<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *Chinese Intelligent Systems Conference, CISC 2015; Yangzhou; China*.

Citation for the original published paper:

Lu, J., Chen, D-J., Wang, J., Li, W. (2015)
An Investigation of Model-Based Design Framework for Aero-Engine Control Systems.
In: *Proceedings of the 2015 Chinese Intelligent Systems Conference* (pp. 625-638).
Springer
Lecture Notes in Electrical Engineering
https://doi.org/10.1007/978-3-662-48386-2_64

N.B. When citing this work, cite the original published paper.

This is a pre-print of an article published in Proceedings of the 2015 Chinese Intelligent Systems Conference. The final authenticated version is available online at: https://doi.org/10.1007/978-3-662-48386-2_64.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-181082>

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/300243446>

An Investigation of Model-Based Design Framework for Aero-Engine Control Systems

Chapter *in* Lecture Notes in Electrical Engineering · November 2016

DOI: 10.1007/978-3-662-48386-2_64

CITATIONS

0

READS

11

4 authors, including:

[Lu Jinzhi](#)

KTH Royal Institute of Technology

5 PUBLICATIONS 2 CITATIONS

[SEE PROFILE](#)

[De-Jiu Chen](#)

KTH Royal Institute of Technology

76 PUBLICATIONS 579 CITATIONS

[SEE PROFILE](#)

[Jiqiang Wang](#)

Nanjing University of Aeronautics & Astronau...

39 PUBLICATIONS 66 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:

TRACE [View project](#)

ESPLANADE (Efficient and Safe Product Lines of Architectures eNabling Autonomous Drive) [View project](#)

All content following this page was uploaded by [Lu Jinzhi](#) on 26 October 2017.

The user has requested enhancement of the downloaded file.

The Investigation of Aero-engine Control System Design Approach based on Model-based Framework

Jinzhil Lu^{1,2}, DeJiu Chen¹, Lei Feng¹, Jiqiang Wang³, Wentao Li²

1. Industrial Engineering and Management, KTH Royal Institute of Technology, Stockholm 100 44

2. AVIC, Shenyang Engine Design and Research Institute, Shenyang, 110034

3. Nanjing Aeronautics and Astronautics University, Nanjing, 210016

E-mail: jinzhl@kth.se

Abstract: In the whole aero-engine design phase, modeling and simulation technologies have been widely used for supporting the conceptual design and performance design. Due to the increasing complexity of aero-engine systems, there are however new challenges in the areas of quality management and process optimization. In particular, the integrations of various domain physical models for early system quality assessment are becoming a challenging area for the reason of quality management and engineering efficiency. In this paper, we present the initial concepts of a framework based on formal system models for allowing automated management of simulation services in the development of aero-engine systems. We exploit in particular EAST-ADL as the framework for system modeling, which captures the design and implementation of target systems, including the related requirements, constraints, verification and validation cases, etc. With the specification of verification and validation cases as inputs, an additional co-simulation service will provide the support for an automated configuration and execution of simulation tools. Currently, the co-simulation service is specified and managed based on SysML.

Key Words: Aero-engine, Control system, System Design, Model-based Framework.

1 Introduction

Aero-engine control systems have been evolved continuously over the past decades. The control logic is implemented from the traditional mechanical system to FADEC (Full Authority Digital Engine Control) including flexible fuel systems, electronic systems, controllers, etc. The functional and technical complexity of aero-engine control system continuously increases and more engineers and designers are needed to participate in the design activities of aero-engine design life cycle than the past [1].

Based on the view of Dale Ennsa[2], prevailing scheme to design the aero-engine control system is based on the divide and conquer approach to many complex engineering tasks. From Andrea Prencipe [3], we know that in the institutes or companies for aero-engine design, the whole complex system will be separated into different parts and there are different groups and companies to design the specified

components or subsystems. Designers in different groups often use the specific domain simulator tools to assist their design job in order to run some system analysis in advance.

The performance of the aero-engine is strongly dependent on the performance of each subsystems which in turn dependent on the characteristics of different components in them. So it's important for system integrators to understand the system behaviors and analyze system performance in the initial design phase.

As we know, the whole design phase includes requirement analysis, high level design, detailed level design, implementation, test issues, etc. Nowadays, different CAD and CAE tools can help designers to build models and design these subsystems to satisfy the demand of each phase. But the whole system performance is very different to predict, because the designers of subsystems often use their own tools to build models and it's hard for system integrators to integrate the detailed models to predict the whole system performance. Now there is not a completely integrated framework for different kinds of models based on a management perspective.

This paper is concerned with model-based technical roadmap and methodology of information integration for aero-engine control system model-based design approach. We use system models as a formal description which can be shared with stakeholders, including project managers, safety engineers, system engineers, modeler, system designer and simulation testers. The ontology is imaged to own the capabilities to describe requirements, parameter setting,

interface contrasts and information which need to be used for other model users during simulation. Also, optimization behaviors used for model parameter settings can also be added in the framework of ontology.

1.1 Structure

This paper has four main sections. In Section 2, there is a brief introduction of simulation technologies, Modelica and co-simulation. The Section 3 argues system modeling approach, EAST-ADL, AADL, SysML, etc and test case to describe a co-simulation procedure of fuel distributor. The Section 4 discusses mapping between system models and physical system models. The Section 5 provides a roadmap for future aero-engine control system design architecture.

2 Common simulation technology for aero-engine control system

Time consumption and R & D cost have become more important, since more new technologies are used for the aero-engine design. Nowadays, computation design and simulation technology can help to improve the former key factors [4]. In the aero-engine design phase, model-based design is very important to predict the system performance in advance. Designers have used many types of modeling technologies. Software program was firstly used to design the computational aero-engine models [5]. After that with the help of modularly structured code, the code of this specified software program for aero-engine became reused and objected-oriented. Then in [7], the object-oriented and visual modeling method advanced to predict the system performance. Lately in [4], Modelica language is used to build the aero-engine model. Many component libraries can be built in the simulator tools which strongly improve the efficiency of building models and model reuse. In [8], AMESim is used for fuel system modeling and in [9], hydro-mechanical controller component model is built in AMESim and Simulink is used for aero-engine controller model. Then co-simulation is used for predicting the performance of hydro-mechanical controller behaviors. This chapter discusses Modelica, co-simulation technology and approach of model scaling to separate physical models of aero-engine.

2.1 Modelica

Modelica language is a non-proprietary, object-oriented, equation based language to conveniently model complex physical systems [10]. In the Modelica platform, the aero-engine control system model can be separated into models in different areas. These models can be mathematically described by differential algebraic equations and discrete equations. And these models are object-oriented and schematics. From the point of view in [4], Modelica has been used for aero-engine system design.

2.2 Co-simulation

Co-simulation is a technology which can solve the multi-physics simulation integration [11]. It means a particular case of simulation scenario which there are at least two simulators solve coupled-algebraic equations and exchange the data with each other during simulation [12]. In [13], co-simulation can be defined based on the different behaviors of solvers. Fig.1 demonstrates 3 different types of co-simulation.

- Co-simulation

Co-simulation means simulator tools use their own solvers to calculate their models and communicate with each other at the fixed time step point.

- Coupled-simulation

Coupled-simulation represents subordinated simulators transport their models into a formal mathematical descriptions which the main simulator can read and be integrated into its model. Then the main simulator uses its own solvers to calculate the coupling models.

- Solver-coupled simulation

Solver-coupled simulation is a co-simulation method which subordinated simulators can transport their models and solvers into the form which the main simulator can understand. Then the solver of main simulator tool can call the solvers of the other simulators to calculate corresponding models.

Fig. 1 co-simulation, coupled-simulation and solver-coupled simulation

From [9], we know, co-simulation has also been one technical solution for aero-engine control system design. However, no matter Modelica or co-simulation, simulation technology is used to help system designers to take decisions based on different design goals. When system designers run a 'project', there are different models used in each phase of the model-based design life cycle, even though the modeling object is fixed [1]. So it's very different to manage the models, because models present their builders' option on the matter. So a new standards or technologies should be provided to solve these problems.

2.3 Model scaling and management

Traditionally, the model-based design procedure of aero-engine control system is separately disintegrated process of different subsystems. Currently, there are no industrial standards to define the aero-engine control system

model types or interfaces. But, in fact, we can distinguish (a) "control logic", (b) fuel system or hydraulic system, (c) mechanical system, (d) CFD for aerodynamic, (e) FEM for structure design, (f) thermodynamics of combustor, (g) electronic system and many more. So from model management view, we need to define the model libraries based on the design phase and detail level of models [14, 15]. From the perspective of Marco Bonvini [14], for each level, such properties should be shown including purpose,

hypotheses, analysis protocol, structural limitations, practice-based limitations and decision-making usefulness of the models, etc. In different design phase, models can be used based on their properties of detail level. Table.1 provides a brief example of subsystem characteristics for different levels.

Table 1 Overview of subsystem model properties for detailed level setting

Subsystem	Detail level 1	Detail level 2	Detail level 3	Detail level ...
Control logic	Basic control logic	Basic control logic	Control Scheme	...
Fuel system	Sample Model(No fluid leakage, incompressible flow, Laminar orifice)	Sample Model(No fluid leakage, incompressible flow, Laminar orifice)	More Realistic Model(fluid leakage, compressible flow, Laminar orifice)	...
Hydraulic system	Sample Model(No fluid leakage, incompressible flow, Laminar orifice)	Sample Model(No fluid leakage, incompressible flow, Laminar orifice)	More Realistic Model(fluid leakage, compressible flow, Laminar orifice)	...
Mechanical system	Kinematic	Kinematic	Kinematic and multi-body dynamic	...
Electronic system	Principle models	Basic element models	Complex system model	...
Aerodynamics	Fixed operating point (parameter)	Map based		...
FEM for structure	Fixed operating point (parameter)	Map based		...
Thermodynamics	Fixed operating point (parameter)	Map based, input filtering		...
...				...

3 System modeling approach

From the perspective of system engineering, physical system model is not the unique concern. Though there are a lot of complex and completed model libraries or model management platform existing, it's also different for system designers in each layer to understand the whole physical system model and share the model information with each other except for documents or reports. In this chapter, several system modeling tools, (e.g., SysML, AADL, EAST-ADL, etc.) are investigated. Based on different system modeling language, ontology of simulation for aero-engine control system is designed.

3.1 SysML

SysML is a general-purpose modeling language for systems engineering with a subset of UML2 and additional extensions to satisfy the demands of the language description. SysML can provide powerful and simple constructs for modeling systems engineering problems of requirements, structure, behavior, allocations, and constraints [16]. In [17], Henson Graves provided an approach of SysML using in the aerospace design. So SysML has the capabilities to describe the simulation behaviors.

3.2 AADL

AADL is formal modeling concepts for description and analysis of systems architecture in terms of individual

components and their interactions including software, hardware and system component abstractions. It can specify and analyze real-time system, embedded system, complex SOS (system of system) and map software onto computational elements in hardware. So AADL has the potential to describe the architecture and behaviors of the aero-engine control system.

3.3 EAST-ADL

EAST-ADL is a language to describe automotive electrical and electronic systems. It can grasp all the detailed information for documentation, design, analysis and synthesis from the top level characteristics to tasks and interface & communication framework [19]. As we know, there are many similar features between aero-engine control system and automotive electrical and electronic system, so EAST-ADL also has the potential to describe the system characteristic of aero-engine.

In Section.2, we have introduced that physical system model should be scaled and managed in different phase and simulation setting with interface or solve also need to be shared with other system designers. So ontology needs to describe such characteristics and features. Next part, we will discuss about the capabilities of different language and build a system model as a case for an integrated model of co-simulation method.

3.4 Comparison between SysML, AADL and EAST-ADL

The framework of ontology should be able to share and information for several behaviors of physical system modeling procedure, constructs of aero-engine models, simulation requirements, parameter setting, simulation

setting, optimization behaviors and validation & verification, etc. Table.2 shows the different capabilities three kinds of language which can describe such characteristics. System integrators can choose the models in different levels based on simulation targets and requirements.

Table 2 Behaviors of simulations and capabilities of three languages

Subsystem	SysML	AADL	EAST-ADL
Simulation Requirements	Requirements	Annex libraries(AL)	Requirements
Constructs for Models	Blocks	System Structure and Instantiation	No
Modeling Procedure	Activity (No specified)	No	SystemModeling
Simulation setting	No	No	FeatureModeling
Interface setting	Point	Point	FunctionConnector
Validation & verification	Activity (No specified)	No	VerificationValidation
Optimization behaviors	Activity (No specified)	No	ComputationConstraint
...			

From the Table.2, we can see the characteristics which a simulation of physical system models have can be described by EAST-ADL. In 3.5, system model based on EAST-ADL and SysML will be shown to describe an integrated model of co-simulation for aero-engine control system.

Fig. 2 Extensive structure of system model

In Fig. 2, we can see a modeling structure. Behaviors models can use to describe control system model in Simulink. Requirement diagram is used for describing the requirements of stakeholders. Simulation diagram can contains simulation setting and solver setting. Structure diagram can describe the physical system models in different tools. Data diagram can be used for managing data from simulation. Feature diagram can be used for describe the behaviors of simulation testers and stakeholders.

3.5 User case for system model describing physical system model

From [12], the author uses co-simulation between Saber, Simplorer, Flowmaster, AMESim and Simulink to predict the performance of fuel distributor in the aero-engine. Based on this case, system model of this co-simulation procedure is built. Fig. 3 and Fig. 4 show the co-simulation strategy for flue distributor model from different tools. A set of system models will be built to describe this event as an example.

Fig. 3 Co-simulation strategy for flue distributor

ig. 4 Model of co-simulation for flue distributor

Depending on the extensive structure of SysML, system model is built to describe the procedure of verifying if co-simulation integrated model can product the same simulation result from the system model only built in AMESim [12]. From Fig. 5, we can see, five design phases are separated, including system requirement analysis, model design, subsystem modeling, co-simulation and verificationValidation.

Fig. 5 System Model describe different design phase

physical system models can be built from model libraries
and

Fig. 6 Requirement Description

In the model design phase, feature model can depict the properties of each component in the top system model and setting selection of the co-simulation in Fig.4. In this phase, interface design strategy can be made and each subsystem model can be related to corresponding physical system model. In Fig. 8, the construct model of the top system model can be produced by the feature model.

Fig. 7 Feature function describing Model Design

Fig. 8 Construct block for the top system model

During subsystem model design, the construct feature of the physical system model can be shown. The components, lines and parameters in each subsystem model will be acted as the elementary block in the feature model, which is shown in Fig. 9. Based on the feature model, construct model for each tool can be built in Fig. 10. The model selection in different tools can be decided by modeling target and model complex level depending on construct models. The really

Fig. 9 Description of Feature function for subsystem model design

Fig. 10 Subsystem model in Simplorer

The system model for co-simulation includes the information of interface between different tools, parameters setting during co-simulation, solver behavior and data information in Fig. 11.

Fig. 11 Feature function for co-simulation

Fig. 12 Verification Validation Procedure

In the phase of VerificationValidation, block Log can represent the operation during VerificationValidation. Based on the requirement, system model uses Parameter

Matrix (1) to run the co-simulation and compares the simulation result with the results provided by paper.

4 Mapping between System models and physical system models

During co-simulation, different parameter matrix will be used for different scene, scenarios and situation. Here, scene means a set of tasks of one simulation target. Scenarios signify different models in one simulation task. Situation represents different parameter matrix for one model in one simulation target. Specified parameter matrix for one situation is all the parameters for physical system models. When the parameter matrix for the physical system model is changed, this behavior can be described by VerificationValidation case in last section. And the capabilities of mapping from system model into physical system model are needed in order to run the simulation automatically.

From the view of Arne Seitz [20], he provides several schematic for integration of propulsion system into aircraft conceptual design process. The framework can dynamically set the configuration of different conception models. In our framework, we can see the schematic of dynamic configuration of simulation parameters from system model to physical system model. In Fig.13, parameter and data model can be dynamically configured based on the block 'VVlog'. 'VVlog' can set different parameter matrix from 'VVStimuli'. After co-simulation, data model returns the data to 'ActualOutcome'. Depend on the comparison between 'ActualOutcome' and 'IntendResult', simulation target will be checked and system will decide if the simulation model needs to continue the iteration.

Fig. 14 Schematic for dynamic parameter setting

5 Future work for aero-engine control system model-based design framework

In the future, a model-based design framework should be developed to improve the efficiency of aero-engine control system design. In Fig.11, four layers should be arranged including the engineering management platform, business process model layer, system model layer and physical system model & data model layer. Stakeholders can manage the project on the engineering platform. Depend on the information of the engineering management platform, business process model can be built to manage the project, time plan and human resource, such management issues. In different projects, system model can be used to describe the simulation behaviors including requirements, safety, behaviors, functions, verification Validation and so on. Through the information provided by system model, different component models and constructs will be selected from the data libraries and model libraries to build physical system model.

Fig. 13 Framework for model-based aero-engine control system design approach

In the future, ontology of each model layer should be developed to describe all the information which can capture the corresponding model. And the technical approach should be developed in each layer of the whole framework. An intelligent automation system will be developed for system models transporting to physical system models.

6 Conclusion

From the test case, we can see EAST-ADL and SysML can describe the information of simulation procedure. Dynamic parameter setting can support to capture all the system behaviors of parameter optimization. But now this framework has a limitation. The models mentioned in this paper are the ones for physical system and control system analysis which are tool-based or language-based. So currently, geographical models haven't been considered for managed and description. In the future, dependent on model-based framework, models of different layers can be integrated by modeling mapping technology and the efficiency of the system design will be improved if an integrated system can be developed.

References

- [1] Dong Zhang, Jin-zhi Lu, Lin Wang, and Jun Li, Research of Model-based Aeroengine Control System Design Structure and Workflow, Asia-Pacific International Symposium on Aerospace Technology, 50(4): 511–515, 2014.
- [2] Dale Ennsa, Dan Bugajskia, Russ Hendricka & Gunter Steina, Dynamic inversion: an evolving methodology for flight control design, in International Journal of Control, 1994: Volume 59, Issue 1 71-91.
- [3] Andrea Prencipe, Technological competencies and product's evolutionary dynamics: a case study from the aero-engine industry, Research Policy 25 (1997) 1261 – 1276.
- [4] Grid-based distributed simulation of an aero engine, in the International Journal of Advanced Manufacturing Technology, 2006, Volume 27, Issue 7-8, pp 631-637.
- [5] Daniele CJ et al (1983) Digital computer program for generating dynamic turbofan engine models. NASA TM-83446
- [6] Schobeiri MT, Attia M, Lippke C (1994) GETRAN: A generic, modularly structured computer code for simulation of dynamic behavior of aero- and power generation gas turbine engines. J Eng Gas Turbines Power 55–66
- [7] Visser WPJ, Broomhead MJ (2000) A generic object-oriented gas turbine simulation environment. In: ASME Turbo Expo 2000, Munich, Germany, 2000
- [8] P. Kai, F. Ding, Z. BU, J. LI, F. YIN, "Failure analysis and parameter optimization for fuel distributor for aeroengine," Journal of Propulsion Technology, Vol.32, No.2, April., 2001(Chinese).
- [9] Yingqing Guo, Kuo Li, "Modeling of Aeroengine Hydro-mechanical Controller and Analyzing of the Control System Performance", 2009 Fourth International Conference on Innovative Computing, Information and Control, IEEE, 446 – 449, 2009
- [10] Modelica® - A Unified Object-Oriented Language for Systems Modeling, <https://www.modelica.org/>, 2012
- [11] Sicklinger, S.; Belsky, V.; Engelmann, B.; Elmqvist, H.; Olsson, H.; Wüchner, R.; Bletzinger, K.-U, "Interface Jacobian-based Co-Simulation". International Journal for Numerical Methods in Engineering 98 (6): 418–444.
- [12] Jinzhi Lu, Jian-wan Ding, Fanli-Zhou, Research of Tool-Coupling Based Electro-Hydraulic System Development Method, IEEE International Conference on Industrial Engineering and Information Technology, 2014.
- [13] Jinzhi. Lu "Co-simulation for heterogeneous simulation system and application for aerospace", Master. dissertation, Univ. Huazhong University of Science and Technology, Wuhan, China, 2011.
- [14] Marco Bonvini, Alberto Leva, Scalable-detail modular models for simulation studies on energy efficiency, Proceedings 8th Modelica Conference, Dresden, Germany, March 20-22, 2011.
- [15] Philip Jordan, Gerhard Schmitz, A Modelica Library for Scalable Modelling of Aircraft Environmental Control Systems, Proceedings of the 10th International Modelica Conference, 2014, Lund, Sweden.
- [16] OMG Systems Modeling Language (OMG SysML™), <http://www.omg.org>.
- [17] Henson Graves & Yvonne Bijan, Using formal methods with SysML in aerospace design and engineering, Annals of Mathematics and Artificial Intelligence September 2011, Volume 63, Issue 1, pp 53-102.
- [18] Peter H. Feiler, David P. Gluch, John J. Hudak, The Architecture Analysis & Design Language (AADL): An Introduction, CMU/SEI-2006-TN-011
- [19] EAST-ADL-Specification_V2.1.12, EAST-ADL Association, <http://www.east-adl.info/>
- [20] Arne Seitz, Advanced Methods for Propulsion System Integration in Aircraft Conceptual Design, Technischen Universität München, München, Germany, 2011.