ELSEVIER

Advances in Engineering Software 36 (2005) 127–134

ADVANCES IN ENGINEERING SOFTWARE

www.elsevier.com/locate/advengsoft

Computational modular model library of gas turbine

Yuan Cao[a,*], Xianlong Jin[a], Guang Meng[a], Jay Fletcher[b]

[a] *School of Mechanical and Power Engineering, Shanghai Jiaotong University, Shanghai 200030, China*
[b] *United Technologies Research Center, East Hartford, CT-06108, USA*

Abstract

Designing and developing new aircraft engine is time-consuming and expensive. Computational simulation is a promising means for alleviating this cost, but constructing flexible simulation software capable of evaluation of integrated aircraft engine system architectures is hard work. This paper addresses the design of a tool—a generic modular-modeled library of air gas turbine, this library is based on object-oriented technology and hierarchical decomposition and provides a flexible component-based representation for defining, modifying, and simulating the aircraft gas turbine system, subsystem and components. It enables users to customize and extend the framework to add new functionality or adapt the simulation behavior as required, and it allows new models to be composed programmatically or graphically to form more complex models. The model library can be used in steady-state and transient analysis of the aero-engine. It is also a user-friendly, accurate, fast PCbased and easily reusable simulating tool. The advanced object-oriented simulation language Modelica is used to construct this library. Modelica provides a powerful tool to design the library. All of the work described in this paper is developed based upon Dymola/Modelica.

Keywords: Gas turbine; Object-oriented; System-level simulation; Dymola/Modelica

1. Introduction

As the aircraft industry enters 21st century, there is increasing pressure to reduce the time, cost and risk of aircraft engine development and maintenance. To compete effectively in global marketplace, innovative approaches to reducing aircraft gas turbine engine design-circle span, manufacture and maintenance cost are needed. An opportunity emerged to realize this point with computational simulation. As modeling techniques have improved, and computers have progressed, simulation has assumed an essential role in planning, executing, and evaluating operations. So too, in the design, manufacturing, and operating of aircraft turbine engines, accurate performance simulations have become essential. Therefore, throughout the years, many computer-based models for the simulation of the operational characteristics of the aircraft gas turbine engine have been structured and evolved into a very wide range of applications [1–4].

To make sure the requirement of commercial users of engine models, a detailed survey was prepared, and distributed to users and creators of engine models by Applied Vehicle Technology Pane (AVT) in Research Technology Organization (RTO) of NATO. The responses varied in details, forms and sources, but were adequate to lead to the following conclusions: 'There is a trend towards object-oriented and graphical approaches; the general trend is to move towards workstations and PC systems which offer graphical user interfaces; flexible and modular systems are required. Also, there is a requirement to model components and systems at varying levels of detail' [3].

Based upon these, the objective of our work is to develop a set of extensible generic model libraries of components and subsystem of the aero engine system that may be used to rapidly assemble a system level dynamic model and to evaluate the dynamic performance of integrated power system architectures during conceptual and preliminary design. As such, the intended use is not for particular component design but rather to understand the interaction of components.

Due to the wide range of interactions between different engineering fields, especially in aero-engine, systems are getting more complex and heterogeneous. For example, the gas turbine engine system includes several engineering fields, mechanics, hydraulics, electrics, thermodynamics

---

* Corresponding author. Tel.: +86-21-629-32256.
E-mail address: caoyuan@sjtu.edu.cn (Y. Cao).

0965-9978/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2004.07.006

and so on. When trying to create models of these systems, problems arise with the interaction of the different parts. Fortunately, among the recent research in modeling and simulation, there are two concepts that are closely related to these problems: object-oriented modeling language and non-causal modeling. During the last few years, a new advanced multidisciplinary object-oriented modeling language Modelica has been developed. It is designed to allow convenient, component-oriented modeling of complex physical systems, e.g. systems containing mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents [5]. So in this paper the model libraries will be developed based no the Modelica language.

2. Basic design IDEA

2.1. Library design

Following the top–down design philosophy and considering the different engineering fields related to aero-engine system structuring, the whole system can be divided into several sub-systems; each sub-system consists of various components. They are shown in Fig. 1.

Based on the above decomposition, each subsystem has its respective library of component models. Although the scope of this work is the development of the libraries of several subsystems that can be assembled to capture the physical characteristics of the whole aero-engine system, our emphasis in this paper is to construct the aircraft gas turbine system model library. So the library of gas turbine system will be particularly illustrated in details in the next section.

To build a complete aircraft gas turbine system-level model, the strategy is determined and shown in Fig. 2. Both the prospective model library and the system-level model will be designed according to the strategy.

2.2. Object-oriented modeling and hierarchical structure

Traditionally, simulation software development has followed the top–down structured design approach, which applies the method of functional decomposition to establish program structure. Although this method has been successful in some applications, it fails to reflect the real world. As a result many attempts have been made to tackle this problem by applying Object-oriented technology (OO technology).

So the whole aircraft gas turbine system is designed as an object-oriented framework, which is a set of classes that embodies an abstract design for solution to a family of related problems (Johnson and Foote, 1988) [6]. The set of classes define 'semi-complete' applications that capture domain-specific object structures and functionality. Specific functionality in new applications is realized by inheriting from, or composing with, framework components.

In this paper, new object-oriented simulation software Dymola/Modelica is used to develop the gas turbine model. It is also structured to enable the assembly of object-oriented hierarchical libraries, which can be later used to assemble system-level dynamic models. And several other Modelica libraries, such as mechanical, electrical and thermoflow library, can be obtained freely.

The layout configuration of the aircraft gas turbine is shown in Fig. 3. From a structural view point, a gas turbine can be essentially as assembly of engine component—inlet, fan, compressors, combustor, duct turbine, shafts and nozzle. These components can be represented in the computational domain as objects [6].

During the simulation, the engine's mathematical model is mapped to collections of interacting objects, rather than decomposed into segments of different functions that can implement certain algorithms. Each object mimics the behavioral and structural characteristic of a physical or conceptual entity shown in Fig. 3. And it represents an instance of a software class, while the classes are united into

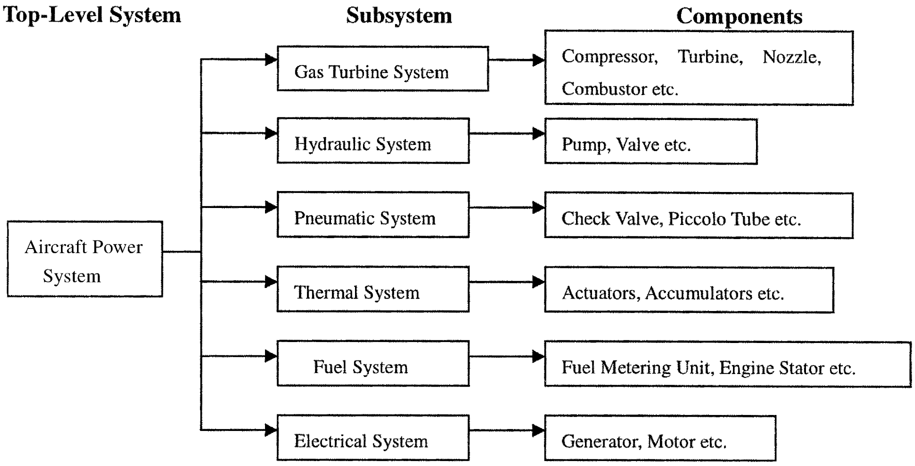

Fig. 1. Sub-systems and components of aero engine system.

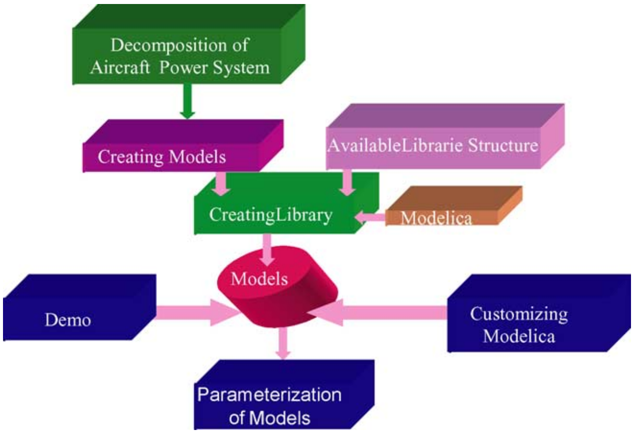

Fig. 2. Flow chart of aero-engine model design.

a hierarchy via inheritance relationships [6]. For example, the class (or partial model) 'BaseCompressor' is used to generate different kinds of compressors. However, it inherits from a more generalized class 'CompressorMap'. If a new compressor object need to be modeled, it can easily be instantiated from the existing class 'BaseCompressor'. In this manner, a well-planned class hierarchy provides the 'slots' into which the future codes are to be plugged.

2.3. Non-linear model

In the design of the mathematical model of the aircraft gas turbine, the following factors are considered:

The inertia of rotor and mechanical efficiency;
Mass and energy storage and release (mass, energy, and momentum balance);
Air of cooling system, or leaking;
Variation of specified heat capacity and adiabatic exponent;
Pressure loss in combustor, and combustion efficiency etc are considered by setting relevant variables on customized windows of the models.

Non-linear mathematical model of the gas turbine has been discussed in many books and papers. In this paper, the model is divided into several modular models with the object-oriented technology. Each modular model represents a component of the gas turbine, and can be used to model the corresponding component's function. For example, the variations of specific heat capacity and adiabatic exponent can be acquired in the work fluid modular (medium) model.

Based on the modular model the hierarchical structure is easily structured, in addition, 'Message-based modeling' [1] is used for the interaction among models.

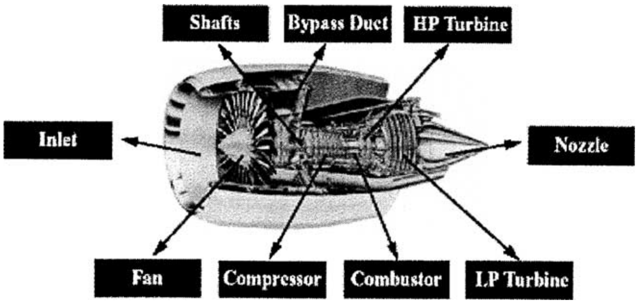

Fig. 3. Aeroengine structure [7].

3. Common engineering model

In an aircraft gas-turbine library, physical engine component structures and substructures are represented by the different combinations of the volume model, flow model, medium model, and connector model objects. And all these objects are designed based on the OO technology. The object composition provides a powerful mechanism for representing the physical topology of the gas turbine engine.

3.1. Volume model and flow model

The difficulty in modular is keeping the independence of every modular model, at the same time, considering the coupling among different models. So the concept of volume and flow is developed. In recent years, control volume models have been used to model the thermodynamics systems that involve mass flow and energy in and out. Besides the volume model, the flow model is also indispensable in modeling the gas turbine components [8]. So another model named flow model is introduced for calculating the mass flow and the convective energy associated to the mass and energy flow. Flow models are the result of modeling abstraction, where the volume is neglected. The flow models contain either an algebraic equation that relates pressure drop and mass flow, or a mathematical expression for dynamic momentum balance. The storage of mass and energy are modeled in the control volume model. Volume models and flow models always have to alternate each other [8] as Fig. 4 shows.

Volume model and flow model are connected by flow connectors (see Section 3.3). In the volume model the output variable quantities are pressure, specific enthalpy, density, temperature, specific entropy and ratio of specific heats, respectively. And the input variable quantities are mass flow and convective heat. While the flow model's output variables are mass flow and convective heat flow, the input information is pressure, specific enthalpy and so on.

Suppose the gas (or air) in the volume model is uniform. Any heat transfer and momentum are neglected. The energy and mass balances of a volume can be expressed in the follow balance equations (take the enthalpy and pressure as the state variables):

$$m_{2i} - m_{1i} + V_i\left(\frac{\partial \rho_i}{\partial p_i}\frac{dp_i}{dt_i} + \frac{\partial \rho_i}{\partial h_i}\frac{dh_i}{dt_i}\right) + \rho_i\frac{\partial V_i}{\partial t_i} = 0 \qquad (1)$$

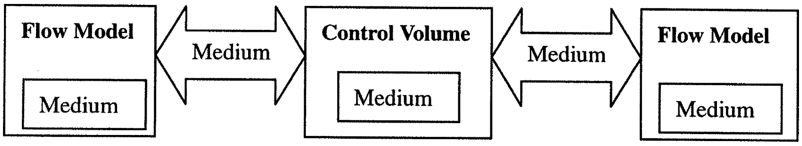

Fig. 4. Volume and flow model.

$$m_{2i}h_{2i} - m_1 h_{1i} + V_i\left(\rho_i + h_i\frac{\partial\rho_i}{\partial h_i}\right) + V_i\left(h_i\frac{\partial\rho_i}{\partial\rho_i} - 1\right)$$

$$+ (\rho_i h_i - p)\frac{\partial V_i}{\partial t_i} = 0 \qquad (2)$$

Where *m* is the mass flow; positive and negative sign are associated to flow into and out of the control volume, respectively; *V* is the cubage of volume model; ρ is the density of airflow; *h* is the enthalpy; *u* is the internal energy and *i* is the subscript, representing the different element. For simplicity, dissipative works has been neglected here. Based on these balance equations, the basic energy equations in volume model can be acquired as following:

$$V_i\left(\rho_i\frac{\partial\rho_i}{\partial P_i} + \frac{\partial\rho_i}{\partial h_i}\right)\frac{dp_i}{dt_i}$$

$$= \left(\rho_i + h_i\frac{\partial\rho_i}{\partial h_i}\right)m_i - \frac{\partial\rho_i}{\partial h_i}u_i - \left(\rho_i^2 + \frac{\partial\rho_i}{\partial h_i}p_i\right)\frac{dV_i}{dt_i} \qquad (3)$$

$$V_i\left(\rho_i\frac{\partial\rho_i}{\partial p_i} + \frac{\partial\rho_i}{\partial h_i}\right)\frac{dh_i}{dt_i}$$

$$= \left(1 - h_i\frac{\partial\rho_i}{\partial p_i}\right)m_i + \frac{\partial\rho_i}{\partial p_i}u_i + \left(\frac{\partial\rho_i}{\partial p_i}p_i - \rho_i\right)\frac{dV_i}{dt_i} \qquad (4)$$

As Fig. 4 shows, between two volumes a flow model has to be connected with them. Here, the flow model is applied in the compressor, turbine, nozzle, etc.; to calculate the corrected mass flow rate of air based on the pressure ratio, temperature, angular speed, etc. In the flow model the mechanical power is transmitted to internal energy of the air by increasing the air pressure and vice versa.

3.2. Medium model

This paper wishes to provide the users with the capability to assemble an engine the same way as that in thermodynamics, that is, by coupling the physical components together using thermodynamic linkages, such as the working fluid flows and the shafts. So in this modular model library, it also includes a medium model for calculating the thermo-physical properties. The medium model is chosen according to the applications field.

Take the working fluid flow as an example. During the gas path balancing, it acts as a means of data transmission between two components. As depicted in Fig. 5, if component A and B are coupled together by a working fluid flow, the value of any thermodynamic parameter at A's outlet should be equal to that at B's inlet. If the value on one side is known and that on the other side is unknown, the fluid flow must assign the known value to the other side. The transmitted data are of various types, ranging from temperature, pressure, to enthalpy, mass flow rate, and/or chemical composition [8]. Some unexpected types of data may also need to be involved when the components are of special type. Besides, although the working fluid flows are unidirectional from upstream to downstream, the data transmissions do not have to behave in the same way. The previous models tried to wrap all these complexities in one single object, whose mechanism of data transmission turned out to be too complicated for new parameter types to be inserted in and hence new component types and balancing methods are extended and customized.

There are too many medium composition, it is not possible to create all of them. So based on the OO technology, a new medium model can be easily added by users follow the old medium model structure in the library.

3.3. Connector model

Component interaction is defined by Connector models. The connector represents an exchange of information data between successive models or components.

A consequence of allowing multifidelity and multidisciplinary models to be incorporated in an engine component is that an engine model maybe composed of component models having differing fidelity and discipline. There are different connector models available in the library. Such as: mechanical connectors, thermo connectors, hydraulic connectors, etc. Structure of the Connector is tied to the discipline level of the component model. That is to say, the different connectors are used corresponding to the variables transferred.

In this paper several connector combinations are introduced and classified into the sub-package. For example, there are two output ports in the high-pressure compressor of a recuperative gas turbine, one for the main

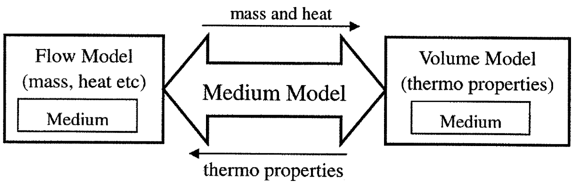

Fig. 5. Medium models.

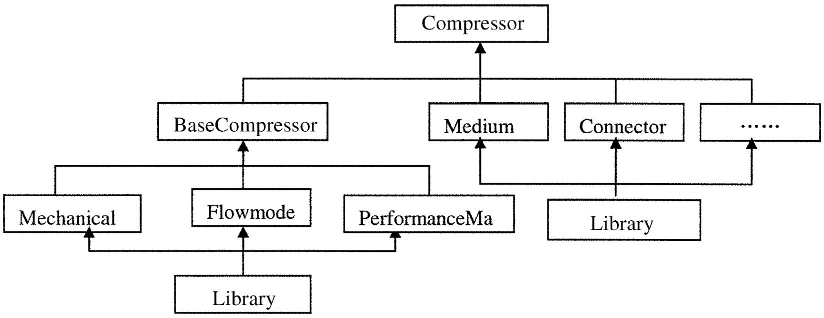

Fig. 6. Compressor flow model structure.

airflow and one for the coolant. So the three ports interface model named 'ThreePortsSingleDynamic' is introduced.

Although in some conditions, the thermo connector models used in the volume models are the same form as the flow models', the structures of them are different. In volume model pressure, specific enthalpy, density, temperature, specific entropy and ratio of specific heats of entrance connector are respectively equal to these of export connector, however, in flow model the mass flow and connective heat flow are the same but other variables are not.

4. Component example—compressor

In this section, a component of gas turbine—compressor will be taken as an example to illustrate how to use the model library explained above to build a component model.

The compressor model in the library can be divided into two parts: one is an essential flow model. In this model, the pressure potential and kinetic energy of the working fluid in the compressor will be increased by transmitting the mechanical energy of the shaft. And the other is the volume model. The compressor model is also used to model the Fan component in the engine model.

In the volume and flow models only thermodynamic parameters are considered, i.e. pressures and temperatures, as related to engine performance and performance analysis. The engine shaft speed is another important engine parameter. Modeling of the thermodynamics depends on the shaft speeds and vice versa. In this case generalized semi-empirical engine component models (performance maps) and mechanical model that are related to the compressor may be used.

Compressor performance is represented by a set of overall performance maps normalized to design point values. Baseline performance maps provide normalized corrected mass flow rate and normalized efficiency as a function of normalized pressure ratio and normalized corrected spool speed.

$$m_{\text{corrected}} = f(\omega, p, T) \tag{5}$$

The normalized adiabatic efficiency value is obtained from the baseline compressor performance map. The adiabatic efficiency η_{is}, is then computed by multiplying the normalized adiabatic efficiency by the designing adiabatic efficiency value.

$$\eta_{\text{is}} = \frac{\left(\frac{p_2}{p_1}\right)^{(k-1/k)} - 1}{\left(\frac{p_2}{p_1}\right)^{(k-1/\eta_p k)} - 1} \tag{6}$$

Based on shaft rotation speed, the pressure, temperature and enthalpy, etc., which are transferred from the previous and next volume models period. The power or rate of energy transmitted from the shaft to the fluid in compressor flow modes is:

$$P_{\text{comp}} = m(h_{\text{out}} - h_{\text{in}}) \tag{7}$$

Based on the OO technology, the compressor flow model can be built through the inheritance and the objects' composition. Its structure is shown in Fig. 6.

A complete compressor component model is constructed through combining compressor flow model and volume model. It is shown in Fig. 7.

5. Gas turbine models library and an example

5.1. Gas turbine library

The previous sections have described the architecture of Gas Turbine library, which is an object-oriented framework for modeling and simulation of aero gas turbine systems. This library can be classified into four parts: (1) Partial Components: it is the package of the aero-engine physical components and volume components models. These

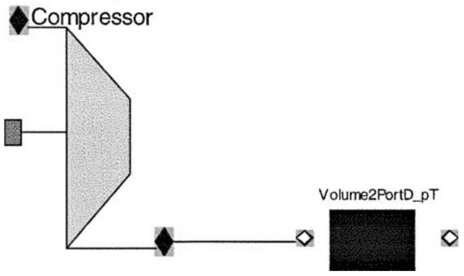

Fig. 7. Compressor model.

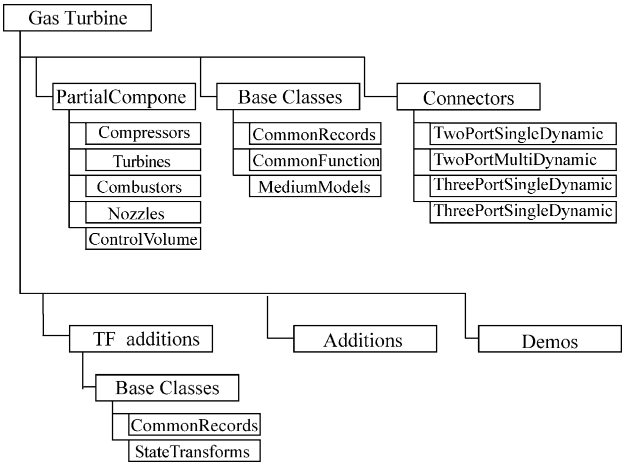

Fig. 8. Gas turbine library.

components are the basic blocks for building Aeroengine models; (2) Base Classes: they are the bases of other models in the library. They can be inherited or extended by other models. The records sub-package is the class of the variables and parameters, and the functions sub-package is the class of all functions. They are both put in the Base Classes package as sub-packages. (3) Interfaces: the connectors among the components are defined here; and (4) Demos. The structure is shown in Fig. 8.

With the OO technology this library provides a flexible and extensible environment that can be used to compose new engine component models graphically or programmatically, to inspect and edit the existing models, and simulate the display execution results. It also provides plug-compatible software components which users can combine to form increasingly complex gas turbine engine models according to application requirements.

5.2. An aero-engine model

In this section, we utilize the gas turbine library and electrical library developed by our team to graphically build and simulate the characteristics of gas turbine connecting with a switch reluctance generator. An acceleration process, the shaft speed from 7000 to 14000 rpm, is simulated. The compressor performance map is compared with the GSP's result. The NLR's GSP (Gas turbine Simulation Program) is an off-line component-based modeling environment for gas turbine. It is developed by the Netherlands National Aerospace Laboratory.

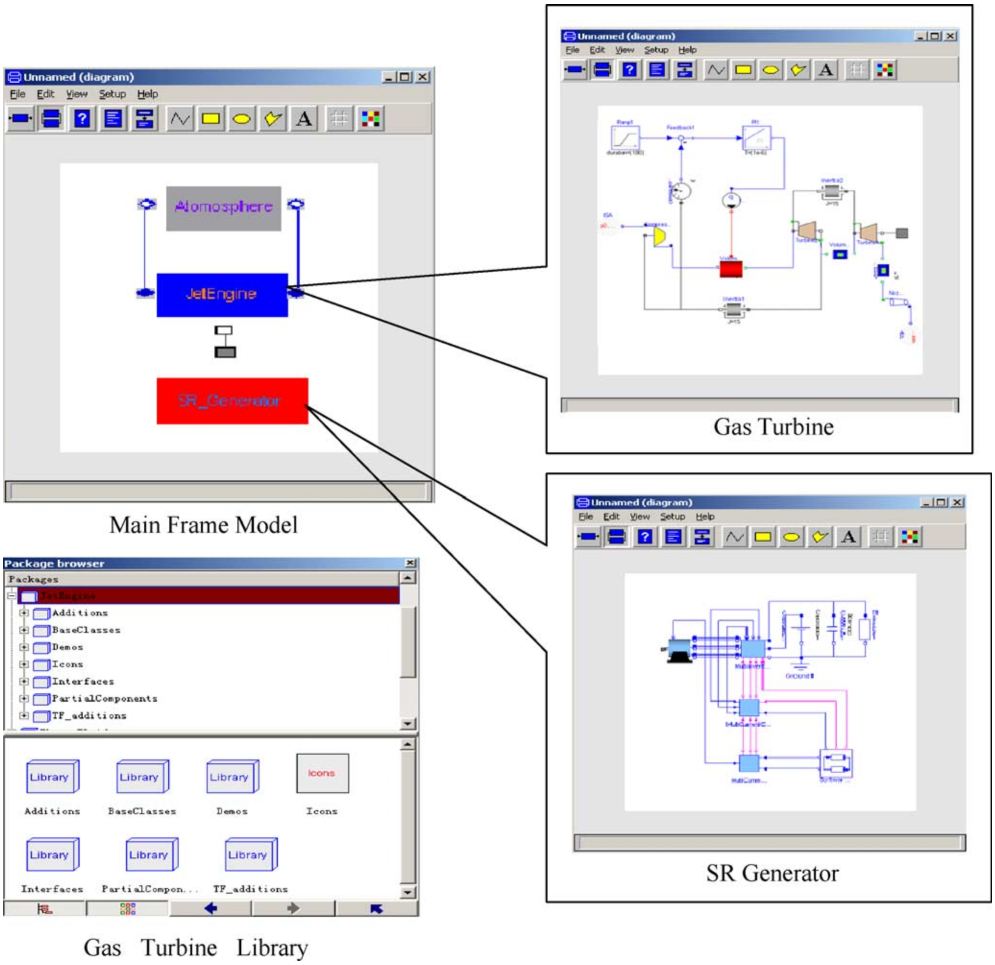

Fig. 9. Overview of gas turbine framework.

Components are the basic blocks for building engine models. In gas turbine library, a basic component model is represented by an icon. Components are dragged-and-dropped from library windows onto the model window and arranged to form the desired aero engine system configuration (see Fig. 9) according to the prototype. The main frame model (left figures of Fig. 9) is divided into three sub-models: Atmospheres, Gas Turbine Engine, and SR_Generator.

Atmosphere sub-model represents the environment. In this sub-model some variables, such as the atmosphere temperature, pressure, Mach number and so on are calculated.

The Gas Turbine Engine sub-model represents a two-spool engine. It includes the low-pressure compressor and turbine, combustor, high-pressure compressor and turbine, low and high-speed shaft, fuel pump, nozzle, power turbine. The volume models are placed between two flow models to define the initial conditions, boundary conditions and handle inter-component fluid dynamics.

The SR_Generator sub-model represents an aero generator. It is driven by the gas turbine through a gearbox.

In unsteady simulations, the independent variables are functions of time and may be implicitly integrated to predict their value at a future time based on known past values. So before proceeding to the transient analysis, the simulation

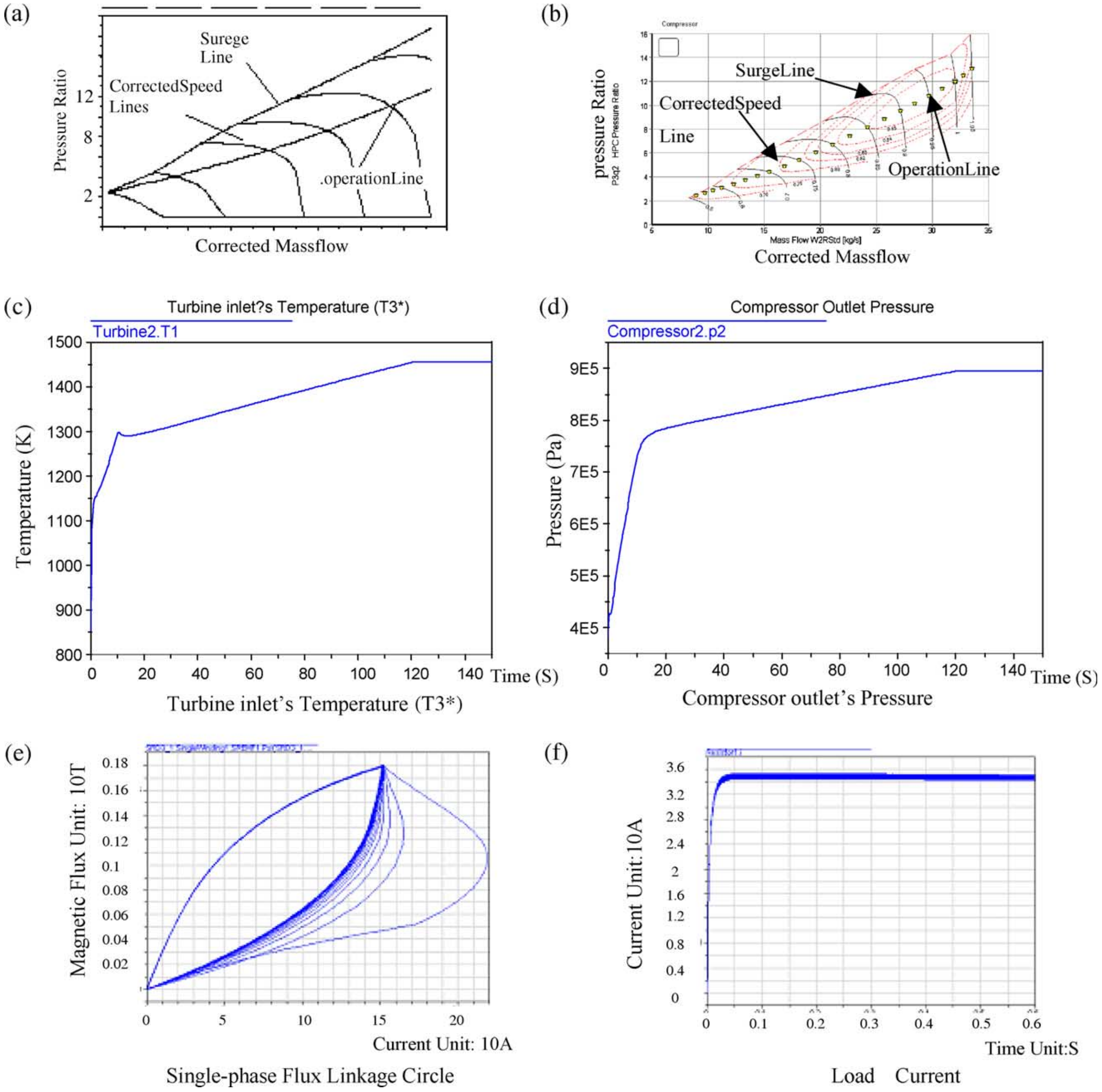

Fig. 10. Some results of the model.

first attempts to drive the engine to balanced (stable-state) conditions to make sure the values of initial operating point. This ensures that the engine model is in a consistent, physically valid operating state before beginning the transient operation.

After the model being built, each of these components is shown on a model window whose properties and characteristics can be specified by user via data modification. Compressor model has input fields named compressor customizer for defining design point values and performance map. So do the turbines, nozzle, and duct. The initial operating point values are set on volume model's customizer.

The DASSL numerical algorithm method is used to resolve the problem of the modeling. Relative tolerances have to be assigned to tell the integration methods how accurately the solution should be computed. The accuracy of the computed state variables is comparable to the relative error tolerances. The integration step-size is chosen in such a way, that the local error is smaller than the desired maximum local error, defined via the relative and absolute tolerance. The relative tolerance is 1E-4.

Once the initial and boundary condition values are fixed, the model can be simulated. When the simulation finishes successfully, then the results are available. All variables for each component in the simulation model can be plotted. Fig. 10(a) shows the compressor performance map, and Fig. 10(b) shows the same map of GSP. Comparing these two plots, we can conclude that our model is effective and correct, Fig. 10(c) and (d) show the temperature before turbine (T3*) and the compressor outlet's pressure. Especially, in Fig. 10(c) there is a decline of T3* during the acceleration process. With the mechanical inertia and cubage of volume model decreasing, the T3* decline will gradually reduce. That is to say, the inertia and the volume in gas turbine affect the acceleration remarkably. Fig. 10(e) and (f) also show the plot of single-phase flux linkage circle and load current of the SR_Generator model.

6. Conclusion

Designing and developing new aero engine is a time-consuming and expensive process. Computational simulation is a powerful means for alleviating this cost, thanks to the flexibility it provides for rapid and relatively inexpensive evaluation of alternative designs in design phase. So a framework named Gas Turbine Library representing engine components, subcomponents and subassemblies is developed and illustrated in this paper.

The Gas Turbine Library described in this paper provides framework components that, together, form an integrated system for aircraft gas turbine engine. It can help one to understand the engine performance behavior and to identify the causes of possible deficiencies in engine performance in a highly cost-effective manner. It has also become very useful for efficient mission analysis, the preliminary design studies of engines and their matching to airframes.

The object-oriented technology is utilized to produce a reusable component-based architecture that can be extended and customized to meet future application requirements. The modular individual components in this library can be easily modified and extended, so new components models can be added conveniently. As a result the library will be enlarged and changed by users, the library can be adjusted and updated by users with the aero engine developing.

The Dymola/Modelica is a new powerful simulation tool. In this paper it is applied in a new engineering field, and proved to be successful. On the other hand its libraries group has been enriched too.

Acknowledgements

The work described in this paper was made possible through funds from the United Technology Research Center (UTRC). We would like to thank all of the members of the team at Shanhai Jiaotong University and UTRC.

References

[1] Xie Z, Su M, Weng S. Extensible object model for gas turbine engine simulation [J]. Appl Thermal Eng 2001;21:111–8.

[2] Visser WPJ, Broomhead MJ. A generic object-oriented gas turbine simulation environment. ASME Turbo Expo 2000, Munich, Germany, 8–11 May 2000.

[3] RTO/NATO. Performance prediction and simulation gas turbine engine operation. Joseph Ottawa/Hull, Canada J8X 1C6.

[4] Bayt RL, Sahm MK, Lents CE. Collaborative engineering in integrated aircraft power system design. ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Canada, September 29–October 2, 2002.

[5] Michael MT. Introduction to physical modeling with Modelica[M]. Dordrecht: Kluwer Academic Publishers; 2001.

[6] Johnson RE, Foote B. Designing reusable classes. J Object-Oriented Programming 1998;11(2):22–35.

[7] Reed JA, Afjeh AA. Computational simulation of gas turbines. International Gas Turbine and Aeroengine Congress and Exhibition, Indianapolis, Indiana, June 7–10 1999.

[8] Pérez A.G. Modeling of a gas turbine with Modelica[D]. Master Thesis. Lund, Sweden: Department of Automatic Control Lund Institute of Technology; 2001.