid domain also have the option to activate a dynamic momentum balance. The balance equations are discretized into finite volume elements. To somewhat limit computational effort and to improve robustness, the heat exchangers use lumped momentum balances and flow reversal is not supported. The parametrization of the heat exchangers is geometry based and can be modified to change surface geometries and flow patterns.

2.3 Phase 4: System Test

After the components and sub-systems have been designed and the prototypes have been manufactured and tested, they are integrated into the final system prototype. Simulation activities may include analysis of component behaviour with integration effects taken into account or hardware in the loop simulations to validate the developed control system.

Models for this kind of activity make up the Detail Level 4 in the library. The models used include dynamic behaviour, but the models are simplified in favour of faster simulation speed. The models are designed towards the use of fixed step solvers, while actual hardware-in-the-loop capabilities have not been implemented within the scope of TEMPO. A heat exchanger model of this Detail Level calculates outlet temperatures based on transfer functions. Fast states such as pressures in air components have to be removed to allow the usage of a sufficiently large step size with a fixed step solver.

3 Integration of Detail Levels

Two aspects regarding model structure for scalable models are discussed in this paper. This section treats how the detail level specific models are combined to achieve the model switching capability on the system level. Section 4 contains a description of the organisation of the models within the library structure.

The different available Detail Levels are collected on the component level. Components in this context are units such as heat exchangers, valves or compressors. To create a system model, scalable components from the library are used to build up the desired architecture. As a sum of its components, the system model is also scalable. Scaling the system model is done by switching all components to a different Detail Level using a global parameter, while the system architecture information is kept.

In order to change the Detail Level setting of a system model, the `inner/outer` construct is used. The system model contains an `inner`-object of the class `SystemState`, in which the Detail Level setting is stored. In the library, the default instance name of the `SystemState` instance is “system”. All component models inherit from a `PartialComponent`-class which provides the interface to the `inner`-object “system” on the system level. In this class, the user can access the Detail Level setting from the graphical user interface.

3.1 Component Level Integration of Detail Levels

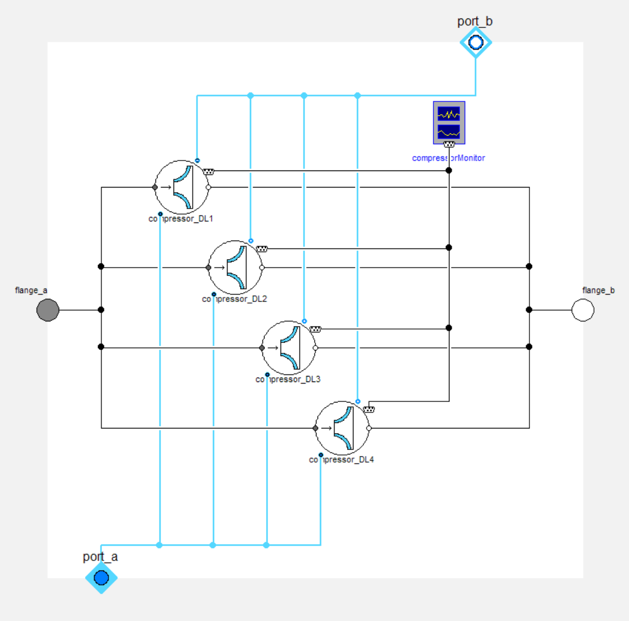


Figure 2: Container model structure seen as in the diagram layer view of a scalable component model

Polymorphism in Modelica is usually achieved by using the `redeclare` and `replaceable` language elements. Variations of encapsulated elements of the model can be exchanged in order to create the desired variant of the component. However, the Modelica language does not support parameter-based redeclaration. As a consequence, it is not possible to invoke a series of redeclarations by one central parameter setting. Therefore `redeclare` and `replaceable` constructs can not be used to achieve the Detail Level switching behaviour needed for this application.

Instead the container method of class parametrization is applied to build scalable components from the Detail Level specific models. The container approach is a very basic parametrization method, where the scalable component model contains a set of conditionally declared variants of the component. Figure 2 shows the diagram layer of a scalable compressor model. Depending on the Detail Level parameter setting, only the desired model is instantiated. An advantage of this approach is the complete encapsulation of the Detail Level specific code. Model equations and class structure (internal class hierarchy levels for example) of a component may vary greatly between Detail Levels. The encapsulation allows task-oriented modelling independent of requirements for other Detail Levels, as long as the interfaces are compatible.

By default, the components are configured to follow the global Detail Level setting. This global Detail Level setting can be manually overridden individually for each component.

3.2 System Level Integration of Detail Levels

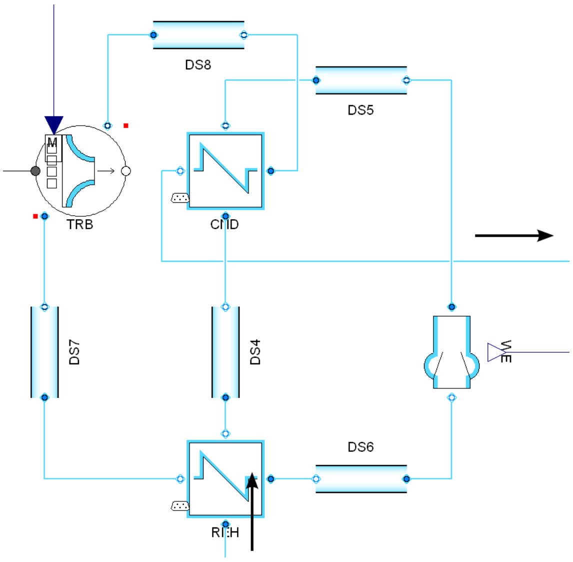


Figure 3: High pressure water extraction loop model with Detail Level dependent decoupling modules

A system model expressed in the equations of different Detail Levels can not always naturally be decomposed in the same manner. By building up a scalable system model from of scalable component models, at least one layer of decomposition is fixed and the library has to address this issue. A common example of such a decomposition issue is a closed loop fluid system like the vapour cycle included in the application example in section 5. With dynamic mass and energy balances on the fluid side, the system is well-posed if it is made by connection of balanced component models according to the Modelica specification. A purely static variant of the same system on the other hand can not be created the same way. The connect-statement closing the loop introduces a redundant mass balance while the overall system lacks a “grounding”-equation.

A well-posed static model made from locally balanced components could be achieved by removing the connect statement and manually adding a pressure or total mass constraint. To be able to scale a system model without the need to modify the system topology, the option to add or remove equations has been shifted to the component models. The consequence is that individual components can now be configured to be locally over- or under-determined. With reference to the application example, this means that for the Detail Level 1 version of the vapour cycle, that the valve is configured to act as a loop breaker while the refrigerant reservoir is locally overdetermined and provides the necessary grounding. Local over- and under-determination is also used to adapt the causality of the overall system if required by the simulation task while the system level topology remains unmodified. In the presented vapour cycle system, the evaporator superheat is controlled by the expansion valve opening position. Be it that the control logic is not available and not to be simulated in the conceptual design phase (Detail Level 1). Instead, the evaporator superheat can be fixed. The additionally introduced equation in the evaporator model is compensated by configuring the valve coefficient as an unknown. This requires extra care by the user, since a globally balanced system is no longer guaranteed.

Another aspect which has to be considered for integrating the Detail Level specific models into a scalable system model are the constraints imposed by the system architecture information. When the Detail Level specific equations are incorporated into a scalable component model, the architecture of the system in which they will be used in is yet unknown. But the overall equation system which will have to be solved includes the constraints associated with the system architecture information. Unfavourable combinations of model equations and connections can lead to high index systems or non-linear algebraic loops. Such unfavourable equation system structure cannot always be avoided, but the modeller will generally try to adapt the system model and the component configuration for improved robustness and performance. In thermo-fluid systems for example, complex networks of static components usually lead to large non-linear systems of equations. If the time scales of interest allow it, artificial dynamic states can be introduced to decouple the static components. In a scalable system model, it may occur that such measures are necessary for one Detail Level, but not for another. In a scalable system model, as it is presented here, multiple system models in fact “share” one set of system level constraints, i.e. the system architecture information. This means, that measures of which the necessity depends on Detail Level and system level constraints, need to be integrated into the Detail Level management. In the presented library, dynamic fluid states for decoupling large non-linear systems of equations are implemented in dedicated components.

The image in figure 3 shows the Modelica Diagram view of a model of the high pressure water extraction loop in air generation unit of an environmental control system (ECS). Such a system is installed on the Boeing 747-8 for example [7]. In a purely static configuration, the component equations together with the circular connections lead to a difficult to solve algebraic loop. The decoupling modules located between the components include the artificial state variables, but depending on the Detail Level setting, they are only activated when the Detail Level setting requires it.

4 Library Structure

The library structure is organized around component models such as valves or heat exchangers, which are ready to be connected in an executable system model after they have been parametrized. A general modelling philosophy is the aim for a flat inheritance structure for improved readability at the cost of possibly duplicate code elements. Where possible, the components contain no more than two inheritance levels. A common base class provides the component icons and common GUI elements like start value parameters.

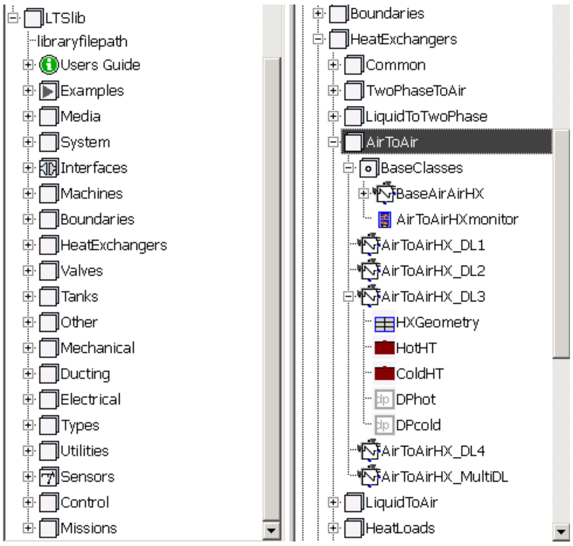
The library covers three fluid domains: Air-, refrigerant- and incompressible liquid coolant domain. Thermofluidic components are by design permanently associated to a fluid domain. For example, there is a dedicated valve model for the air domain and a dedicated valve model for the incompressible liquid domain. The separation allows simpler—because more specialized—code for the components. The association of a component to a fluid domain is indicated by colour marking of the icon and connectors. The colour of air domain components is blue, refrigerant domain components are identified by green coloured icons and connections and the incompressible liquid domain components are of orange colour.

4.1 Fluid Modelling

Within TEMPO, proprietary fluid property models provided by the SGO project partner are used. The data is accessed using external functions. The Modelica interface to the fluid data follows the `Modelica.Media` model structure for easy conversion to MSL fluid models. Each of the available fluid domain—air, refrigerant and incompressible liquid—have their own base package, fitted to the characteristics of the fluid type. The fluid connectors are identical to the `Modelica.Fluid`-connectors of the Standard Library. Currently, no distinction is made between total and static values for pressure and specific enthalpy.

Variants of the air type fluid model cover a range of dry and moist air models from simple and fast to more complex and accurate. Variants of the refrigerant and incompressible liquids cover several different fluids.

4.2 Component Organization



LTSlib
libraryfilepath
Users Guide
Examples
Media
System
Interfaces
Machines
Boundaries
HeatExchangers
Valves
Tanks
Other
Mechanical
Ducting
Electrical
Types
Utilities
Sensors
Control
Missions

Boundaries
HeatExchangers
Common
TwoPhaseToAir
LiquidToTwoPhase
AirToAir
BaseClasses
BaseAirAirHX
AirToAirHXmonitor
AirToAirHX_DL1
AirToAirHX_DL2
AirToAirHX_DL3
HXGeometry
HotHT
ColdHT
DPhot
DPcold
AirToAirHX_DL4
AirToAirHX_MultiDL
LiquidToAir
HeatLoads

Figure 4: Package tree of the TEMPO library

Two main criteria are used to group the library elements in an efficient manner. The top level criterion is a generic functional classification of the type of component as shown on the left side of figure 4. Within each functional group, the components are classified according to the fluid domain they are associated with. On the right side of figure 4 it is illustrated for the `HeatExchangers`-package. All variants of the Air-to-air heat exchanger model extend from a common base class. The scalable component model `AirToAirHX_MultiDL` is the container model integrating all Detail Level specific models. The finite volume based dynamic heat exchanger model of Detail Level 3 is assembled from a set of subcomponents. These are located in the `Common` package.

5 Application to a System Model

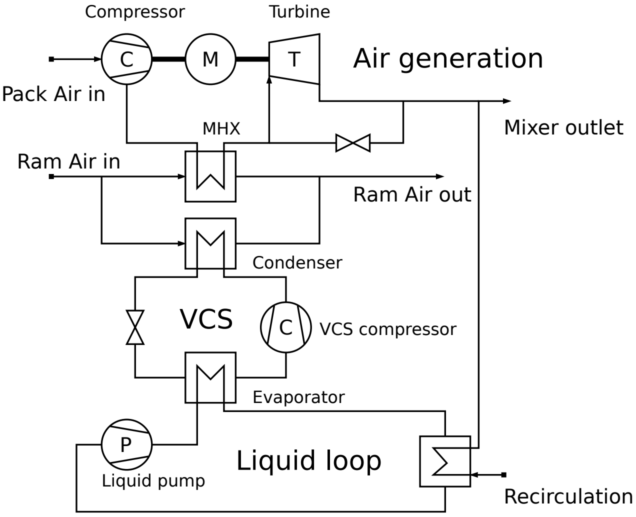


Figure 5: Reference architecture: Generic bootstrap cycle with motorized compressor, liquid cooling loop for cabin air recirculation cooling and vapour cycle

The application of the library is demonstrated on a generic ECS architecture. The architecture is modelled with the scalable components and simulated in different Detail Level settings. Fully real-time capable models which could be used in a real time environment are not yet supported by the library and are beyond the scope of TEMPO, therefore the simulated Detail Level settings cover the design phases 1–3 described in section 2.

The data used in the models such as performance maps and geometry information is proprietary, so that some the plotted results have been normalized.

5.1 Reference Architecture Description

The modelled ECS architecture is shown in figure 5. The system includes the air generation unit (pack) which provides pressurized air to the cabin. The presented system is based on a generic bootstrap cycle. Ambient air is pressurized in the compressor, excess heat is discharged in the main heat exchanger (MHX) using ram air as heat sink. The pack air is then cooled down

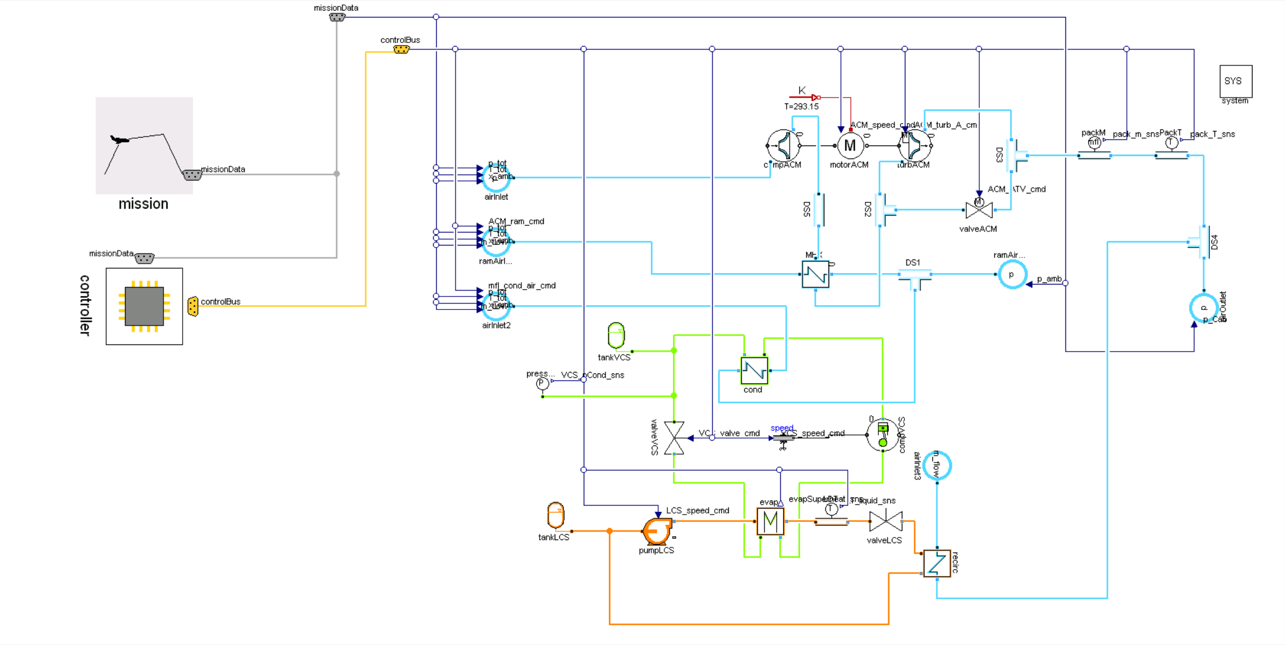


Figure 6: Diagram layer view of the ECS model

further in a turbine before it is released into the air distribution system. A liquid cooling loop is used to cool recirculation air from the cabin. A vapour cycle (VCS) is used to discharge the thermal load via ram air.

The diagram view of the reference architecture model is shown in figure 6. Boundary conditions and system control signals are provided by additional models. Mission (ambient conditions) and controller models are not integrated into the Detail Level management, i.e. the container model parametrization is not applied on them. The use of ambient data is independent from the Detail Level setting, therefore the data can be used in all Detail Levels. The controller model is assembled individually for a system as different controller models may be used on one ECS model.

In the model used for the simulations for the application example, the aircraft cabin is represented by an altitude dependent pressure boundary. The ram air mass flow control is simplified by directly imposing the mass flow rates and the cooling load from the recirculation air remains constant over time.

5.2 Detail Level 1 Use Case

All components of the system model are modelled statically. The model is intended for easy parametrization effects on the system and fast computation speed. This is reflected in simplified modelling assumptions and allowing to set properties as parameters that normally depend on the operating point. These properties are heat exchanger effectivenesses, superheating or subcooling temperature differences for vapour cycle heat exchangers, pressure ratios and efficiencies for turbomachinery or vapour cycle pressure levels for example.

For a simulation with the ECS model in Detail Level 1 configuration the ambient conditions for the model were set constant to represent a cruise flight at 38 000 ft with Mach number $Ma = 0.8$ at ISA conditions.

With the local over- and under-determination of the components, the degrees of freedom of the model can be changed. In the simulated example, the effectiveness as well as the superheat temperature difference of the VCS evaporator are fixed, so that the heat flow rate is determined twice. In return, the valve constant is set to be a free variable.

In figure 7 the evolution of the required condensation pressure and the power required is plotted for different settings of the condenser effectiveness. The pressure level has been normalized with a reference maximum pressure, the power demand has been normalized with a reference power.

5.3 Detail Level 2 Use Case

The Detail Level 2 configuration provides static models with more complex and accurate modelling assumptions. The components are mainly based on performance maps. Data available from detailed component design and prototype testing (Phases 2 and 3) can be

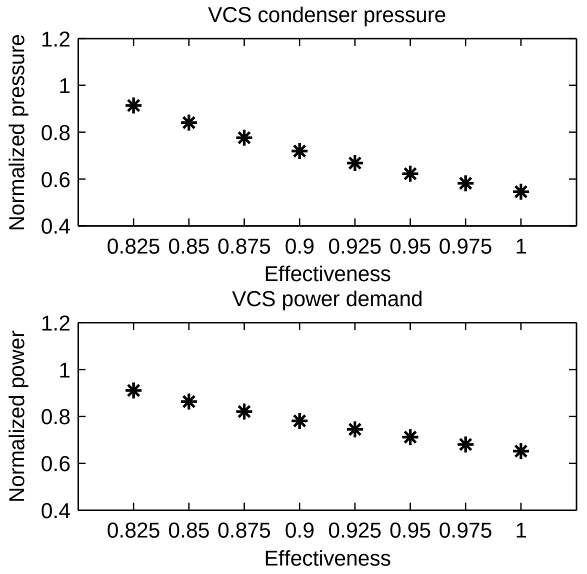


Figure 7: Evolution of required VCS high pressure level and VCS power demand for a given condenser effectiveness

used to verify the system performance. The additional variability of component performance properties allows to evaluate the effects of component interaction on the system level.

The ambient conditions for the Detail Level 2 simulation are unchanged and correspond to level flight at 38 000 ft, $Ma = 0.8$. In this example, the vapour cycle superheat temperature is again set as a parameter. The ram air mass flow rate is reduced during the simulation. In figure 8 the output of the simulation is shown. From top to bottom, the curves show the evolution of the mass flow rates in the condenser, the normalized pressure levels of the VCS, evaporator liquid side outlet temperature deviation from target and the VCS power demand. The mass flow rates are normalized with respect to a reference mass flow rate, the pressure levels are shown in reference to a maximum allowed pressure. The result of the reduction of the ram air mass flow rate is an increase of the pressure levels. The change in evaporator pressure has an impact on the driving temperature difference and the evaporator heat flow rate decreases. As a result, the evaporator liquid outlet temperature increases. The result is presented in form of a deviation of a target temperature. In this case, the reduced ram air flow still provides enough cooling for the liquid cycle. However, the increased VCS power demand counteracts the reduced ram air flow.

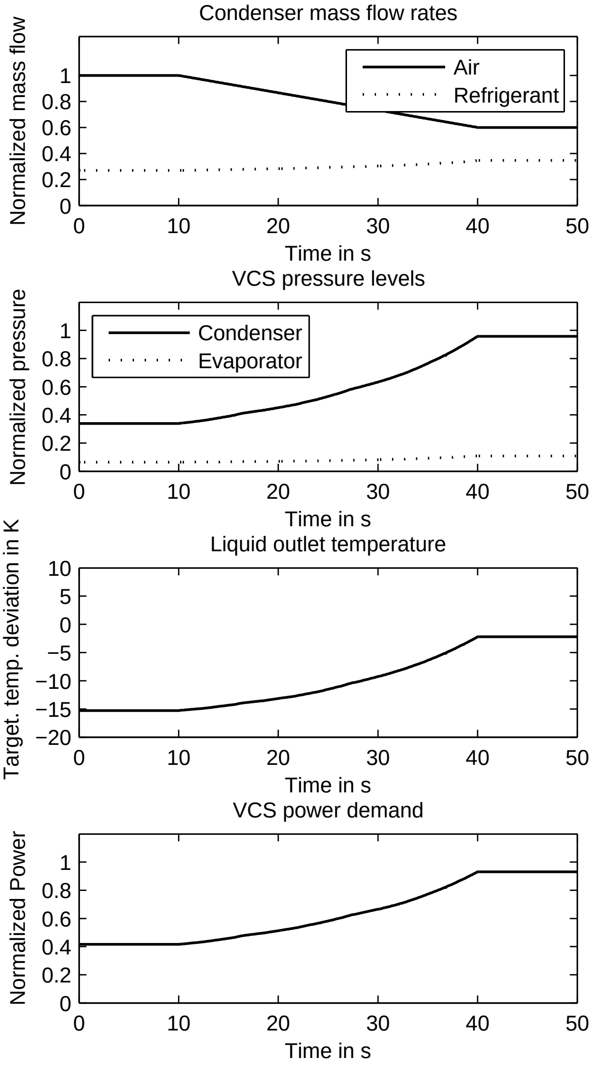


Figure 8: Results of simulation in Detail Level 2 setting

5.4 Detail Level 3 Use Case

The system model in Detail Level 3 setting switches the underlying equations to a combination of dynamic and static components. Valve, pump, turbine and compressor models are static and use performance maps similar to the Detail Level 2 setting. System dynamics are captured by the heat exchanger models. The liquid cycle components dynamic momentum balance option have been switched off.

The system model is now used to test a control strategy. In the Detail Level 3 example, pack outlet temperature, pack mass flow rate and evaporator liquid side outlet temperature are controlled to maintain target values. The controller model used together with the

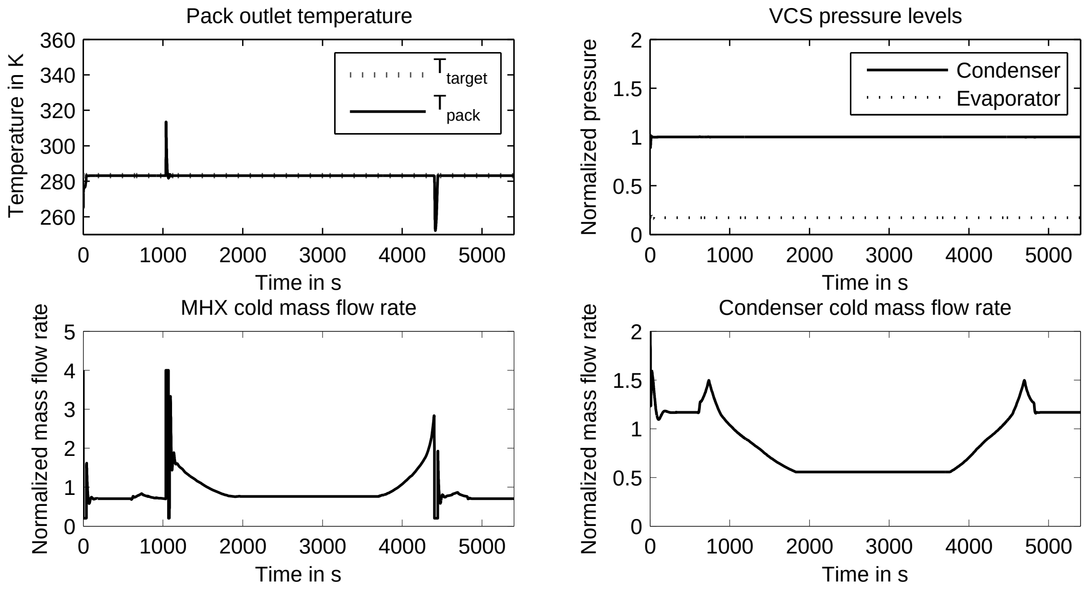


Figure 9: Dynamic mission simulation results

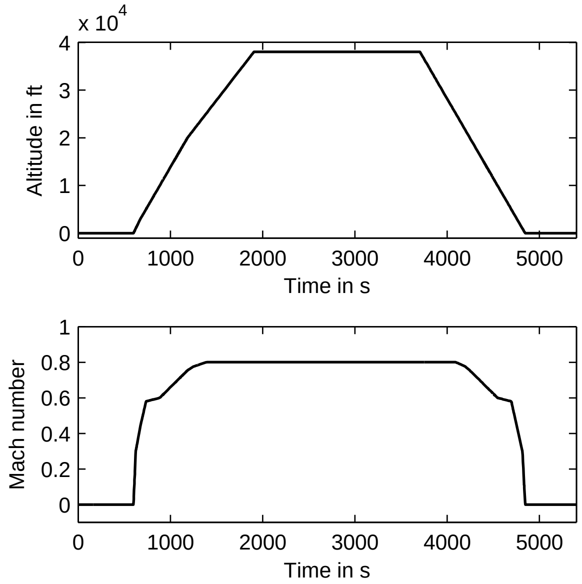


Figure 10: Mission altitude and Mach number profile for the dynamic Detail Level 3 use case

ECS model is assembled from has been implemented with elements from the Modelica Standard Library and the `Modelica_StateGraph2` library.

Figure 10 shows the mission profile which was used for the simulation. After an initial taxi-out period, a climb phase follows until a cruise altitude of 38 000 ft is reached. The descent begins after 30 min of cruise followed by a taxi-in phase on ground. The Mach profile is a function of the altitude, cruise speed is $Ma = 0.8$.

The top left graph in figure 9 displays the achieved pack outlet temperature, which matches the target output temperature except for two occasions. The bottom left graph shows the ram air mass flow rate in the main heat exchanger, which is modulated to control the pack outlet temperature. The mass flow rate shown is normalized by a reference mass flow rate. The controller tested in this simulation produces some overshoot and oscillation and should probably be revised and re-tested.

The control of the vapour cycle pressure levels however is successful in this simulation as it can be observed in the top right graph. The plotted data is normalized by the target condenser pressure. The bottom graph shows the condenser ram air flow which is used to maintain the condenser pressure. With increasing altitude, less ram air is needed due to the colder temperatures.

6 Conclusion and Future Work

In this paper, the Modelica library developed within the Clean Sky project TEMPO is presented. The library is designed for aircraft environmental control and cooling systems modelling. The focus of the library is the thermo-fluid domain, covering air, vapour cycle and liquid cooling systems. A key feature of the library is the integration of multiple levels of detail, where each Detail Level corresponds to a set of models specifically fitted to a certain simulation task. The detail levels are defined according to the simulation activities which are carried out along the process of developing an environmental control system from the initial conceptual phase to the finished prototype. The library is organized around the Detail Level concept and allows to switch a system model from one Detail Level to another. The underlying model equations are thus exchanged to meet the requirements of another simulation task while preserving the information on the system architecture, freeing the developer of the need to entirely recreate the system model from scratch. As an application example, a scalable model of a basic aircraft environmental control system has been assembled and simulated for three different Detail Level settings. The development of the TEMPO library is ongoing work. A cabin model for commercial aircraft is currently under development. For the near future it is planned to make the library fully compatible to the recently extended `Media`-Library of the Modelica Standard library and to separate other protected intellectual property of the project partner from the core library. This version of the library will then be made available to the public.

7 Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) for the Clean Sky Joint Technology Initiative under grant agreement No. 270561.

References

[1] M. J. Provost, "The more electric aero-engine: A general overview from an engine manufacturer," in *Proceedings of the International Conference on Power Electronics, Machines and Drives*, Institution of Engineering and Technology (IET), 2002.

[2] T. Jomier, "More open electrical technologies (MOET-0.02-AF-DEL-PublicReport-0001-09-R1.0)," technical report, MOET Consortium Partners, 2009. Available online at `http://www.eurtd.com/moet/PDF/MOET_Public_Technical_report.pdf`.

[3] Clean Sky JTI, "Systems for green operations technology demonstrator overview." `http://www.cleansky.eu/content/project/system-green-operations`, accessed October 2013.

[4] M. Bonvini and A. Leva, "Scalable-detail modular models for simulation studies on energy efficiency," in *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical University, Dresden, Germany*, 2011.

[5] J. Brunnemann, F. Gottelt, K. Wellner, A. Renz, A. Thüring, V. Roeder, C. Hasenbein, C. Schulze, G. Schmitz, and J. Eiden, "Status of ClaRaCCS: Modelling and simulation of coal-fired power plants with CO_2 capture," in *Proceedings of the 9th International Modelica Conference, September 3rd-5th, 2012, Munich, Germany*, 2012.

[6] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*. Upper Saddle River, NJ: Prentice-Hall, 3 ed., 1998.

[7] Boeing Corporation. `http://www.boeing.com/commercial/aeromagazine/articles/2012_q1/4/`, accessed in November 2013.