Proceedings of the 1999 IEEE
International Symposium on Computer Aided Control System Design
Kohala Coast-Island of Hawai'i, Hawai'i, USA • August 22-27, 1999

A graphical user interface for flight control development

Reinhard Finsterwalder

Universität der Bundeswehr, München
Wissenschaftliche Einrichtung Mathematik & Informatik
D-85579 Neubiberg, Germany
Reinhard.Finsterwalder@unibw-muenchen.de

Hans-Dieter Joos, Andras Varga

DLR Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Robotik und Systemdynamik
D-82234 Weßling, Germany
Dieter.Joos@dlr.de Andras.Varga@dlr.de

Abstract

The FSA (First Shot Approach) Demonstrator of DLR and DASA/Airbus is a first prototype for an integrated multidisciplinary environment for flight control development. The FSA Demonstrator represents a state-of-the-art computational tool which facilitates the study of trade-off between competing specifications and performance metrics. It integrates an object-oriented modeling environment, a data and tool management system, general purpose system analysis, simulation and synthesis tools and an automatic multi-goal attainment optimization program. The FSA Demonstrator comes with a dedicated graphical user interface for problem setup and pushbutton program operation which allows easy setup and operation even by non-specialists. The main payoffs of the FSA are a significant reduction in the design cycle and an improved performance of handling qualities.

Keywords: Flight control development, graphical user interaction, visual decision support, Java programming language

1. Introduction

The DLR Institute for Robotics and System Dynamics in conjunction with DASA/Airbus are currently specifying the functional requirements for an integrated multidisciplinary framework for flight control development. For exploration of the potentials of such a framework, a prototype has been implemented referred to as FSA Demonstrator .

The basic view behind the FSA Demonstrator is the assumption that the flight control system developer has already conducted a preliminary functional design study to determine an appropriate control-law architecture. Then the control system analyst can use the FSA Demonstrator to evaluate the baseline design and to tune the design parameters for best system behavior. FSA implements the process model as shown in Figure 1.

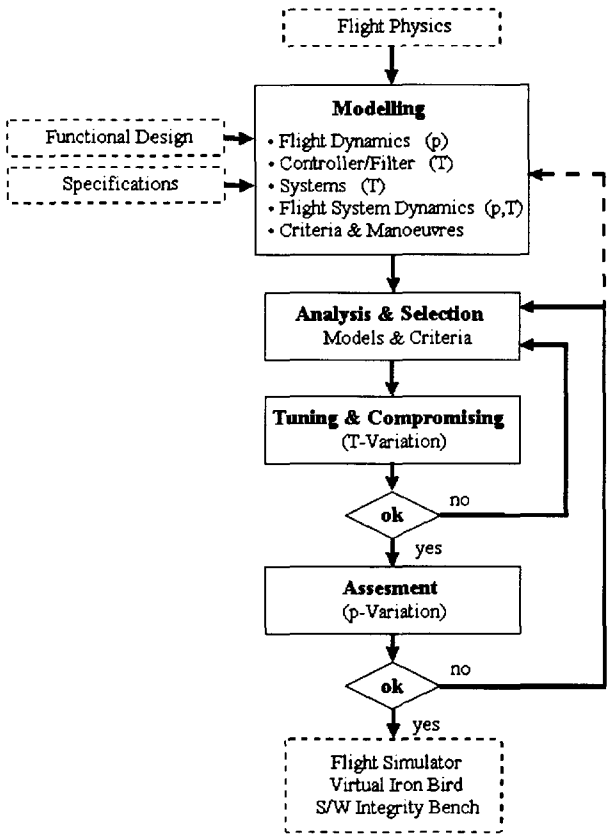

Figure 1: FSA design process

The FSA Demonstrator framework integrates an object-oriented modeling environment (Modelica / DYMOLA) [1], general purpose system analysis, simulation and synthesis tools (ANDECS) [4], interactive result visualization (ANDECS_AVIEWER), a data and tool management system (ANDECS_RSYST) [5], and an automatic multi-goal attainment optimization tool (ANDECS_MOPS) [7]. Predefined computation chains are provided for special tasks like trimming, linearising, eigenvalue computation including result visualization.

Computation experiments are controlled by a graphical user interface (GUI) that is adopted to the special needs of the

flight control design process. All design steps are supported by dedicated menus. Figure 2 shows the main window of the FSA demonstrator. The pulldown buttons in the menubar reflect the different phases of the design process:

- *Project* to create a new workspace / to open an existing workspace
- *Model* to start the modeling environment DYMOLA
- *A/C-Analysis* to analyze the open-loop aircraft dynamics for different operation conditions
- *MOPS* to perform a multi-objective optimization based controller tuning
- *Assessment* to perform post-design assessment and worst-case analysis for the (optimized) closed-loop system

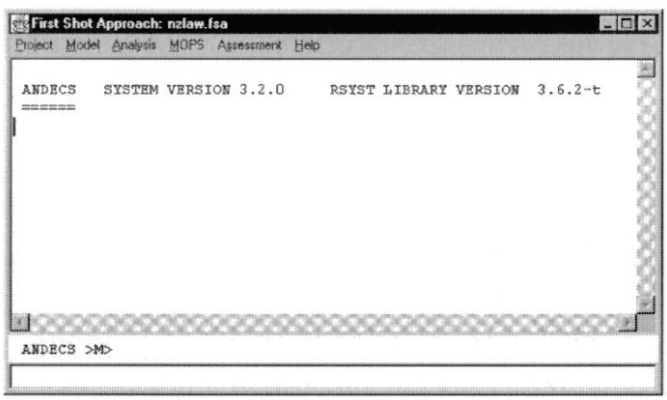

Figure 2: FSA Demonstrator–main window

2. Modeling

Parameterized models of the aircraft dynamics (A/C) and of the flight control system (FCS) are needed as basis for analysis and design computations. Both the aircraft model parameters p as well as the free controller/filter/systems parameters T are used as search variables either to perform worst case parameter studies or for optimization based automatic controller tuning, respectively.

The object-oriented modeling language Modelica/DYMOLA [1] is used for modeling of the aircraft dynamics. For that a flight-mechanics class library has been developed. This library contains all the elements needed to define aircraft flight dynamic models, including different types of engine, atmosphere and gravity models. Model building is done by means of a graphical object diagram editor, as shown in Figure 3. By using graph-theoretical algorithms and symbolic formula manipulations, DYMOLA transforms the object diagram into state space form and generates efficient simulation code for several simulation platforms.

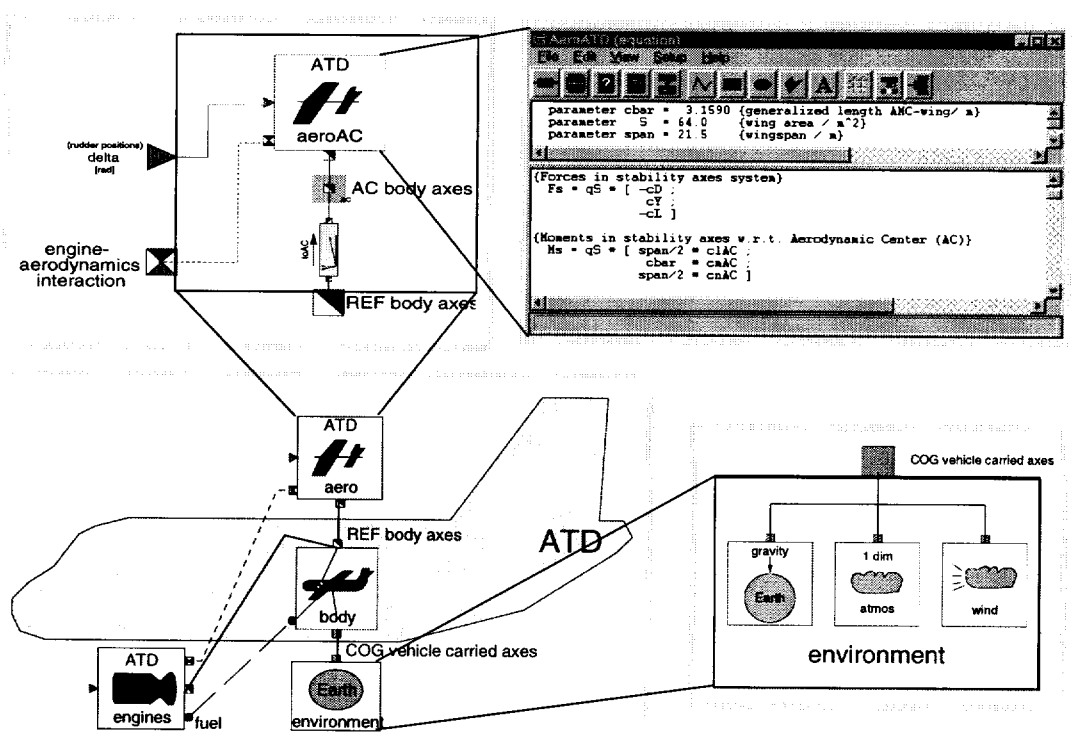

Figure 3: Object-oriented modeling of aircraft dynamics [9]

3. A/C-Analysis

The purpose of A/C-Analysis is to explore the open-loop aircraft dynamics for different flight conditions (load mass, fuel, etc.). This involves the simulation of nonlinear A/C models, trimming and linearising in predefined points of the flight envelope, application of linear analysis methods (e.g. eigenvalue computation).

The A/C-Analysis is mainly used to select representative points on the flight envelope. The corresponding linear models define the multi-model used for controller parameter tuning and compromising. The selection of models is done interactively using the graphical outputs of the simulation and computational results over a complete sweep of predefined trim conditions. The left part of Figure 4 shows the graphical user interface for aircraft analysis. This menu gives information on the model parameters *p* (name, current value, minimum, maximum, number of intermediary grid points) and on existing predefined parameter sets and trim conditions.

Three experiment modes are provided:

- *One Experiment*: single analysis experiment in the current trim point
- *Parameter Study:* variation of one or more parameters
- *Scan DB:* visualize precomputed analysis runs that are stored on database

Clicking on *Compute* starts a new analysis run. The results of an analysis experiment are automatically displayed in multiple windows (see right part of Figure 4). Furthermore multiple analysis results originating from successive analysis experiments can be visualized simultaneously. Advanced user interaction techniques support the fast selection of representative models among a large number of available parameter sets. Picking any point/trajectory highlights all curves belonging to that experiment. In addition the corresponding parameter values are displayed.

4. Tuning and Compromising

Tuning & Compromising is based on a multi-criteria/multi-model parameter tuning facility which provides a systematic way for optimization based control law tuning by directly specifying bounds and demands on FCS specifications and handling qualities as well as physical control realization constraints [8]. The variations in the model parameters and operating conditions are covered basically by the multi-model formulation. For a given control structure, the free parameters of the controller, *T*, are automatically tuned by optimizer to their best values satisfying the specifications and flying/handling quality demands. The multi-goal tuning of parameters is achieved by using the optimization tool ANDECS_MOPS.

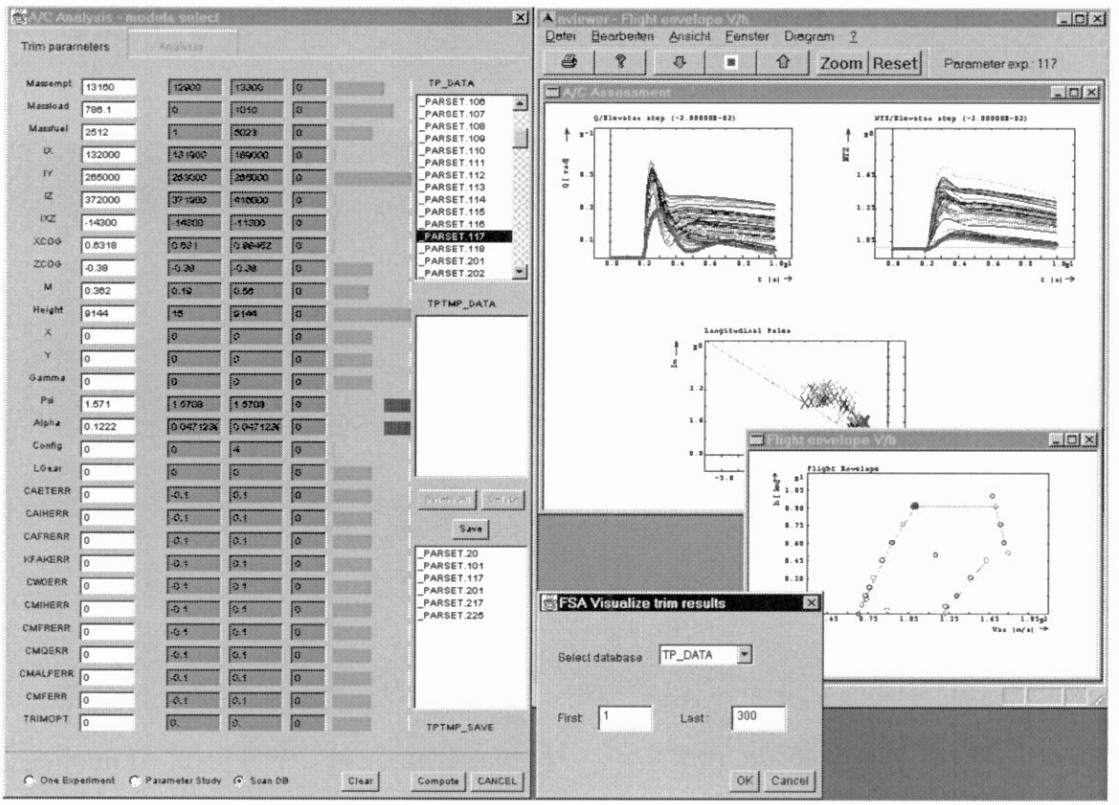

Figure 4: A/C Analysis – visualization of pre-computed results

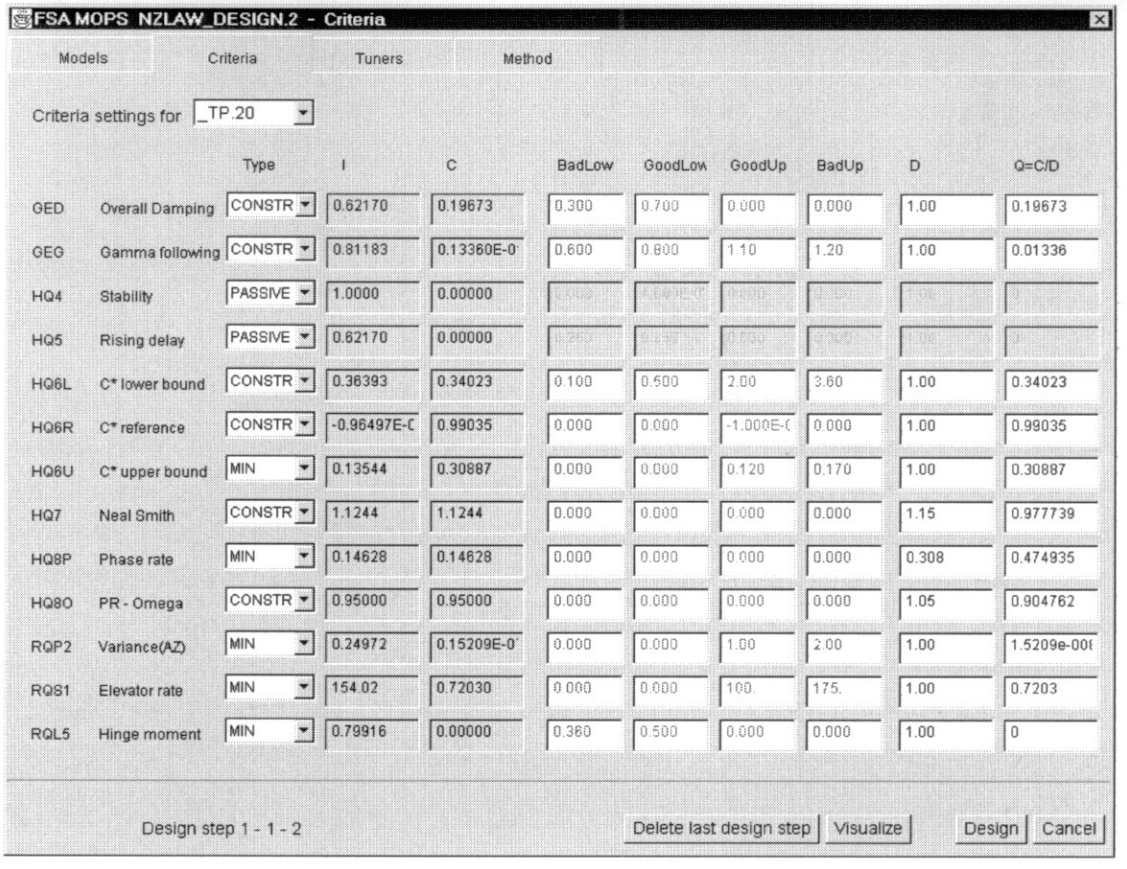

Figure 5: Tuning & compromising – submenu *Criteria*

The set up and steering of an optimization experiment is done via four submenus: *Models, Criteria, Tuners, Method.* Figure 5 shows the submenu *Criteria*. This mask serves for input of criteria related data, like demand values *d*, or the definition of a criterion as a constraint. The criteria on display are standard design objectives in flight control.

When the multi-objective parameter tuning is running, all intermediate iteration steps are visualized to give the user maximum insight into the tuning process. At any time the computation can be interrupted. Then the user can examine actual and previous results with the help of the following graphical facilities, shown in Figure 6:

- Selecting a particular design candidate by picking any point/curve
- Scanning the design history stepwise (forward/backward) yields all data belonging to a specific design iteration being visualized
- Trade-off analysis via the parallel coordinate editor [3], [6] gives information about what criteria are competing and how strongly

Then the best design candidate is selected and can be stored on the project database. It can also be used as starting point for further optimization.

5. Assessment

Task of the assessment is to detect hidden weaknesses in the designed FCS. Usually systematic gridding based parameter studies are performed where a large number of parameter combinations and different operating conditions have to be examined. A more efficient way is to search worst cases by parameter optimization, i.e. for given optimal tuner values T^* the worst-case models parameter p^* are searched such that a selected performance criterion, e.g. stability, is as bad as possible. Worst case models can be used to update the multi-models used for tuning & compromising.

To perform assessment the user has to specify one or more design cases to be used as a single constant controller or as a gain-scheduled controller, respectively. The set of available designs is displayed in the controller window, where for each selected design, the corresponding speed value is also displayed, see Figure 7. From several designs at different speed values, a gain-scheduled controller can be assembled to cover all air speed values between 0 and 500 m/s. The controller gains between two speed values are linearly interpolated.

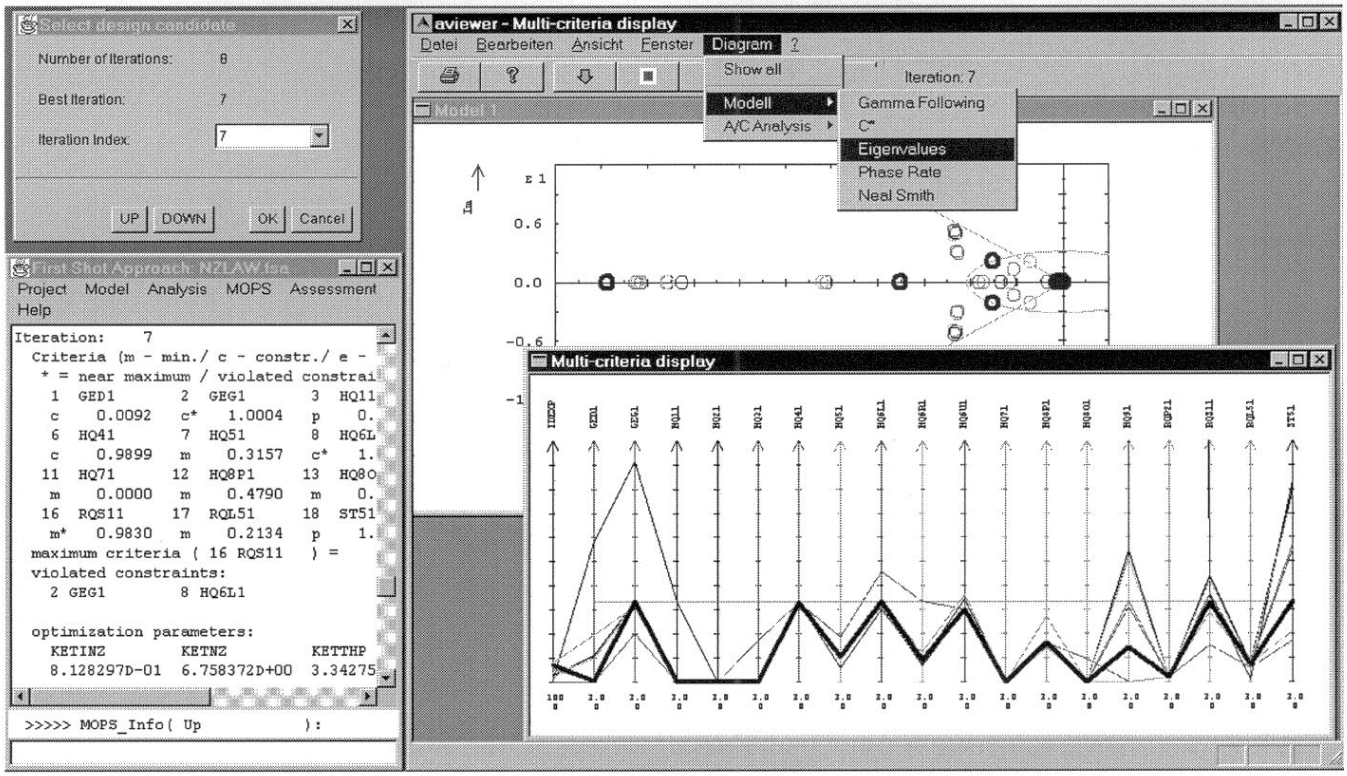

Figure 6: Tuning & Compromising – visual decision support

To facilitate the speed selection, the flight envelope together with the trimming points used for all existent designs are displayed in the graphical output window. The trim points corresponding to one design (usually at the same speed) are displayed with the same colour.

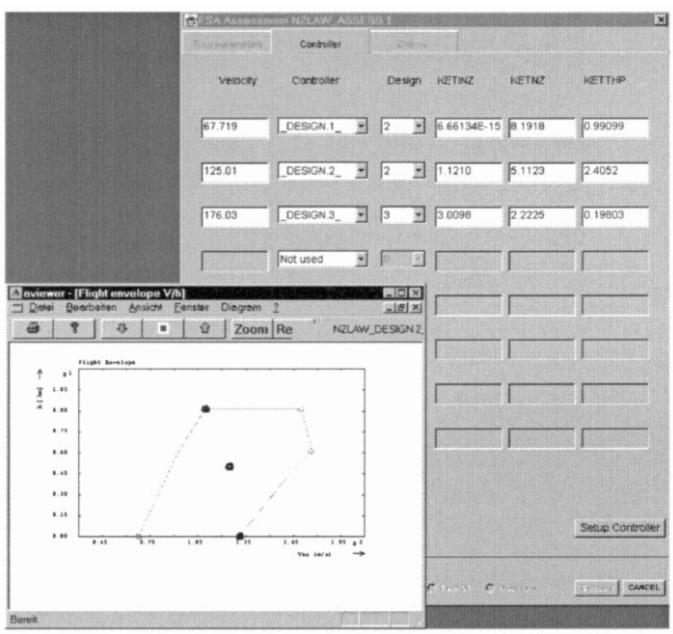

Figure 7: Assessment – setup of a gain-scheduled controller

Four experiment modes are provided:

- *One Experiment*: single analysis experiment in the current trim point
- *Parameter Study*: variation of one or more parameters
- *Scan DB*: visualize precomputed analysis runs that are stored on database
- *Worst case:* worst case parameter search using multi-objective optimization

The assessment task realized in the FSA Demonstrator includes trimming, nonlinear step response simulation, linearization and criteria computations. The results of each assessment run are automatically visualized in the graphical output window. For experiment evaluation similar facilities are provided as described in tuning & compromising. The worst cases found in the assessment are stored on database. Thus they can be used to define new multi-models for further parameter tuning.

6. Software Architecture

Figure 8 shows the architecture of the FSA-Demonstrator. Graphical user interface and the computation engine ANDECS are realized as independent programs that communicate with each other via TCP/IP sockets. This enables the distributed execution of the software in a heterogeneous network.

The user interface is implemented in the Internet-programming language Java [2]. Since Java code is not

bound to any computer platform, it can be executed both on PC and on UNIX workstations. JAVA provides the programmer a powerful class library for building graphical user interfaces. The java.awt (abstract window toolkit) library contains all usual components like push buttons, radio buttons, lists, menubars, etc.. For process communication the *java.net* package is applied.

Visualization is done with the independent plot program AVIEWER. This plot module makes use of the MDI (multiple document interface) technology of Microsoft. It supports the simultaneous display of multiple diagrams in multiple windows. This feature is used for clear visualization of analysis results even in the multi-model case. AVIEWER also provides versatile facilities for interactive result evaluation. Both post-computational analysis and runtime-monitoring are supported.

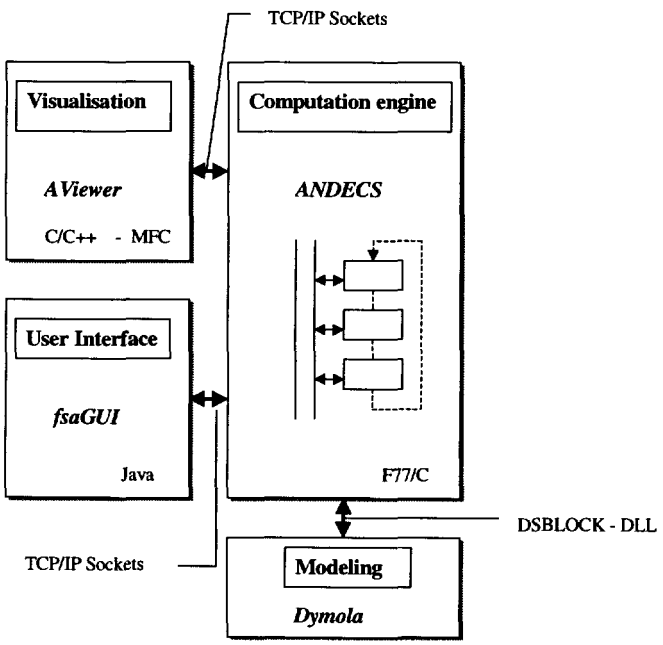

Figure 8: Overview of the FSA architecture

7. Summary and conclusions

A new framework for flight control development has been presented. This framework offers a state-of-the-art graphical computer interface for steering the design process. Advanced graphical user interface technology in combination with interactive result visualization supports the user during all design phases: *modeling, A/C-analysis, tuning & compromising and assessment.* Simultaneous visualization of information belonging together in different displays are very much assistant for the design process itself. Predefined database structures makes it easy to exchange data between the different design phases.

References

[1] H. Elmquist. Object Oriented Modeling and Automatic Formula Manipulation in Dymola. Scandinavian Simulation Society SIMS'93, Kongsberg, Norwegen, 1993.

[2] J. Gosling, B. Joy and G. Steele. The Java language specification. Sun Microsystems Inc., Addison Wesley, 1996.

[3] R. Finsterwalder. A Parallel Coordinate Editor as a Visual Decision Aid in a Multi-Objective Concurrent Control Engineering Environment. 5th IFAC/IMACS Symposium on Computer Aided Design in Control Systems, Swansea, UK, pp. 118-122, 1991.

[4] G. Grübel, H.-D. Joos, M. Otter, R. Finsterwalder. The ANDECS Design Environment for Control Engineering. 12th IFAC World Congress, Sydney, Australia, Vol. 6, pp. 447-454, 1993.

[5] G. Grübel. The ANDECS-CACE Framework A_RSYST for Integrated Analysis and Design of Controlled Systems. IEEE/IFAC Joint Symposium on Computer-Aided Control System Design, Tucson, Arizona, pp. 389-396, 1994.

[6] A. Inselberg. The plane with parallel coordinates. In The Visual Computer, 1985.

[7] H.-D. Joos. Multi-Objective Parameter Synthesis (MOPS). In J.-F. Magni, S. Benani and J. terlouw, Eds., Robust Flight Control: A Design Challenge. Lecture notes in control and information sciences, vol. 224, Springer Verlag, pp. 199-217, 1997.

[8] H.-D. Joos, A. Varga and R. Finsterwalder. Multi-Objective Design Assessment. IEEE Symposium on Computer-Aided Control System Design, Kohala, Hawai'i, USA, 1999.

[9] D. Moormann, P. J. Mostermann, G. Looye. Object Oriented Computational Model Building of Aircraft Flight Dynamics and Systems. Aerospace Science and Technology, 3(2), 1999.