Centrum voor Wiskunde en Informatica
REPORT RAPPORT

Aircraft Conceptual Design by Collaborative Manual and Automatic Agents

K.E. Shahroudi

Software Engineering (SEN)

SEN-R9702 February, 1997

Report SEN-R9702
ISSN 1386-369X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Aircraft Conceptual Design by Collaborative Manual and Automatic Agents

Kamran Eftekhari Shahroudi
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Email: kash@cwi.nl

ABSTRACT

In real applications, it is extremely difficult (if not impossible) to define function(s) that measure the "true merit" of a design object. For example, even the most prominent aircraft designers would not dare to claim that a particular set of merit and constraint functions measures the "true merit" of a class of aircraft.

The traditional approach to set up a "machine centric" optimization cannot effectively address this issue because there is no user interaction with the numerical optimization at design or run time. The semi-automatic concept, however, can help because the user is allowed to interact with the design problem and the design progress in-the-loop, such that the design criteria can be improved relatively easily at design time.

This paper briefly describes the semi-automatic design optimization setup which was introduced in full detail in a previous paper by the author. A simple multidisciplinary aircraft conceptual design optimization problem is then specified based on Torenbeek (1992). Various modes of in-the-loop user control on the search progress and the search problem then illustrate the potential benefits of allowing the user to interact with a numerical agent at various levels of automation.

1991 Mathematics Subject Classification: 65K10, 90C29, 90C31, 93B51.
1991 Computing Reviews Classification System: B.5.2, C.4., D.4.7, F.1.2., J.6.
Keywords and Phrases: Collaborative Manual & Automatic Agents, Conceptual or Preliminary Aircraft Design, Multidisciplinary Design Optimization, Human-in-the-Loop, Computational Steering, Semi-Automatic or Interactive Optimization.
Note: To be presented at the Interactive Computer Graphics session of the 1998 AIAA Aerospace Sciences Meeting, Reno, Nevada. Work carried out under project SEN 1.3 Interactive Visualization Environments.

1. INTRODUCTION

Design is generally accepted to be an iterative process, once a tentative design object is available. Total exclusion of the human from this iterative loop requires that the design problem be stated in a mathematical expression, i.e. an explicit specification of the design criteria such as design space, constraints and merit function. Standard text-books on aircraft conceptual design (e.g. Raymer (1989), Roskam (1985), Torenbeek (1982)) provide merit and constraint functions which can aid this process. However they fail to explicitly identify this as the "key" problem in optimization as in real applications, it is extremely difficult (if not impossible) to define function(s) that measure the "true merit" of a design object. For example, even the most prominent aircraft designers would not dare to claim that a particular function measures the "true merit" of a class of aircraft. They may have a set of functions that can aid the design process, but the trade-off between them that leads to the true optimum aircraft is always debatable and can change with time.

Imperfection in design criteria implies that the results of numerical optimization can also not be perfect requiring the human designer to improve these criteria by using intuition and experience. In most current setups, the human designer has to wait until the numerical results are available (perhaps hours or days later) before being allowed to manually modify the deign object and criteria. Boy et al (1990) present the more advanced Integrated Human Machine Systems (IHMS) view for design which assigns complementary roles for the human and artificial components. They suggest the distribution of the intelligent function among human and artificial agents which can compete or cooperate at design time.

Fig. (1) shows the automation spectrum for an IHMS type design system. Here the suggestion is made that the position in the automation spectrum or the value of *A* depends on whether the user can interact with the independent (or design) variables, constraints, merit function and search strategy inside the optimization loop. An ideal set up would allow the full range including the fully manual and fully automatic bounds. Most implementations operate at the right bound or allow the optimization to switch back and forth between the two bounds. Here we are mainly in the gray area in between which is rarely applied in design optimization.

In a previous paper by *Shahroudi (12, 1996)*, the author introduced the concept and implementation of a semi-automatic optimization setup which conforms to the IHMS and agents based (*Hale et al. (1994), Olsen et al. (1994)*) view of design, with a strong emphasis on human-in-the-loop, level of automation and very high interactive speeds for small scale problems, rather than the ontology whereby the agents communicate. The concept allows both cooperation and competition between the human designer and the numerical optimization agent. Semiautomatic control is provided by enabling the designer to modify design variables, simple bounds, constraint functions, the merit function and numerical control parameters (e.g. tuning parameters) via steering of graphical agents.

This paper briefly describes the semi-automatic design optimization setup. A simple multidisciplinary aircraft conceptual design optimization problem is then specified based on *Torenbeek (1992)*. Various modes of in-the-loop user control on the *search progress* and the *search problem* then illustrate the potential benefits of allowing the user to interact with a numerical agent at various automation levels.

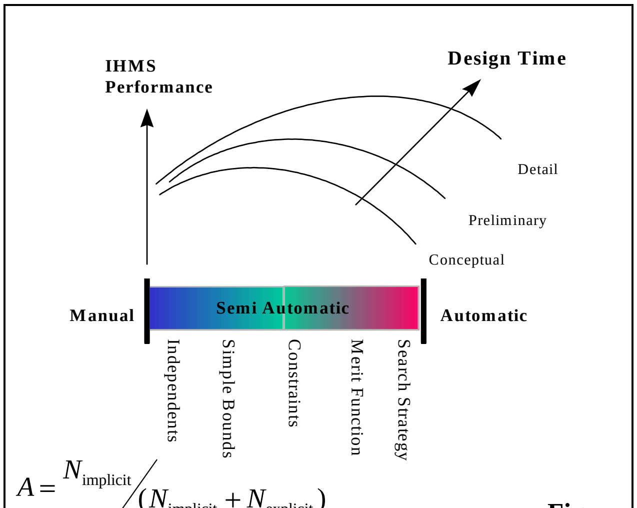

Figure 1: The Automation Spectrum.

The semi-automatic approach to design optimization represents a departure from the traditional "machine centric" view of design optimization which dominates the majority of aircraft conceptual or preliminary design packages today (*AAA (1994), ACSYNT(1992), ADAS (1988)*), where user interaction is typically limited to the problem definition phase or visualization of the results with little or no interaction at design or run time.

2. THE CONCEPT AND IMPLEMENTATION

2.1 The Semi-Automatic Concept

Only a brief description of the semiautomatic optimization concept and implementation is given below, because a detailed version has already been reported in *Shahroudi (12, 1996)*.

Regardless of the difficulty of defining merit functions and constraints, to qualify for a point in the semiautomatic spectrum we must have a mathematical expression of the design problem as shown below:

$$\begin{aligned} &\min_{X \in IR^n} f(X),\ X = x_1, x_2, \ldots, x_n \\ \text{subject to}\quad & g_i(X) = 0 \text{ for } i = 1,2,\ldots,m_1 \\ & g_i(X) \geq 0 \text{ for } i = m_1+1,\ldots,m \\ \text{where}\quad & f: IR^n \to IR \\ & g_i: IR^n \to IR \text{ for } i = 1,2,\ldots,m \end{aligned} \tag{1}$$

fully recognizing that this expression may be imperfect, requiring improvement at design time.

We also need to specify a numerical search strategy *S*, which defines the heuristics and the value of the tuning parameters used by the numerical solver. A practical way

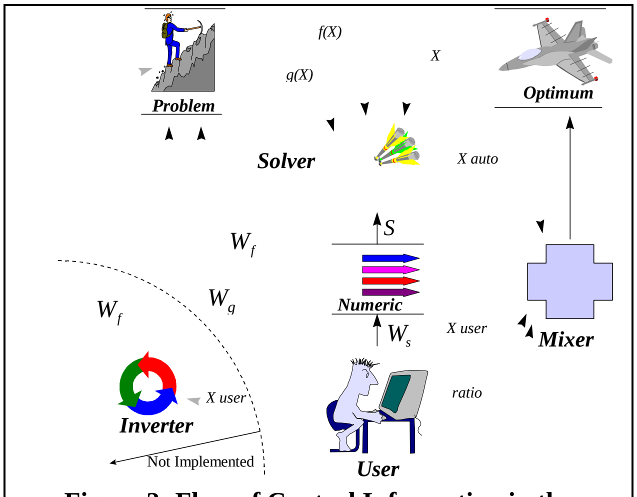

Figure 2: Flow of Control Information in the Semiautomatic Setup.

to define S is to specify a super strategy which consists of a selection of algorithms which run in parallel and output their numerical result. S then includes the values of the weight parameters that determine relative weight of each algorithm as well as their respective tuning parameters.

Interaction with X, $g(X)$, $f(X)$ and S, in that order, represents an increasing level of automation or increasing level of implicit control on design variables. For example suppose we are optimizing the wing weight fraction (merit $f(X)$) of an aircraft which is modeled in finite element from by say a million mesh points (design variables X) subject to a bound on allowable bending stress at wing root (constraint $g(X)$) using a genetic algorithm solver. As the optimizer proceeds the user can explicitly control the individual mesh points (low automation) to compete or collaborate with the numerical results. Controlling the allowable stress constraints can however affect a larger number of mesh points (increasing automation). Modifying the calculation of wing weight fraction(e.g. by adjusting the ultimate load factor) can affect a much larger number of mesh points (still higher automation). A higher level of control is still possible by controlling the S parameters that affect the diversification/intensification functions of the genetic search strategy.

Fig. (2) shows the flow of *control* information for the semiautomatic setup. The *Numerical Solver* agent continuously receives X, $g(X)$, $f(X)$ and S. Its job is to continuously output a better state for the design object X_{auto} .

The user interacts with the solver agent via the *Numeric*,

```
#include "ad_model.h"
#include "../opt3/define_optim.h"
#include "../opt3/op_run.h"
void main()
{
    extern void evaluate_model(), init_model();
    …
/* initialize model  */
    init_model();
/* define optimization problem */
   OP_bgn_prob_def();
     OP_add_model("Aircraft", evaluate_model);
     …
/* indeps */
     OP_add_indep(&M, "M", &M_min, &M_max);
     …
/* non-linear constraints */
     OP_add_con(&b, "b", &b_min, &b_max);
     …
/* merit */
      OP_bgn_merit_def();
         OP_merit_minimize("mu_p", &mu_p);
         …
      OP_end_merit_def();
    OP_end_prob_def();
/* run semiautomatic optimization */
    OP_run();
}
```

Figure 3: Sample Optimizer Code for Problem Definition (aircraft design example).

Problem and *Optimum* agents, whose responsibilities are:

- graphical representation of data to the user;
- receiving steering interaction from user to modify this data;
- broadcasting any modifications to the solver agent.

The right side of the figure shows *Progress Control.* If the user's hands are off the controls, the X_{auto} updates are flushed through to the *optimum* agent. In this way the user can monitor the progress by simply looking at the *optimum* agent. At any moment, the user may decide to interfere and provide new values of X_{user} by modifying the shape of the *optimum* agent. The mixer then mixes X_{user} and X_{auto} according to a mixture *ratio* which is also under human control. In this way the user can collaborate or compete with the numerical results depending on whether the requested change by the user is along or against the progress of the automatic results.

The left side of *Fig. (4)* shows *Problem Control.* Here the user can interact by directly modifying the shape of the *problem* agent and hence the value of the control parameters that define the merit and constraint functions, as the numerical optimization proceeds. In other words, the user can modify or steer a relatively large part of the design object by modifying the criteria that results in the current state of the optimum via the solver. This represents a higher level of automation than the previous section because a larger number of design variables are controlled implicitly (by specifying a few things), instead of explicitly (by specifying a lot of things) .

The *Inverter* agent allows the user to modify the state of the optimum and observe the result in the design criteria. This is the parametric equivalent of the work of *Gruber (1991)* which attempts to acquire the design rationale underlying a particular design, except that here the inversion is achieved by continuous in-the-loop user interaction. This part is not yet formally implemented, but the tools required are already available, e.g. the high speed parametric inverter reported by *Shahroudi (6, 1996).* An example inverse design type exercise is to find out how two aircraft from different manufacturers compare in terms of the tradeoffs used between performance, cost and reliability.

Including the human in the optimization loop, allows us to draw from the best of the two worlds of manual and automatic optimization. The benefits are briefly:

- It alleviates some pressure on the precise statement of the design optimization problem, since these criteria can be varied and their consequences observed at design time;
- It is particularly suitable for multidisciplinary or multiple objective design optimization, where it is interesting to study the tradeoff between various discipline at run time;
- It aids the reconstruction of the design rationale underlying a particular design or for comparison between various designs;
- User intuition and experience is used at run time. The location and direction of search can be continuously

guided by the user in collaboration or competition with the numerical results;
- Freedom of the user to influence the search progress at run time, tends to globalize the optimization, although the numerical algorithm may be a local one.

2.2 The Implementation

To take advantage of the current implementation the user starts with a simulation. The optimization problem is then specified in terms of this simulation by using a purpose made library of functions (*Fig. (3)*), which typically takes a few minutes to complete. Thereafter, a standard graphic user interface is automatically generated which includes the *Problem*, *Optimum* and *Numeric* agents (*Fig. (4)*). The standard interface is then used in collaboration with a user defined interface (*Fig. (5)*) to provide the various modes of control discussed above.

In the standard interface (*Fig. (4)*), the *Bounds* box is part of the *Problem* agent which controls the constraint variables via sliders. The *Merit* box on the left is also a part of the *Problem* agent. For the aircraft design example, three minimization criteria were specified in the optimizer code. Here a slider is assigned to the weight of each criterion such that shifting them sideways affects the trade-off between them. The *Design Variables* box is a group of sliders which form the *Optimum* agent.

The user defined interface (*Fig. (5)*) provides the opportunity to include the *Numeric*, *Problem* and *Optimum* agents in a more application specific visualization. Here the *optimum* agent is the shape of the aircraft. The user defined interface is also very useful for including constraints (e.g. minimum wing sweep angle) and all the parameters of interest which have a side effect but which are missing from the specification of the optimization problem. For example the range of the aircraft was not included in the optimizer code but several constraints and merit function depend on its value. The example interface allows the user to modify the range by pulling on the arrow (lower left of figure), which in turn results in a new optimum shape of aircraft. In the figure, the green, red and orange sections are mapped to payload weight, fuel tank volume and engine thrust at take-off respectively.

Currently the concept is implemented in the Computational Steering Environment (*CSE*) *(Wijk et al. (1994)*) which allows the numerical algorithms, graphical interfaces, simulations and user to collaborate in a distributed fashion around a central data manager.

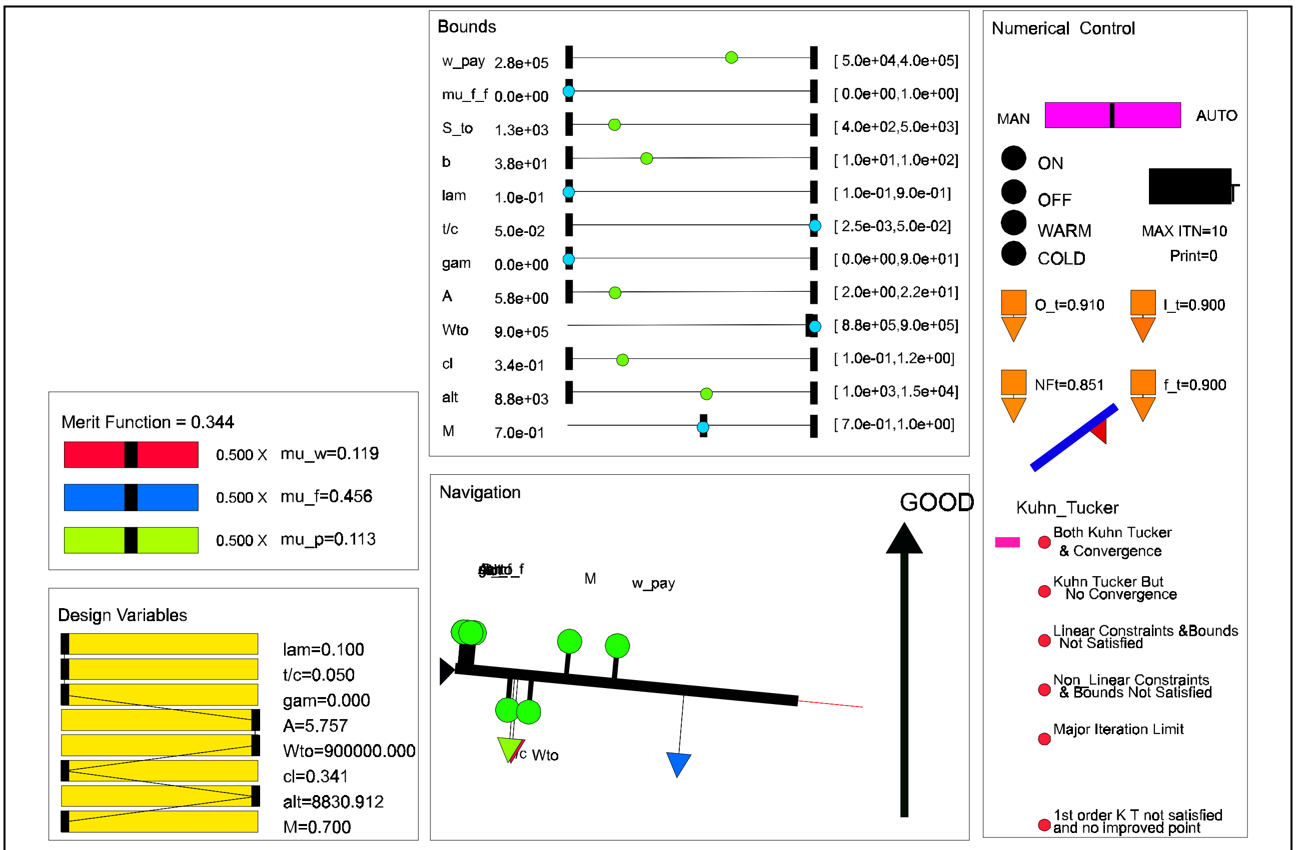

Figure 4: Standard User Interface Generated Automatically for the Aircraft Design Example.

2.3 Significance to Preliminary Design

The ability to modify the search problem and immediately observe the results in the progress of the optimum provides a flexible means to handle the always existent difficulty of defining a good set of constraints and merit functions for design. This is of crucial significance to the *Conceptual* and *Preliminary* phases of design for the following reasons:

- The human designer's knowledge about the design problem is relatively low which requires a lot of flexibility for modifying the design criteria until a point is reached when these criteria can become relatively fixed;
- The level of design detail is such that current computational and graphical resources are capable of computing and displaying the consequences of human interaction quickly. This allows the design activity to approach the *Natural Design Cycle* ideal (*Shahroudi (1994)*) which takes advantage of the superior short term capabilities of the human visualization pipeline;
- Nature of optimization is typically multidisciplinary or multiple objective in early design phases but the exact tradeoff between the various disciplines is not necessarily fixed, requiring flexibility to modify the relative importance of each discipline which fits in nicely with the semiautomatic concept.

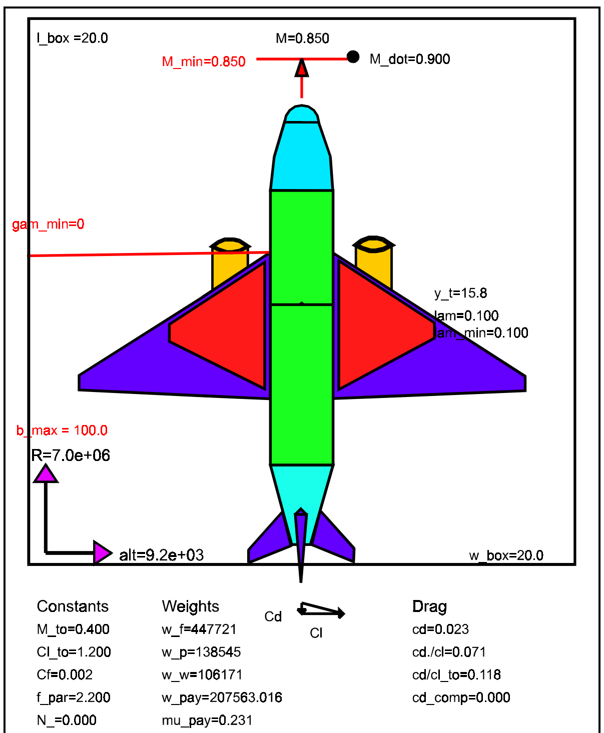

Figure 5: User Defined Interface for the Aircraft Design Example.

3. AIRCRAFT DESIGN EXAMPLE

3.1 The Design Problem

This is a multidisciplinary conceptual design exercise which is in essence similar to that presented by *Johnson (1988)*. A brief specification of the design problem is:

Independent or Design Variables: Flight Mach Number M, Altitude H, Lift Coefficient C_L, Take-Off Weight W_{to}, Aspect Ratio A, Wing Sweep Angle Λ, Wing Thickness to Chord Ratio t/c and Wing Taper Ratio λ.

Simple Bounds: on all the design variables above.

Constraints: Wing Span b, Wing Root Chord, Take-Off Field Length S_{to}, Fraction of Fuel Stored in Wings, Payload Weight W_{pay} and Take-Off Thrust T_{to}.

Composite Merit Function: minimize Fuel Weight Fraction μ_f, Propulsion Weight Fraction μ_p and Wing Weight Fraction μ_w with respect to Take-Off Weight.

Parameters with Side Effect: long list including Range R, Technology Factors, Parasitic Drag Area, Take-Off Lift Coefficient etc.

For full derivation of the three separate elements of the merit function above see *Torenbeek (1992)*. However a brief description is given in the next section for clarity.

3.2 The Simulation

The independent variables of the optimization, M, H, C_l, W_{to}, A, Λ, t/c and λ together with the side-effect parameters are the input parameters to the simulation.

The Propulsion Weight Fraction μ_p is given by:

$$\mu_p = \frac{W_{engines}}{W_{to}} = \frac{\mu_t / \tau}{\delta} \frac{C_D}{C_L}$$

where the engine installed thrust to weight ratio μ_t is a side-effect parameter and can vary between 0.3-0.35 for high bypass ratio turbofans. The corrected thrust lapse

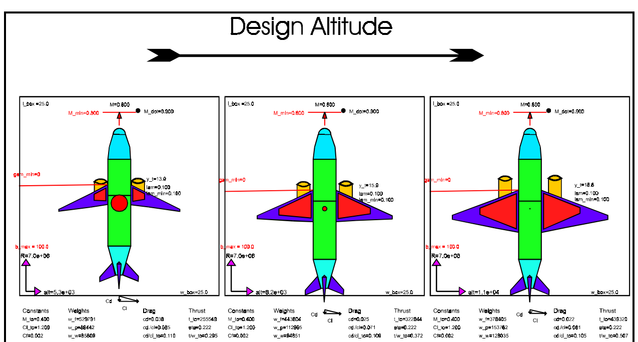

Figure 6: Effect of Design Altitude.

and specific fuel consumption model for high bypass turbofan engines is due to *Mattingly et al. (1992)*:

$$\tau = \frac{T}{\delta\, T_{SL}} = 1 - 0.49\sqrt{M}$$

$$SFC = (0.5 + 0.6M)\sqrt{\theta}$$

where the relative atmospheric pressure and temperature, δ and θ, are calculated by a standard atmosphere model. The drag coefficient C_D comes from the modified Korn's simple sweep theory including compressibility and parasitic drag effects.

The Fuel Weight Fraction μ_f is given by:

$$\mu_f = \frac{R / R_H}{\eta} \frac{C_D}{C_L}$$

where R_H is the ratio of fuel heating value to gravitational acceleration and is roughly equal to 4400 Km. The installed overall efficiency of the engines η can be easily calculated from the *SFC* and aircraft speed.

The Wing Weight Fraction μ_w is calculated by:

$$\mu_W = \phi_A A \sqrt{\frac{A}{\delta\, C_L}} + \frac{\phi_S}{\delta\, C_L}$$

where ϕ_A and ϕ_S are weight factors related wing root stress and smeared thickness of wing skin.

3.3 Run Time Interactions

The reader should note that the run time interactions described in this section are high speed (or real time) human-in-the-loop interactions, where the term "loop" refers to the design optimization loop. Elsewhere in the literature, the term "human-in-the-loop" is frequently loosely used to describe interactive systems without a clear definition of which loop it concern and where the human is interacting. Here, we have a high speed design optimization loop which continuously updates a better state for the optimum. Since user interaction is allowed inside this loop, then all the consequences of user interactions are also immediately available for observation. This has the effect of giving a natural feel to the design activity and is discussed in detail by *Shahroudi (1994)*.

With the numerical optimization switched off, the user can manually search for a better state by steering the *Optimum Agent*, (e.g. by manually dragging the Mach slider in the standard interface (Fig. (4)) or the red arrow in the user defined interface), and continuously observing the consequences in the shape of the aircraft, the constraints and the merit function. In practice however, this is a frustrating exercise because one or more constraints can easily become violated. Attempting to manually fix these violations can result in the violations of some other constraints and so on.

With the numerical optimization in run mode, *Fig.(6)* shows that increasing the design altitude, increases the size of the wings and the engines. The resulting larger wing volume allows an increasing proportion of fuel to be stored in the wings.

Fig.(7) shows the result of interaction with a simple bound, namely the lower constraint on the flight Mach number. Here, increasing the flight Mach number results in a more sporty look for the optimum aircraft, i.e. the wings sweep back , bigger engines etc.

In *Fig (8) and Fig. (9)*, the user interacts with constraints, at a higher automation level in order to learn the effect of take-off constraints on the optimum aircraft. Enforcing a short constraint on runway length, forces the wings to become larger and reduces the payload weight. Smaller engines can carry less payload and make the wings slender.

Fig. (10) shows interaction with the constraint on the ratio of fuel that can be stored in wings.

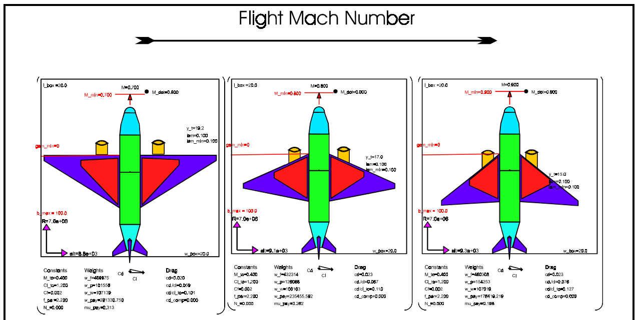

Figure 7: Effect of Lower Constraint on Flight Mach Number (Simple Bound).

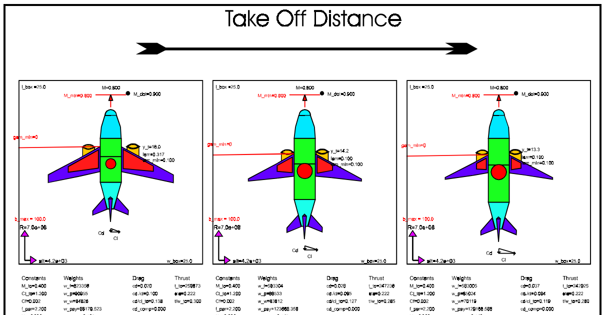

Figure 8: Effect of Maximum Take-off Distance.

The above figures show the state of the optimum if the three minimization criteria that form the merit function are equally significant. But the correct trade-off depends on many factors such as fuel and engine prices. *Fig. (11)* shows user interaction at a still higher level of automation to answer questions related to the trade-off between the wing, fuel and engine weight fractions. Here the user can vary the trade-off on the fly and immediately observe the change in the shape of the aircraft.

Finally, *Fig. (12)* gives an example of interacting with the range which is a parameter with side effect. Like other side-effect parameters, *R* is not mentioned in the specification of the optimization problem but its value affects a number of constraints and the Fuel Weight Fraction which is an element of the merit function. Increasing *R* reduces the payload weight and increases the aspect ratio of the optimum aircraft . A smaller wing volume, in turn, reduces the size of the fuel tank inside the wings, so that a larger proportion of fuel has to be carried in internal fuel tanks as shown.

The above conclusions are not fixed and depend on the simulation, the current position in the search space, the tradeoff between merit elements, new additional constraints, and so on. The current implementation enables easy redefinition of a design problem in terms of a new or modified simulation, in order to reach more accurate conclusions or to adapt existing ones to new situations. Once defined, the user can directly compete or collaborate with the numerical optimization results or interact at higher levels of automation with constraints, tradeoffs and side-effect parameters. Lacking such a flexible design tool, has forced designers to derive and work with a large number of local sub-conclusions, while keeping the number of changes to the design problem (e.g. tradeoffs, constraints etc.) to a minimum. For example, *Torenbeek (1992)* derives an expression for the partial optimum lift coefficient for a given altitude and aspect ratio by differentiation of an expression for drag to lift ratio, C_D/C_L. This ratio is a significant component of μ_p and μ_f but does not appear in the expression for μ_w. Furthermore, C_D/C_L is not the only significant element that appears in these expressions and the tradeoff between μ_w, μ_p and μ_f is not necessarily fixed. While these sub-conclusions are instructive and useful for analytical work, they offer little to the overall design activity. An integrated human machine design optimization tool such as the one described in this paper allows a more declarative style, whereby we can move towards a more global set of conclusions about the design object.

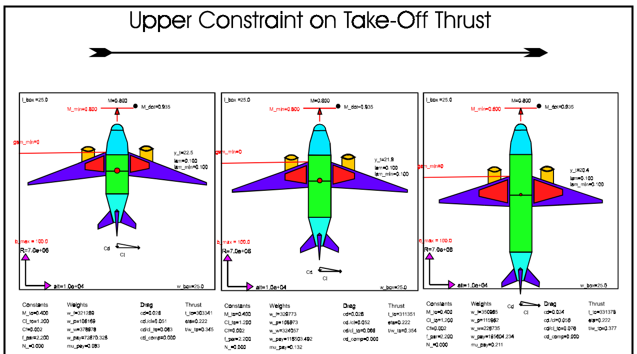

Figure 9: Effect of Upper Bound on Take-Off Thrust.

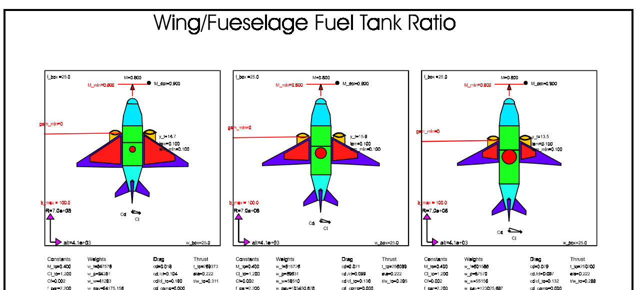

Figure 10: Effect of Wing/Fuselage Fuel Tank Ratio.

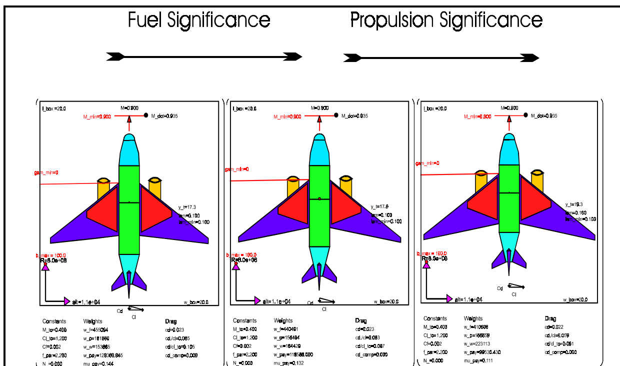

Figure 11: Effect of Modifying the Merit Function.

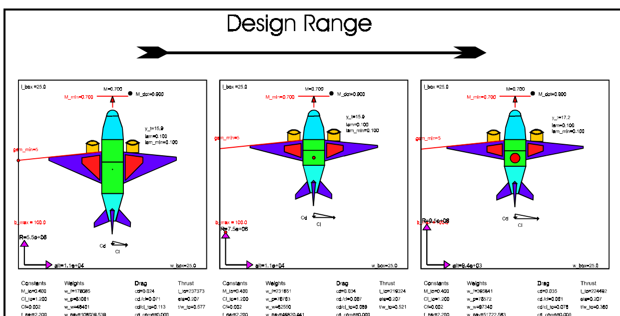

Figure 12: Effect of Range on Optimum Aircraft (A Side-Effect Parameter).

5. CONCLUSIONS

This paper briefly describes a semi-automatic approach which allows user interaction with the design optimization loop at various levels of automation. The current implementation allows user interaction by steering the *Problem*, *Numeric* and *Optimum* agents in order to control the *search progress* and the *search problem*. This approach conforms to the IHMS view of design and agent based integrated product design methods, with strong emphasis on human-in-the-loop, automation level and very high interactive speed for small scale problems, rather than the ontology by which agents interact.

Majority of aircraft conceptual or preliminary design packages today, conform to the "machine centric" view of design optimization. Standard textbooks in aircraft design neglect to emphasize the "key" problem inherent to design optimization in general, because until recently, design environments, processing speeds and visualization technology were not sufficiently advanced to solve it. The "key" problem is to derive a good set of design criteria (i.e. the specification of a design space, constraints and merit function) which can measure the "true merit" of a design concept.

A multiple objective aircraft design example illustrates the potential of the semi-automatic approach for aircraft conceptual or preliminary design optimization. The paper shows how the user can improve the design criteria and reach complex conclusions about the design by observing, in real time, the consequences of steering the design variables, simple bounds, constraints, tradeoffs and parameters with side effect.

ACKNOWLEDGEMENTS

This work is supported by the Netherlands Computer Science Research Foundation (SION), with financial support of the Netherlands Organization for Scientific Research (NWO).

REFERENCES

AAA, Annemaat, W. (1994), *Advanced Aircraft Analysis AAA Software User Manual*, DARCorp., Kansas.

ACSYNT Institute (1992), *ACSYNT Overview and Installation Manual*.

ADAS, Bill, C. (1988), *Development and Application of a Computer Based Design System for Conceptual Aircraft Design (ADAS)*. Delft University Press, ISBN 90-6275-484-8.

Boy G. et al. (1990). *Intelligent Assistant Systems: Support for Integrated Human-Machine Systems*, Proceedings of 1990 AAAI Spring Symposium on Knowledge Based Human-Computer Communication, Stanford University.

Gruber, T. R. (1991), *Interactive Acquisition of Justifications: Learning "Why" by Being told "What"*, To appear in IEEE Expert, based on a lecture presented at the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, Japan, 1990.

Hale, M. A. et al. (1994), *Preliminary Development of Agent Technologies For a Design Integration Framework*, AIAA-94-4297, 5th Symposium on Multi-disciplinary Analysis and Optimization, FL, Sept. 7-9.

Johnson, V. S. (1988), *Optimizing Conceptual Aircraft Designs for Minimum Life Cycle Cost*, Proceedings of the second NASA/Air Force Symposium on Recent Experiences in Multidisciplinary analysis and optimization, Hampton, VA.

Mattingly et al. (1992). *Improvements in Teaching Aircraft Engine Design*, AIAA-92-3758, 28th AIAA/ASME/SAE Joint Propulsion Conference.

Olsen, G. R. et al. (1994). *Collaborative Engineering based on Knowledge Sharing Agreements*, Proceedings of the 1994ASME Database Symposium, Minneapolis, MN.

Raymer, D. P., Aircraft Design: A Conceptual Approach, 1989, AIAA Education Series.

Roskam, J. (1985), *Airplane Design, Pt. 1, Preliminary Sizing of Airplanes*, Roskam Aviation & Engineering.

Shahroudi, K.E. (1994). *Development and Validation of a Computer Assisted Design Methodology for Gas-Turbine-Based Aircraft Engines*. Delft University Press, ISBN 90-407-1070-8.

Shahroudi, K.E. (6, 1996). *The Flipping Analytical Coin Tool: Closing The Information Flow Loop in High Speed Analysis*. Proceedings of the 2nd International Conference on Inverse Methods, Nantes, France.

Shahroudi, K.E. (12, 1996). *Design by Continuous Collaboration Between Manual and Automatic Optimization*. To be published in the Journal of CMSE, Sage press.

Torenbeek, E. (1982). *Synthesis of Subsonic Airplane Design*. Report LR-775, Delft University of Technology.

Torenbeek, E. (1992). *Optimum Wing Area, Aspect ratio and Cruise Altitude for Long Range Transport Aircraft.* Report LR-775, Delft University of Technology.