

MANAGING DESIGN DATA FOR PROBABILISTIC EVALUATION OF AIRCRAFT CONCEPTS

Björn Johansson, Daniel A. DeLaurentis, Dimitri N. Mavris

Abstract

In this paper, tools and standards associated with the extensible markup language, XML, are introduced to facilitate integration of analysis codes for evaluation of aerospace vehicle concepts. The work covers several parts where the most important one is the development of an information model for design data enabling probabilistic sizing and synthesis of the vehicles. This enables straightforward transformation of the design data to various analysis codes for probabilistic analysis. The presented approach has been implemented in computer software where the XML data is translated and the analysis code is executed in accordance with a process description. Finally, the information model and the developed tools are evaluated with a High Speed Civil Transport (HSCT) as a design example.

The advantage with the presented approach is that a rich design document can be used as a central repository for the design. Thanks to the XML standard, it is possible to use a wide range of tools to transform and manipulate the data. Tasks that otherwise would be very demanding to implement becomes more straightforward which enables the engineer to focus on actual design problems rather than programming difficulties. Further, the design information becomes more structured, which increases control and traceability of the increasing amount of information in aircraft design.

Keywords: Design information management, XML, probabilistic design

1 Background and introduction

In the process of conceptual aircraft design, decisions about the design are to a large extent based on results from computational tools. The tools have been developed continuously in the aerospace community during a long period of time and are very important for the design engineer. In order to evaluate future innovative concepts, the tools must be efficiently integrated and design methodologies adopted for computational design must be introduced. One example where efficient computational capabilities are utilized is *probabilistic design*. Using this methodology, variability in the design data is handled throughout the process and the impact on the design characteristics is given valuable measures.

A problem is that integration of analysis codes and implementation of new design methodologies such as probabilistic design is a rather demanding task for the design engineer. This is because most existing analysis codes do not handle that type of information and requires and generates data differently. In order to implement probabilistic capabilities, special solutions for all involved codes are required which is a time consuming and demanding task.

The objectives for the work presented in this paper are to design an information model that includes relevant information for design and analysis of future aircraft concepts. The information model supports probabilistic information for both input and response data. It will also be implemented in a computer software environment to enable integration of analysis

codes and probabilistic analysis, primarily of aircraft concepts. The presented environment is illustrated with a design example where the concept feasibility of a high speed civil transport is evaluated using probabilistic analysis.

1.1 Probabilistic analysis

In all stages of engineering design, and especially in conceptual stages, most available information suffers from uncertainty and it is important to bring this into the design process and through modeling and simulation tools. This is because important information about the design otherwise is omitted. For example, following information can be extracted if the uncertainty is taken into account:

- The probability of meeting a set of requirements and achieve a technically feasible design with a particular concept, the *probability of success*.
- How much it will be necessary to relax the requirements in order to have a sufficiently high probability of success.
- The effect on the probability of success if the concept is modified, for example by infusing new technologies.

The information above can not be achieved using a deterministic approach (where uncertainty is not handled), but is necessary in order to make good decisions based on the simulation. This reasoning has led to an increased interest in the *probabilistic design methodology* where uncertainty is handled and given specific metrics throughout the design process. Well-defined methods have been developed for probabilistic analysis and have been evaluated in several projects showing the significance of this methodology [1], [2]. A typical approach to perform probabilistic analysis is to connect a Monte Carlo simulation to the analysis code which simulates the parameter uncertainty according to a probability density function (PDF). The Monte Carlo simulation is run for a large number of iterations in order to properly simulate the variability and generates input to the actual analysis code. As a result of the varying input to the analysis code, the responses will also vary, enabling a probabilistic analysis of the responses. Probabilistic analysis could also be related to stochastic analysis where also the uncertainty changes over time.

Implementing the probabilistic design methodology where the computational resources are existing old (legacy) codes is a rather complicated task. Since the codes normally not can be modified, the Monte Carlo simulation must be placed outside the analysis code and input data must be generated dynamically, see Figure 1. This requires the user to develop a software environment that executes the Monte Carlo simulation, modifies the model and executes the analysis code.

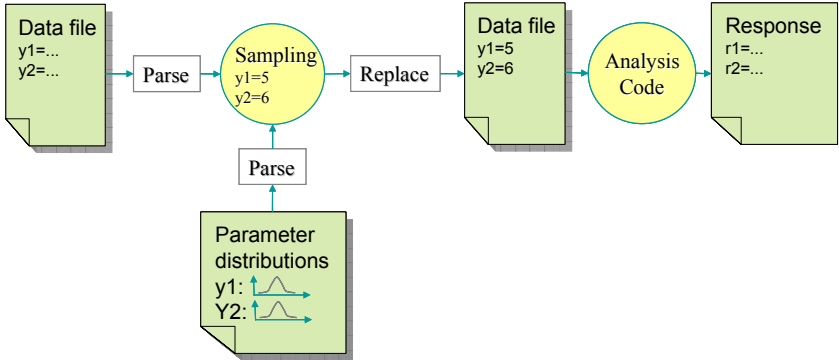


Figure 1. The basic procedure of performing a probabilistic analysis using a non-probabilistic analysis code

It would be an advantage to enable storage of all information related to one concept, including uncertainty, units, documentation etc. in one place, for example a database. This would give a more model centric and uniform approach to handle the design information. It would also facilitate access and analysis of the data from a wide range of tools. The main idea here is also to embed uncertainty information for all variables in the model by choosing from a set of standard probability distributions or by defining custom distributions. It should further also be possible to handle data from external software, such as design of experiments (DOE) software, or an optimization algorithm.

1.2 XML-based data management

Efficient data management requires an ability to access and modify the *content* of a data source, without caring about *structure* and *formatting* of the data. This is a classic information management problem and can also be related to achieving *program-data independence*, [3]. Here, the extensible markup language, XML, is used to reduce this dependence.

XML is a markup language where the fundamental characteristic is its ability to describe and insert markers in a text document and to define the meaning those markers bring to the text. Along with the XML standard a large number of generic tools for parsing and manipulation of the content of the document is available. The XML standard is developed and maintained by the World Wide Web Consortium (W3C,[4]); it is an open standard and is supported by all major software developers. XML is undergoing a rapid development and has been proven very powerful in a wide range of various applications; see for example [5], [6] and [7].

An XML document can easily be converted, either to another XML document or to a non-XML document. The transformation can be defined in a so-called stylesheet using the Extensible Stylesheet Language for Transformation (XSLT). For detailed information about XSLT transformation, see for example [8] and [9].

The approach in this work is to dynamically generate the correct input to the analysis code currently of use by transforming data from an XML data repository, see Figure 2.

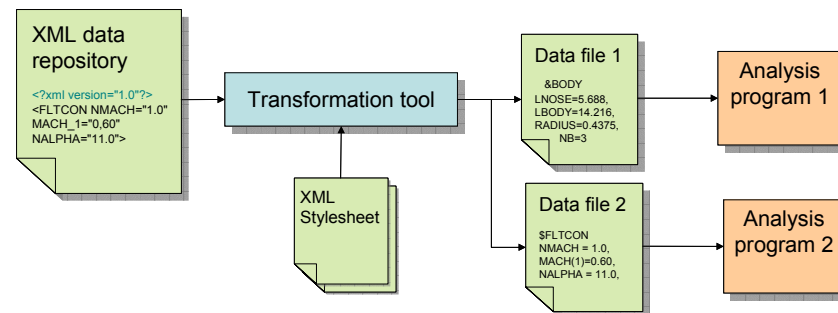


Figure 2. XML solution for reducing program-data dependence

2 Data model for probabilistic aircraft design data

With an XML-schema [4], the structure of the XML instance document is defined. It is a model of the document where all elements are defined as well as their structural relation to each other. Other types of information (*meta-data*) about the instance document can be defined in the schema as well, such as the required number of elements of a certain type, if an element is optional or required, and what data type the stored data will have etc. A schema can be designed in a large number of ways and is an important task since the schema will highly influence the usability of the XML document. The challenge is to find a balance

between generality and complexity in the schema. In this section, the XML-schema will be briefly described.

2.1 System and sub-system data

The task at hand is to design an XML schema for conceptual design of *revolutionary* aircraft. This means that it is very difficult to create a data model with elements representing traditional aircraft components, such as wing, fuselage, tail, etc., since revolutionary concepts might not include any of these items but well other ones. This requires a schema that is general and flexible to suite many purposes. Therefore, the schema is designed by generic elements such as *system*, *sub-systems*, *variables* and *native data*. A top level structure of the data can be seen in Figure 3 and the different parts of the data model are described in more detail below.

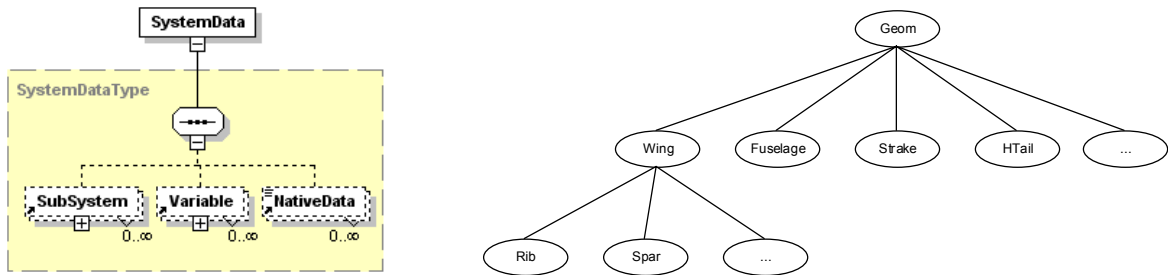


Figure 3. The model of design data at the highest level

The sub system element gives an ability to build object-oriented and hierarchical data sources. It is an advantage to be able to group data belonging to the same physical object and also organize the components hierarchically.

2.2 Design variables

The *variable* is the important building block in the schema which contains the most information. For every variable, meta-data about the variable can be attached such as variable type, description etc. A probability distribution or settings for a design of experiment can also be attached to the variable as in Figure 4. The representation of probabilistic data is further described below.

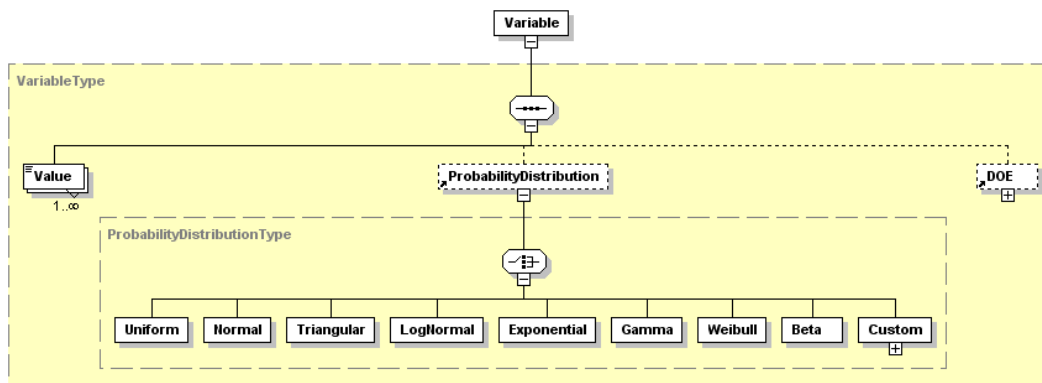


Figure 4. Structure of the variable element

2.3 Representing uncertainty information

Uncertainty information can be stored in a large number of ways. In this case, the following information is stored in order to define the uncertainty information:

- Parametric values for the major standard distributions
- Custom values for non-standard distributions
 - Interval values
 - Single values

This means that no mathematical functions for the distributions are stored. For example, in the case of a normal distribution, the mean value and the standard deviation are stored and not the mathematical function describing the full probability density function, PDF. The PDF is instead created in the software environment implementing the information model.

3 Implementation of software environment

The XML-Schema includes all information about the data in the actual XML instance, not only the structure, but also definition of the data types. The schema can be used to generate a skeleton for a programming environment where the XML data can be directly accessed and modified according to the schema. The procedure is referred to as *data binding* and in this project, the Castor tool for data binding has been used [10], [11]. Castor takes the XML Schema as input and generates Java classes corresponding to the XML elements. This is a very powerful approach to automatically generate a customized interface between data and the software implementation.

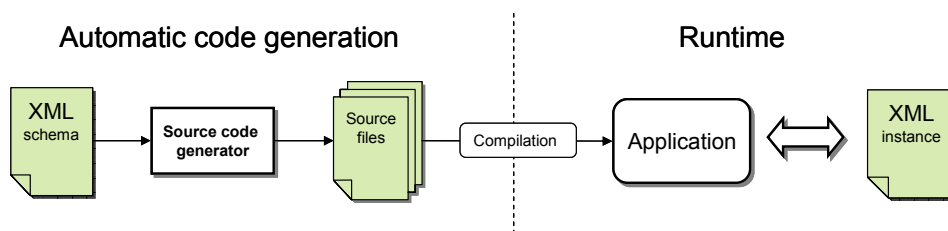


Figure 5. Generation of software application skeleton through data binding

In order to achieve a flexible environment, it is of interest to be able to control the flow of information and the actual process where different activities are executed. An XML-based process description has therefore been developed together with a sequencer. The sequencer is a software module that can interpret an XML-based process description and execute a sequence of *activities* in a generic way.

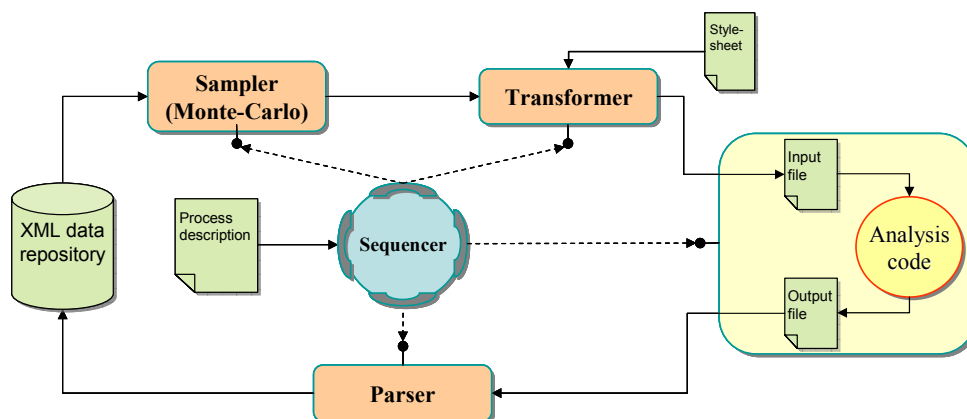


Figure 6. Implemented software framework

The Monte-Carlo simulation is performed using statistical Java package [12] where mathematical representations are built and simulated from the parameters defined in the XML-document.

4 Evaluating example – concept feasibility assessment

In this section, the presented XML schema and software environment will be further illustrated with an example. A High Speed Civil Transport (HSCT) is used for evaluation and is chosen because models and analysis results were available from several previous studies.

In the example, a *concept feasibility assessment* will be performed. This is a probabilistic methodology aiming at selecting or modifying concepts and find the ones that are technically feasible.

The *design space* is represented by the ranges of the *design variables* i.e. the ones the designer is able to modify. A *design space exploration* is conducted by giving the design variables uniform probability distributions, implying that every value has the same probability to occur. Values are sampled from the design space, the analysis is executed and the responses are compared with the constraints. By examining the whole design space, the *probability of feasibility* for a concept can be determined. In Figure 7, a flow chart for this process is illustrated. The design is investigated from a “baseline”, which is initial variable setting. The process will be briefly described in this example. For a detailed description, see [1].

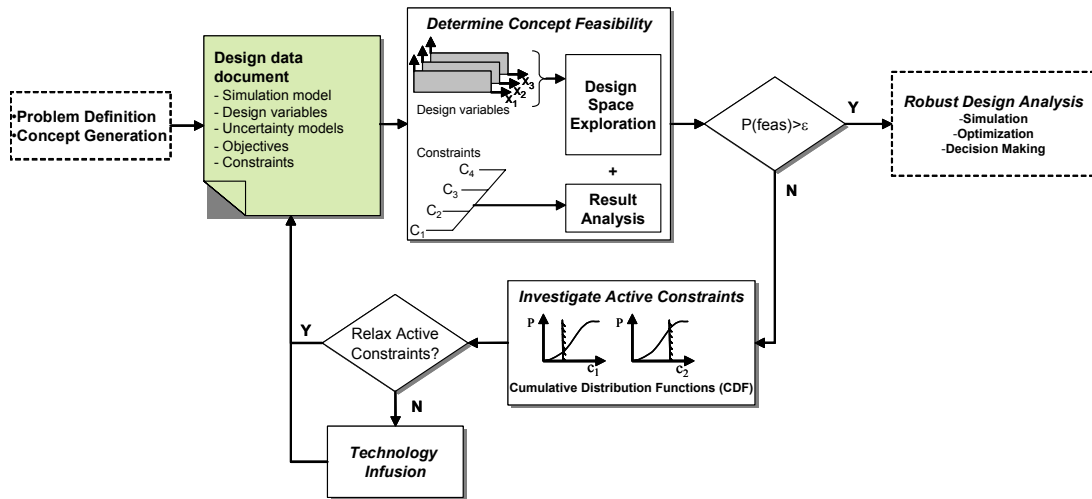


Figure 7. Flowchart of concept feasibility assessment

4.1 Defining the design space

The design space is here assumed to be a continuous space, meaning that a specific design variable is modeled with a continuous uniform distribution with a maximum and a minimum value defining the range. In the following table, four design variables for the HSCT are defined with symbols, units and limits.

Table 1: Design space definition

Design variable	Range	
	Min Value	Max Value
Aspect ratio (AR)	2.1	2.5
Wing area [sq.ft] (SW)	8100	9100
Overall pressure ratio (OPR)	19	21
Turbine inlet pressure [$^{\circ}$ R] (TIT)	3000	3250

The design variables are defined in the XML data file, and uniform distribution, this information is attached to the variable as exemplified below for the aspect ratio variable.

```

<Variable name="AR" info="Aspect Ratio" variableType="Controllable" variabilityType="UniformDistribution">
  <Value>2.3</Value>
  <Distribution>
    <Uniform min="2.1" max="2.5"/>
  </Distribution>
</Variable>

```

So far only control variables has been taken into account, but there is also variability present for the non-control variables, in terms of uncertainty. Uncertainty of the non-control variables is not part of the actual design space exploration and is not evaluated in this example. With this approach, using an XML repository that contains all design data it is however possible to include all data from the beginning, and then choose which data to extract and use at various stages of the process.

4.2 Defining constraints

Since this example only will illustrate the process of investigating the design space, no objective function will be taken into consideration. The constraints, however, can be defined as:

Constraints: Takeoff Gross Weight (TOGW), Takeoff Field Length (TOFL), Landing Field Length (LFL)

$C_1 = \text{TOGW} \leq 770,000 \text{ Lb}$ $C_2 = \text{TOFL} \leq 10,500 \text{ ft}$ $C_3 = \text{LFL} \leq 10,500 \text{ ft}$	<pre> <Response name="TOGW"> <ParseInfo search="TOGW" direction="forward" occurrence="1" offset="0" read="3"/> <Constraints lessThan="770000"/> </Response> </pre>
--	--

Figure 8. Constraints and exemplified XML representation

4.3 Design space exploration

For every loop in the Monte-Carlo simulation, the analysis code *Flight Optimization System, FLOPS* [13], is called with a new input file where the variable values are modified. When the execution is finished, an output file is generated with all the responses that FLOPS computes. It would not be reasonable to store all results from every execution of the code, why a parser is called that extracts the responses that are of interest for the specific analysis. The values of the responses are written back to the XML repository.

4.4 Examining constraints and evaluating probability of feasibility

When the Monte-Carlo simulation is finished, the constraints can be examined on both an individual basis, as well as a combined measure to evaluate the total probability of meeting all constraints, referred to as probability of feasibility, $P(\text{feas})$. Here, the designer's discretion must be used to decide whether $P(\text{feas})$ is "acceptable" or not. $P(\text{feas})$ is too low, the individual responses are investigated to determine which are the most restrictive.

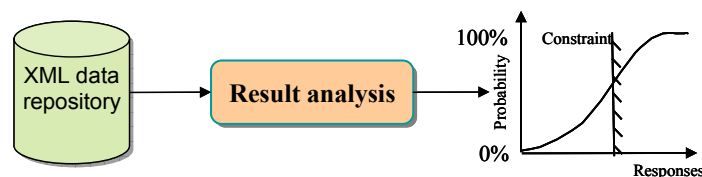


Figure 9. The simulation results are extracted from the XML repository for analysis

The results are investigated by computing a Cumulative Density Function (CDF) for the response of interest. By fitting a normal distribution to the values of the responses, the probability of achieving responses that meet the constraints can be computed.

For the design space and constraints used in this example, the probability of feasibility is:

$$P(\text{feas}) = P\{(GW \leq 770,000) \cup (\text{TOFL} \leq 10,500) \cup (\text{LFL} \leq 10,500)\} = 3\%$$

It is obvious that the probability of feasibility is unacceptably low, why some action must be taken. First, the constraints are examined on an individual basis, giving the following values. In fact, this is information that is already available from the computation of the total probability of feasibility.

$$P(\text{TOGW} \leq 820,000) = 4.4\%$$

$$P(\text{TOFL} \leq 10,200) = 68.5\%$$

$$P(\text{LFL} \leq 10,200) = 100\%$$

The tools designed for this work do not generate a graphical representation of the CDF. However, since the responses are stored in XML, it is easy to use some other software for visualization of the results. For example *Crystal Ball*, from Decisioneering Inc., can be used together with MS Excel to create a graphical representation of the cumulative distribution function.

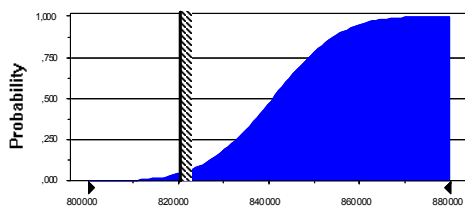


Figure 10. CDF for Takeoff gross weight

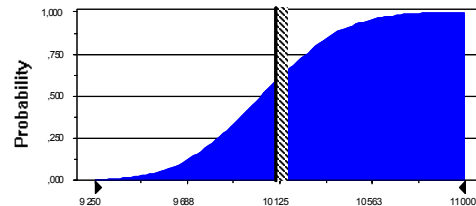


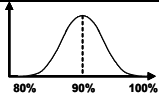

Figure 11. CDF for takeoff field length

It can be seen that the takeoff gross weight is the most restrictive constraint and the following step for the designer would first be to negotiate the constraints and see if they can be relaxed. If this is not possible, the design space must be opened in order to increase the probability of feasibility. To open the design space, either a completely new concept (baseline) must be chosen, which would be the most drastic consequence. First, however, the designer would try to improve the system by infusing technologies that are likely to improve the performance in some respect and thereby open the design space.

The effect of a new technology can at an early stage in the modeling process be approximated with a value representing the change, for example in percent, that the new technology would give and affect the probability of meeting the requirements.

In the example here, we improve our baseline with a series of technologies. First, a series of known engine improvements is proposed to decrease fuel flow. The fuel flow factor is estimated to 90%, but due to the uncertainty of the value, a probability distribution ranging from 80% to 100% is selected to describe the impact of the infused technologies.

The next set of technologies is of a more geometrical type, resulting in a factor to reduce lift independent drag. This factor is also estimated to 0.9, but as for the fuel flow factor, it is modeled with a distribution in order to capture the uncertainty in the value.

Technology factor	Value	Distribution
Fuel flow factor	Normal distribution Mean value: 90% Std. dev. 4%	
Factor to decrease lift independent drag	Normal distribution Mean value: 90% Std. dev. 4%	

Below, a snapshot from the XML-file illustrates how a K-factor is represented. The Kfactor is modeled as a regular variable, but with the attribute *variable type* set to *KFactor*.

```
<Variable name="FCDI" info="Factor to modify drag coefficients" variableType="KFactor">
  <Value>0.9</Value>
  <ProbabilityDistribution>
    <Normal mean="0.9" stdDev="0.04"/>
  </ProbabilityDistribution>
</Variable>
```

The probability of feasibility is now re-evaluated with the technology factors included in the model. Values for the system parameters are sampled from both the design space as well as from the probability distributions representing the technology K-factors. With the settings in this example, the probability of meeting all constraints is increased to 80%, which means that the design space is significantly increased, and the probability of achieving a technically feasible design is acceptable. Below, the graphical representations the CDF with the infused technologies can be viewed and compared to the CDFs in Figure 10.

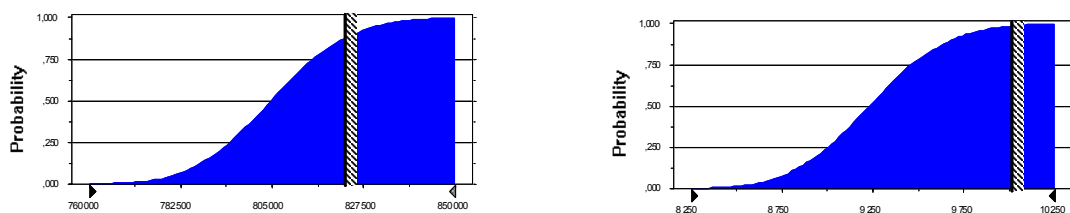


Figure 12. CDFs for takeoff gross weight and takeoff field length with infused technologies

We can now accept the current concept and move on to the next step in the design process, which includes further simulation and optimization to achieve an optimal design point with respect to both performance and robustness. Here it is important to bring in uncertainty models representing the parameters that the designer not has control of, the noise variables. This process is well described in for example in [2], and will not be further discussed in this example.

5 Conclusions and future work

The work presented in this paper illustrates how management of probabilistic aircraft design data can be facilitated using XML technology. In order to create flexible and powerful tools for the designer of complex systems, it is today necessary to make use of the latest technology for information management since this otherwise would become a very demanding task for the design engineer. XML technology has a wide range of advantages to facilitate information management. It has been shown that by using XML technology and standard generic tools, the information currently of interest can be extracted giving functionality comparable to a database system but with greater flexibility.

The work presented here is part of a wider effort to enable integration of simulation tools and methodologies for computational design. By using standards for distributed information exchange, such as the *Web service* technology, different simulation resources can be wrapped with a standardized interface that enables straightforward access over the Internet from a wide range of engineering tools. It also enables definition of a process description i.e. an executable description for the computational process of a specific methodology. This task has been partly covered in this project and will be an important continuation for future work.

References

- [1] Mavris, DeLaurentis “A probabilistic approach for examining aircraft concept feasibility and viability”, Aircraft Design, 3 pp.79-101, 2000.
- [2] Mavris D.N., Bandte O., DeLaurentis D.A., “Robust Design Simulation: A Probabilistic Approach to Multidisciplinary design”, Journal of Aircraft, Vol. 36, No 1, pp298-307, 1999.
- [3] Elmasri, R. and Navathe, S.B., Fundamentals of Database Systems, 3rd ed., Addison-Wesley, 1997.
- [4] World Wide Web Consortium, <http://www.w3c.org>.
- [5] Johansson B., Larsson J., Sethson M. and Krus P. “An XML-Based Model Representation for Model Management Transformation and Exchange” in Proceedings of ASME International Mechanical Engineering Congress, New Orleans, USA, 2002.
- [6] Murray-Rust, P., Rzepa, H.S. and Wright, M., “Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content”, New Journal of Chemistry, pp. 618-634, 2001.
- [7] Nierman, A. and H. V. Jagadish. “ProTDB: Probabilistic Data in XML”, in Proc. of the 28th International Conference on Very Large Data Bases, 2002, Hong Kong, China.
- [8] Apache Software Foundation, The Apache XML Project, <http://xml.apache.org>
- [9] Harold, E. R. and Means, E. S., XML in a Nutshell, O'Reilly & Associates, 2001.
- [10] McLaughlin, B., Java and XML Data Binding, O'Reilly & Associates, 2002.
- [11] Castor, ExoLab Group, San Mateo, CA, USA.
- [12] The Probability/Statistics Object Library <http://www.math.uah.edu/psol/>.
- [13] McCullers L.A. Flight optimization system user's guide, Version 5.94. NASA Langley Research Center, 1998.

Corresponding author:

Björn Johansson

Linköping University, IKP

SE – 581 83 Linköping

Sweden

Tel: Int: +46 13 281190

Fax: Int +46 13 130414

E-mail: bjojo@ikp.liu.se

URL: <http://hydra.ikp.liu.se>