TOOLS AND METHODOLOGY FOR COLLABORATIVE SYSTEMS DESIGN APPLIED ON MORE ELECTRIC AIRCRAFT

Björn Johansson*, Lars Austrin, Göran Engdahl***, Petter Krus***
***Linköping University, **Saab Aerosystems, ***Royal Institute of Technology**

Keywords: *Aircraft systems design, collaborative methods, computational design*

Abstract

This paper illustrates a collaborative approach for distributed model development and simulation applied to aircraft systems design in the so-called "More Electric Aircraft" (MEA) study. A computational framework and a distributed model were created as a joint effort between Linköping University, Saab Aerosystems, and the Royal Institute of Technology in Sweden. Parts of the model were developed at different locations and integrated to a complete system. The principle for the models, the proposed methodology and the developed computational framework are described in the paper.

1 Background and Motivation

Aircraft design in general is a complex engineering task involving several working teams from different domains. The design cycle is defined by different stages starting with conceptual and preliminary design. In order to minimize risk and late changes in the design, integration of multidisciplinary analysis and hence collaborative methods are necessary from outset.

A difficulty is the integration of the required simulation tools, interfaces between models and handling of the information flow between them. A typical concern is also often that the design teams and their tools are distributed geographically and that models contain proprietary information.

Beside the technical difficulties associated with the integration of the models, several other aspects must be addressed in order for the simulation results to be useful. First of all, computational models must be developed for an agreed upon purpose, i.e. each sub-model must deliver characteristics that are requested on a system level. This means that the models must be developed at a sufficient level of detail and capture the properties of interest to evaluate the models against constraints and requirements. At the same time, the models must enable simulation for a large number of iterations and can therefore not be too complex.

The tools and methods presented in this paper show how modeling and simulation involving several participants can be applied at an early stage in the design process. Models are developed by domain experts and integrated in a distributed framework giving all project members access to models and simulation results through standard desktop tools such as Microsoft Excel.

2 Managing Models in Collaborative Design Projects

Looking at the simulation model from an aircraft design perspective the models can provide important information to the design process. The models are perhaps even more important in collaborative projects. Multi-disciplinary modeling and simulation is here necessary to overview and evaluate complete systems. We can also see a trend that system integrators tend to require simulation models to be delivered from the suppliers. This is a tough requirement since it means that different partnerships require validated and updated models for a wide range of different simulation environments. Apart from the fact that this is time-consuming and requires education in the different tools, it also implies heavy costs for software licenses.

Another aspect that becomes very clear in collaborative system design is proprietary issues. As Fig. 1 illustrates, collaboration often also means competition. A supplier collaborates with the system integrator at the same time as competing with other participants in the project. Since the model contains important information about the product, it is necessary to be able to manage proprietary information and integrate subsystem models in the total system model at the same time.

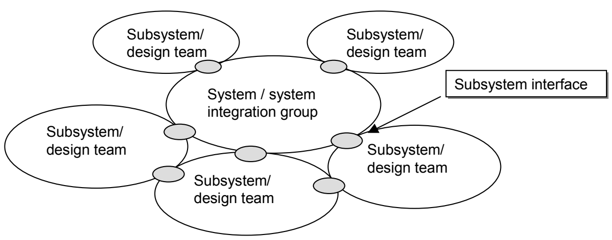

Fig. 1. System design and integration

2.1 Approaches for Model Integration

In order to enable computational collaboration, methods to handle distribution must be introduced. There are two general approaches that apply to all types of distributed computing: The interface-centric and the data-centric approach are defined by Morgenthal [10] as:

- Interface-centric approach. Interface centric systems communicate over agreed-upon protocols. These protocols have typed input and output parameters and typed return values.
- Data-centric approach. Data-centric systems communicate through an agreement of data format.

With the above definitions it is understood that there are two major ways to work with modeling and simulation of multi-domain systems in collaborative environment as illustrated in Fig. 2.

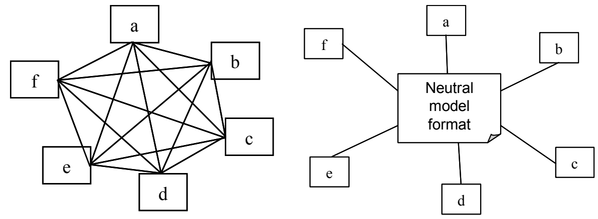

Fig. 2 Interface-centric and data-centric integration is the two fundamental approaches for integration of models and tools.

The interface-centric approach implies that models developed in the specific domains are run in the domain-specific tool. To realize the system behavior, the tools are connected and run in parallel exchanging data throughout the simulation. This approach is attractive when models exist in different domains that have required a great deal of effort to create. Furthermore, this simulation approach is necessary in cases where very advanced models are to be simulated that require domain-specific tools with special functionality. However, this type of approach is difficult to implement and in the case of tight connections between the subsystems, numerical problems must also be handled.

The data-centric approach requires agreement on model format. One promising approach is the *Modelica language*, which is an object-oriented and equation based modeling language for physical systems [9]. The drawback is however, that so far is the Modelica language implemented in more than a couple of tools.

Since no neutral model formats exist that are standard and implemented in a wide range of simulation tools, this means in practice that all partners must run the same simulation tool. The drawback with this approach is that models containing proprietary information cannot be protected.

2.2 Existing Solutions for Model Integration

Commercial packages have been developed for the purpose of model integration and distributed computing, for example ModelCenter by Phoenix Integration [14] and iSight by Engineous Software [3]. Another interesting project is the Federated Intelligent Product EnviRonment (FIPER), [16], which is a collaborative effort involving universities and several large engineering companies with the goal of developing a commercial product development framework. These packages are advanced design environments that contain extensive functionality such as model integration, design optimization, design of experiment etc. A few interesting university frameworks are also being developed, as for example DOME from Massachusetts Institute of Technology [14], X-DPR from Georgia Institute of Technology [2], and i-EDA from Michigan State University [12].

2.3 Requirements for a Model Management Framework

It is required that a model management framework supports circumstances such as collaboration, competition and distribution at the same time. This means that even though an open neutral format for models is desired, it is not always a feasible solution. It must be possible not only to exchange models between groups, but also to integrate models without revealing proprietary information. This requirement also applies to the fact that the large amounts of existing legacy model code cannot be transformed and exchanged and must thus be integrated in their original state.

2.4 Web Service Technology

Along with the existing frameworks presented above, general standards for distributed computing are emerging. These standards are developed within the frame of the extensible markup language (XML) [18], and are referred to as Web Service standards. Web services are programming language-, programming model-, and system software-neutral. Tsalgatidou and Piloura [16] describe the Web Service model as three integrated parties in the general case:

- Service provider
- Service requester
- Service directory or broker

In, Fig. 3., the interaction between the service provider and service requester is illustrated. The two XML-based protocols SOAP (Simple Object Access Protocol) and WSDL (Web Service Description Language) are the key building blocks of the web service technology as illustrated in Fig. 3, see also [18]. The service provider hosting the service authors (automatically generates) the WSDL document that fully describes the service with its operations, incoming and outgoing variables, data types etc. The information in the WSDL document is enough for automatic generation of a skeleton of the client implementing the connection through SOAP. Web services can then be accessed using ubiquitous transport protocols such as HTTP.

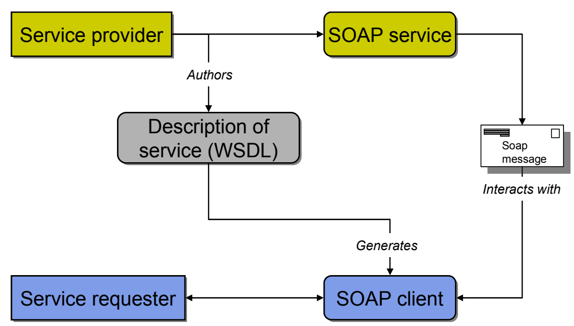

Fig. 3. Fundamental principles of web service technology.

3 A New Framework for Collaborative Modeling and Simulation

A new framework for integrating distributed computational resources has been developed in the course of the work presented here. The framework is based on the Web Service standards for service description and messaging described above. The idea is to enable not only encapsulation of existing model codes but also the development and integration of new models. The framework, as illustrated in Fig. 4, could from the user perspective be represented in three different levels. On the lowest level are the actual models, or simulation modules. These are implemented as web services and could contain an arbitrary standalone code representing either a model or a computational design method. The level above is the so-called *Integration Service*, also implemented using web service standards. This service includes functionality to create mapping between the other modules inputs and outputs. It also includes a repository for storing the design data. The highest level is the user interface, also called the client, which can ultimately be any of the tools that the engineer uses to manage and interact with the design information.

This framework differs from existing ones due to the integrated view of both models, methods, data and user clients and the possibility to deploy both models and computational methods using *open standards*. This creates considerably high flexibility.

The principal architecture is illustrated in Fig. 4. Below, the framework is described further on a rather brief level. More detailed description can be found in [6] and [7].

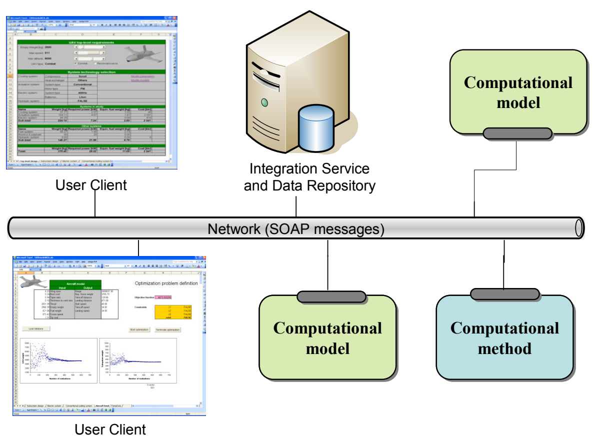

Fig. 4. In the project, a service-oriented architecture for integration of computational models and methods

The principle is to deploy computational models and methods as generic modules with the following characteristics:

- A module can contain either a computational model or a method (i.e. an model operator or and specified design algorithm where models are called in a specific sequence)
- Modules can be combined and integrated to a system model and be simultaneously accessed by different client tools.
- Modules can be instantiated with different sets of parameter values
- Modules can be accessed either directly or through the model integration service.

Important to note is that the framework is intended for integration of models from a design perspective where interaction only is necessary at a comparably *low frequency*. This means that it is not well suited for dynamic, tightly coupled models where interaction with short time-step is required. Due to the fact that data transfer between modules is comparably slow, tightly coupled models are preferably located within a module.

An interesting feature with the framework presented in this paper is the ability to use a standard spreadsheet tool such as MS Excel as the user interface. This is an attractive approach since most engineers already have such software installed on their computers. Furthermore, Excel is also a powerful engineering tool well suited for management of input and output data to and from the simulation models.

Since all the necessary controls for accessing the web service modules are embedded in the Excel document as Visual Basic macros, it is possible to distribute the spreadsheet among user who can run it without any additional software.

In the sections below, More Electric Aircraft project will be described where the presented framework has been used.

4 The More Electric Aircraft Study

The More Electric Aircraft Project (MEA) is a study with focus on selecting and integrating electric systems to a larger extent in future aircraft design. The specific aircraft type for this project is an Unmanned Aerial Vehicle (UAV). The size and type of the UAV is unspecified so far.

The rationales for a MEA-design are to gain performance, to get lower weight and reduced maintenance. The potential for improvement is considerable since about 30% of the weight and maintenance are related to systems where MEA technology can be applied. A possible goal for weight savings and reduction of maintenance is about 5% for the applicable systems as well as in the complete aircraft considering the carousel factor.

The MEA project is a typical multi-disciplinary project, which requires knowledge from several engineering domains to be integrated. A simplified description of the project task is to replace all hydraulic systems and all pneumatic links and have the replacement electrically supplied. Power by wire is the key word.

The impact of one system on another system is considerable and this is why the analysis must be performed at aircraft level. Primarily the electric power system, the flight control system and the environmental control system are affected, however the severe impact is on the hydraulic system that can be fully replaced and the auxiliary system will be affected.

In Fig. 5, an important part of the task is illustrated. This concerns selection of systems and components based on different technologies. In order to analyze the selection of different concepts based on computational models, it is important that the systems models are modular and interfaces between the models correspond to the physical interfaces.

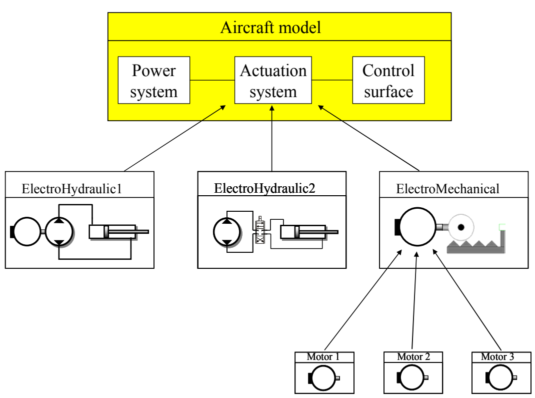

Fig. 5. An illustrative example of concept selection of systems in the MEA project

5 Methodology for Model Development in the MEA Project

The purpose of the modeling effort is to evaluate the advantages achieved with new technology, such as lower weight and reduced cost. It is of vital importance during an aircraft design concept phase to get preliminary data from the models.

The challenge is to evaluate not only the advantages with new technology but the total effect on an aircraft design i.e. it is acceptable with an increased weight on specific components as long as it generates weight savings elsewhere. By characterizing the systems and the vital components by algorithms allowing different power rating and different technology, we are creating the input for the simulation.

The algorithms are based on component data, information from vendors and product catalogs. The algorithms are giving the weight and cost and are based on weight of the vehicle as well as the type of mission.

What can be achieved must be evaluated at aircraft level. This is why models for different systems or sub-systems must be accessed and analyzed in the model.

The important input to the different system models is the aircraft weight. Based on estimations of aircraft weight, max speed and altitude, the required power to the systems are estimated. It is important to be able to modify these parameters on a system level at an early stage and evaluate the effect on the subsystem weight and cost.

6 Analysis Setup for the MEA Project

In Fig. 6, the simulation setup for the MEA project is illustrated. The setup is an implementation and configuration of the framework described in section 3. The different pars of the system are described below.

6.1 Computational Server

In this specific project, the modules do not contain information that needs to be protected between the groups. Therefore, a solution with one computational server has been chosen and setup at Linköping University in order to handle the models and perform the computations. The implementation however includes all technical mechanisms that would be required if the models would be deployed at distributed computers.

6.2 Implemented and Deployed Models

The simulation models of the sub-systems are developed in the Matlab package. This is accomplished by publishing Matlab functions and M-files as web services which then are accessed remotely and integrated with other models and data from other resources. The models that are implemented and deployed for the MEA system study are the following, see also Fig. 6:

- *Cooling system.* Includes compressor, heat exchanger, etc. The weight, cost and required electrical power is computed from required cooling power and technology selection.
- *Actuation system*. An electro-mechanical or electro-hydraulic actuation system including electric motor and mechanical or hydraulic system. The module computes weight, cost and required electrical power from required performance.
- *Fuel system*. Includes fuel tank, electric motor, fuel pump and piping. Computes weight and required electrical power.
- *Electric power system.* Includes batteries, generators, wiring etc. This model is described in more detail below.

6.2.1 Electric Power System Model

The electric power system model is here described in more detail to give a brief overview of the models at all levels and also to illustrate that the sub-systems can have large impact on the weight of the complete aircraft.

The model includes algorithms for how cost and weight of the system depends on different parameters defining the design of the system. For example it can be noted that double frequency gives the potential for the half weight where magnetic components are used as in transformers and generators. Double voltage gives the potential for half weight of the power wiring. The size and weight of the vehicle also have impact on the weight of the wiring.

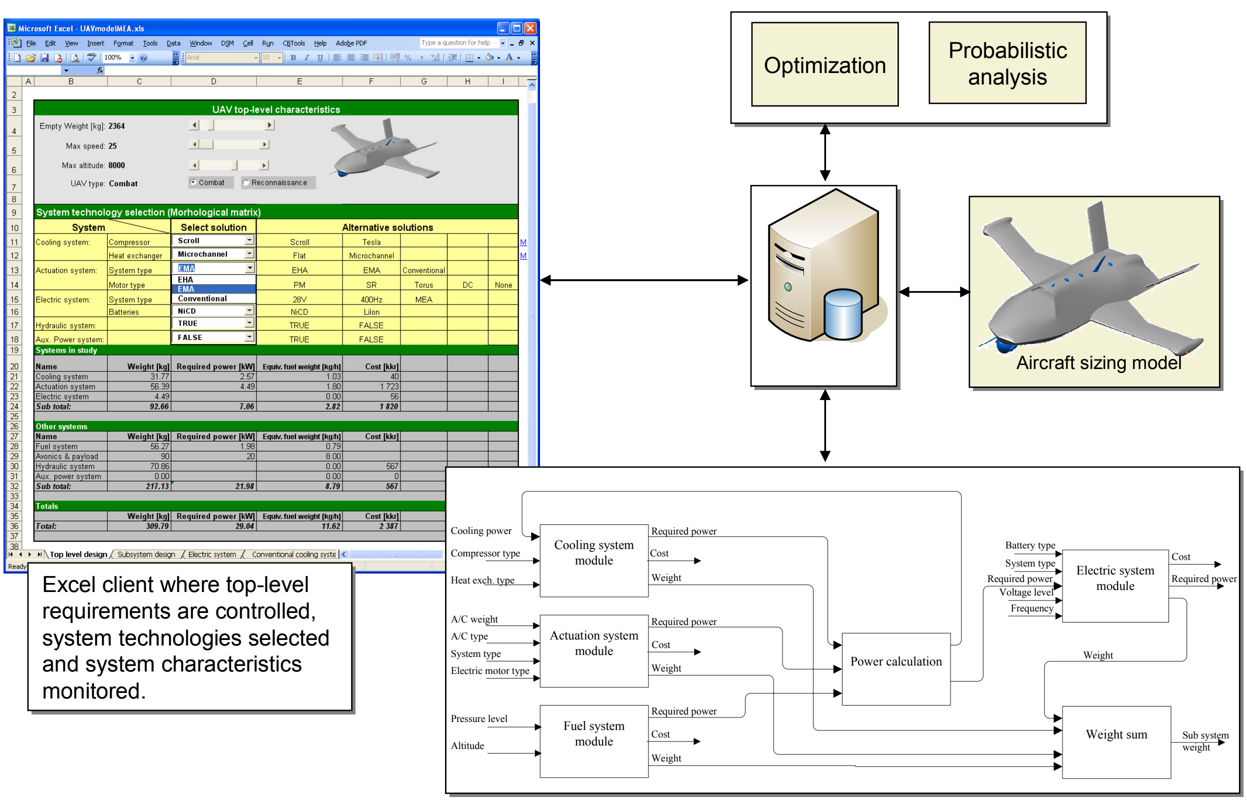

Fig. 6. Simulation setup for the MEA project.

Variables in the model that have impact on the weight and cost output are:

- The chosen architecture: one or two engine configuration.
- Voltage level [V].
- Frequency level [Hz].
- Fixed frequency or variable frequency
- AC or DC power.
- Power rating and payload [kW].
- Vehicle empty weight [kg].
- Technology chosen e.g. NiCd or Li-Ion Batteries

Further, if the MEA technology is chosen, the hydraulic system might be excluded, as well as major components in the auxiliary power supply system. The environmental control system will be affected as well. The consequence of this is included in the models but is not further presented in this paper.

6.2.2 *Aircraft Sizing Model*

An *aircraft sizing model* has also been integrated with the sub-system models. This is a model with rather simple equations for geometry, weight, engine, performance, and aerodynamics. This model is implemented as a Fortran subroutine illustrating that a model deployed as a web-service can easily be integrated even if the actual model is implemented in different programming languages.

6.3 Probabilistic Analysis and System Optimization

A computational module for probabilistic analysis has also been implemented in the presented framework. This enables simulation of uncertainty in the input parameters to the simulation model and is accomplished by a Monte-Carlo algorithm that simulates uncertainty according to a probability density function (PDF). By repeatedly calling this module, statistical results can be obtained where for example the probability of meeting a requirement with a particular concept can be estimated. For more details about the technical implementation and examples, see [5].

The optimization algorithm is the Complex-RF optimization method which is a modified version of the Complex method by Box (1965) [1]. This method is a non-gradient method and has been used very successfully over a wide range of problems and is characterized of simplicity and robustness. Therefore it is suitable as a general purpose method. The Complex-RF method is described in [8].

Both the algorithms for probabilistic analysis and design optimization have also been implemented as computational methods in the framework presented here.

6.4 Implemented Methods for Concept Analysis and Selection

One important task has been to create powerful tools and methods for control and overview at the system level of the design. This means that it should be possible to overview all important system characteristics and to enable control of simulations, optimizations, and probabilistic analysis. In Fig. 6, the user interface implemented in Microsoft Excel is visualized.

6.4.1 *Controls for Top-level Characteristics*

It is desired that the presented framework can be used for design and decision making on several levels. On the highest aircraft level, it is therefore of interest to modify main aircraft characteristics to evaluate how these will affect the subsystems. Simple controls for empty weight, max speed, max altitude, and type of UAV have therefore been implemented in the Excel client, see Fig. 7.

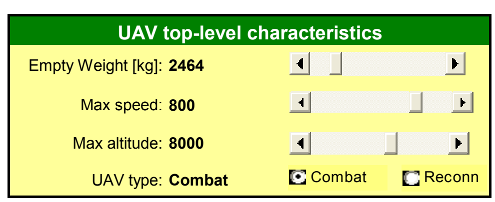

Fig. 7. Controls for rapid modification of top-level characteristics

Changes in these characteristics automatically update the dependent sub-system modules and the framework calculates corresponding sub-system characteristics

System technology selection (Morhological matrix)							
System		Select solution	Alternative solutions				
Cooling system:	Compressor	Tesla	Scroll	Tesla			
	Heat exchanger	Microchannel	Flat	Microchannel			
Actuation system:	System type	EMA	EHA	EMA	Conventional		
	Motor type	DC	PM	SR	Torus	DC	None
Electric system:	System type	MEA	28V	400Hz	MEA		
	Batteries	LiIon	NiCd	LiIon			
Hydraulic system:		FALSE	TRUE	FALSE			
Aux. Power system:		FALSE	TRUE	FALSE			
Systems in study							
Name	**Weight [kg]**	**Required power [kW]**	**Equiv. fuel weight [kg/h]**	**Cost [kkr]**			
Cooling system	23.97	2.57	1.03	142			
Actuation system	68.84	5.22	2.09	2 170			
Electric system	151.58		0.00	1 566			
Sub total:	***244.39***	***7.78***	***3.11***	***3 878***			

Fig. 8. Morphological matrix with interactive controls for technology selection

6.4.2 Interactive Morphological Matrix

By controlling the simulations from the spreadsheet client, it is straightforward to evaluate different technology selections.

Since MS Excel has extensive functionality for matrixes and tables, an interactive *morphological matrix* [13] has been implemented as a tool for concept selection. See Fig. 8. With this matrix, alternative solutions for the different systems can be viewed and selected.

The idea is to store compatibility information for each sub-system and component and as meta-data in each model. The morphological matrix then automatically excludes incompatible and forbidden combinations. This is however not fully implemented.

7 Some Illustrative Results

Finally, an aircraft design example will be presented to further illustrate the MEA study and the tools developed.

In order to optimize the design, a few aircraft variables were selected as design variables. For each variable, an upper and lower limit is set. The optimization module then selects values within the ranges for each variable as defined in Table 1. Note that the values in this example are only illustrative, and might not be realistic.

Table 1: Definition of design variables

Design variable	Variable range: Min Value	Max Value
Wing span [m]	1	10
Root cord [m]	0.2	4
Taper ratio	0.25	1
Thrust [N]	2000	30000
Empty weight [kg]	2000	8000
Fuel weight [kg]	500	1500
Cruise speed [km/h]	10	1000

Below, values for specific characteristics are defined that represents constraints which must not be violated if the design should be valid. Numerical values for the constraints are not presented here.

Constraints: Range (R), Takeoff Field Length (TOFL), Landing Field Length (LFL), Empty Weight (We)

$$C1:\ \text{Range} \geq R_{ref}$$

$$C2:\ \text{TOFL} \leq \text{TOFL}_{ref}$$

$$C3:\ \text{LFL} \leq \text{LFL}_{ref}$$

$$C4:\ \text{We} \leq \text{We}_{req}$$

The objective function is formulized as a function that minimizes the weight of the UAV while not violating the constraints. The ranges of the design variables as well as the constraints are entered in the spreadsheet where also the evolution of the optimization is monitored.

By running an optimization with the above design variables and constraints, the following values for the aircraft empty weight is generated, as visualized in Fig. 9**.** Each 'dot' represents a simulation with specific values of the design variables.

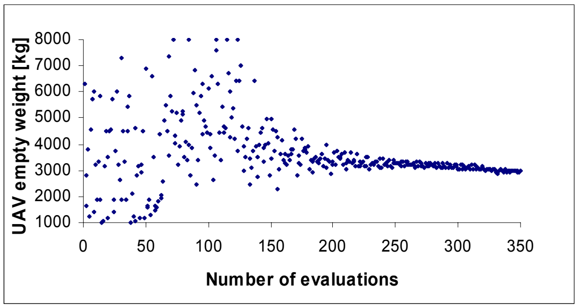

Fig. 9. Results from optimization example

The above optimization converges to an empty weight of the aircraft of approximately 2900kg. Note that some design points implies a lower weight of the aircraft. These points are however not within the constraints, which is why the optimization algorithm converges to a solution with higher weight.

Since the empty weight of the aircraft is input to most of the subsystem modules, the weights of the subsystems can be computed respectively. In Fig. 10, the summarized weight of the sub-systems in the MEA study is visualized corresponding to the design variables that generates the aircraft weight in Fig. 9.

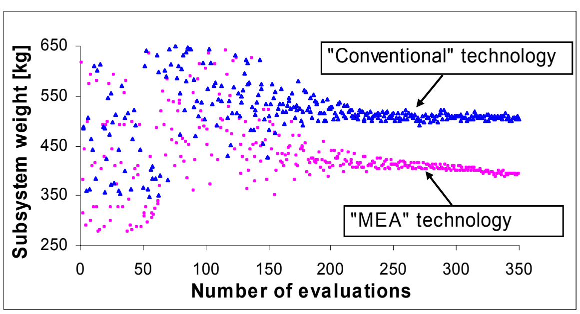

Fig. 10. Results from optimizations with Conventional and MEA technology

In the graph, two different optimizations are plotted with different selections of system technologies. The graph indicates that the MEA technology has lower weight than conventional technology. It is typical that lower weight implies higher cost when selecting technologies. Trade-off calculations are here necessary by bringing such characteristics into the objective function for the optimization. This is however a matter for future work.

8 Conclusions

In this paper, tools and methodologies for a collaborative study of More Electric Aircraft has been presented. Computational tools and methods for collaborative systems development is something that becomes increasingly important when products become more and more complex and product development projects turn more collaborative. By integrating models of the aircraft together with the sub-systems, the complete aircraft can be evaluated and optimized in a more straightforward way.

The presented framework for model integration is built on open standards for web service technology, which is open and well established. This implies a wide range of possibilities, as for example the straightforward deployment of models represented differently. Models can be managed by domain experts and integrated and evaluated by system integrators without exposing the content of the models if not desired.

From an aircraft system design point of view, the presented tools and methods represents a customized design framework with extensive flexibility and potential.

9 Acknowledgement

This work was financed by the Swedish National Aerospace Research Program (NFFP).

10 References

[1] Box M. J., "A new method of constrained optimization and a comparison with other methods," *Computer Journal*, 8:42-52, 1965.

[2] Choi, H.-J., Panchal, J. H., Allen J. K, Rosen D., and Mistree F., "Towards a Standardized Engineering Framework for Distributed Collaborative Product Realization" in *Proceedings of ASME Computers and Information in Engineering Conference*, Chicago, USA, September 2-6, 2003.

[3] Engineous Software, http://www.engineous.com.

[4] Howse, M. "All Electric Aircraft" IEE Power Engineer, Volume 17 Issue: 4, Page:35-37, Aug-Sept 2003.

[5] Johansson B., DeLaurentis D. A. and Mavris D. N., "Managing Design Data for Probabilistic Evaluation of Aircraft Concepts", in *Proceedings of International Conference on Engineering Design, ICED'03*, Stockholm, Sweden, August 19-21, 2003.

[6] Johansson B. and Krus P., "A Web Service Approach for Model Integration in Computational Design", in *Proceedings of ASME Computers and Information in Engineering Conference*, Chicago, USA, September 2-6, 2003.

[7] Johansson B., Jouannet C. and Krus P., "Distributed Aircraft Analysis Using Web Service Technology", in *Proceedings of SAE World Aviation Congress*, Montreal, Canada, September 8-12, 2003.

[8] Krus P., Andersson J. "Optimizing Optimization for Design Optimization", *Proceedings of DETC'03 2003 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, Illinois USA, 2003.

[9] Modelica Association, "Modelica A Unified Object-Oriented Language for Physical Systems Modeling, Language Specification, http://www.modelica.org, 1999.

[10] Morgenthal J.P., *Enterprise Application Integration with XML and Java*, Prentice Hall, Upper Saddle River, 2001.

[11] Mavris D. N. and DeLaurentis D. A., "A probabilistic approach for examining aircraft concept feasibility and viability", *Aircraft Design*, 3 pp.79-101, 2000.

[12] Radcliffe C., Sticken J. and Cosciak G., "The Internet Engineering Design Agent System: iEDA," *Proceedings of IMECE2002 ASME International Mechanical Engineering Congress and Exposition*, New Orleans, USA Nov. 17-22, 2002.

[13] Pahl. G, Betiz. W, Konstruktionslehre, Springer Verlag, ISBN 0-387-0789-7, 1977

[14] Phoenix Integration, http://www.phoenix-int.com.

[15] Senin N., Wallace D.R., and Borland N., "Distributed Object-Based Modeling in Design Simulation Marketplace," *Journal of Mechanical Design*, Vol. 125, 2003.

[16] Tsalgatidou A. and Pilioura T., "An Overview of Standards and Related Technology in Web Services," *Distributed and Parallel Databases,* 12, pp. 135-162, 2002.

[17] Wujek B., Koch P., McMillan M. and Chiang W., "A Distributed, Component-Based Integration Environment for Multidisciplinary Optimal and Quality," *Proceedings of 9th AIAA/ISSMO Symposium of Multidisciplinary Analysis and Optimization*, Atlanta, USA, September 4-6, 2002.

[18] World Wide Web Consortium, http://www.w3c.org.