

IceVal DatAssistant – An Interactive, Automated Icing Data Management System

Laurie H. Levinson*

NASA Glenn Research Center, Cleveland, Ohio 44135

William B. Wright†

ASRC Aerospace, Cleveland, Ohio 44135

As with any scientific endeavor, the foundation of icing research at the NASA Glenn Research Center (GRC) is the data acquired during experimental testing. In the case of the GRC Icing Branch, an important part of this data consists of ice tracings taken following tests carried out in the GRC Icing Research Tunnel (IRT), as well as the associated operational and environmental conditions during those tests. Over the years, the large number of experimental runs completed has served to emphasize the need for a consistent strategy to manage the resulting data. To address this situation, the Icing Branch has recently elected to implement the IceVal DatAssistant automated data management system. With the release of this system, all publicly available IRT-generated experimental ice shapes with complete and verifiable conditions have now been compiled into one electronically-searchable database; and simulation software results for the equivalent conditions, generated using the latest version of the LEWICE ice shape prediction code, are likewise included and linked to the corresponding experimental runs. In addition to this comprehensive database, the IceVal system also includes a graphically-oriented database access utility, which provides reliable and easy access to all data contained in the database. In this paper, the issues surrounding historical icing data management practices are discussed, as well as the anticipated benefits to be achieved as a result of migrating to the new system. A detailed description of the software system features and database content is also provided; and, finally, known issues and plans for future work are presented.

I. Introduction

As with any research group, members of the Icing Branch at the NASA Glenn Research Center generate and use a great deal of data in the course of doing their research. For the Icing group, this data most often takes the form of hand-traced and digitized ice shapes, and the associated environmental and operational conditions at the time those ice shapes accrete. Over time, the management of this data has increasingly become a major concern, as thousands of ice shapes have been generated by different researchers under varying conditions and for different purposes. In the past, the individual researchers performing the tests have each been responsible for managing their own test data, meaning the location, reliability, and type of data retained has been subject to a great deal of variability. If, at a later point in time, a researcher wished to access that data, often this would require a great deal of time and effort; and, in some cases, its use would be complicated by missing or conflicting information.

To address this situation, the Icing Branch has recently chosen to implement an interactive, automated data management solution referred to as the IceVal DatAssistant software system. This system has two primary components: a relational database, used to store both experimental and simulation software-generated ice shapes and the associated conditions data; and a database access utility, used to upload, download, and/or display user-selected data. By migrating to this type of solution, members of the Icing Branch have not only been able to improve their current data management practices, saving time and effort, but they will now also be able to provide an improved, more comprehensive product to their customers, while simultaneously laying the foundation for future enhancements to both products and processes.

* Computer Engineer, Mechanics and Life Prediction Branch, 21000 Brookpark Road/MS 49-8, Non-Member.

† Research Engineer, Icing Branch, 21000 Brookpark Road/MS 11-2, Senior Member.

II. Rationale

The experimental ice shapes collected over the last 20 years by researchers at the NASA Glenn Research Center (GRC) have primarily been generated during tests performed in the Glenn Icing Research Tunnel (IRT). The IRT is an instrumented, refrigerated wind tunnel used by researchers and aircraft design engineers to simulate the natural conditions experienced by aircraft when flying through an icing cloud.[‡] A wide range of tests have been performed in the IRT over its years of operation, providing information as to how ice forms on an aircraft's surface under a given set of conditions,¹ the impact a particular ice shape has on an aircraft's performance,² or the effectiveness of a proposed icing prevention³ or ice removal⁴ strategy. Additionally, tests have been run in order to calibrate the tunnel after equipment repairs or upgrades,⁵ to provide validation data for ice prediction software systems,⁶ or for a variety of other icing research-related purposes.[§]

While the data acquired during these experimental tests has generally been collected with a specific purpose in mind, the data captured most often also has value beyond just the immediate objectives of the researcher conducting the test. For example, the results of an experiment performed for purposes of understanding how ice forms under a particular set of conditions may also be used to validate an ice shape prediction code for use under those same conditions;⁷ or the results of a test conducted at one point in time may be used in planning a subsequent test, or evaluating its results.⁸ In either case, having efficient, reliable access to previously collected data could save a great deal of time and effort, and provide valuable information to the research engineer. In addition, given the fact that testing in the IRT costs the taxpayer well in excess of \$100,000 per week, it would certainly seem appropriate to take the necessary steps to ensure the long-term integrity and availability of any data acquired that might be of use at some later point in time.

Although the above rationale may already provide sufficient motivation for undertaking the development of the type of data management solution implemented in the IceVal DatAssistant software, there are also a number of other reasons for the Icing Branch to migrate to this type of solution. While these reasons may differ in their specific focus, the common thread among them is the recognition that, by compiling a comprehensive database consisting of a standardized, reliable data set, and combining that with an easy-to-use graphical user interface (GUI), providing quick and easy access to the data, a number of present and future enhancements to both products and processes become possible:

1) *More thorough, less costly product validation*

In the past, validation of Icing Branch software, such as the LEWICE ice shape prediction code,⁹ has been an extremely labor-intensive process. Even where testing could be automated, collecting the necessary data and ensuring the consistency of format required to conduct reliable automated testing has been a time-consuming, tedious, and error-prone process. Ultimately, the difficulties involved have necessitated that a choice be made between limiting the extent of the validation – and thus the quality of the final product – or, in the alternative, expending considerable resources to complete the necessary work. In the future, however, with the availability of the IceVal system, a comprehensive validation data set will not only already have been compiled, but it will already have a system-enforced consistent format as well, completely eliminating two of the most common impediments to thorough automated validation testing.

2) *Enhanced quantitative software validation techniques*

Prior to the release of LEWICE 2.0,¹⁰ assessments as to the performance of ice shape prediction codes typically were based on subjective, qualitative criteria, such as visual comparisons of actual ice shapes measured in an experimental facility with those produced by an ice shape prediction code.¹¹ The use of this approach was not only due to the technical difficulties involved in performing more rigorous validation tests, as described above, but also due to the lack of predefined quantitative acceptance criteria established by the icing community. For the LEWICE 2.0 validation, however, GRC researchers compiled data from a large number of experimental icing tests (over 800 ice shapes from approximately 400 IRT test runs); and, using this data, they were able to develop an initial version of a utility called THICK. THICK is a software tool used to examine ice shapes and provide quantitative measures, such as icing limits, maximum horn thickness, horn angle, etc., which can then be used to assess the similarity of two ice shapes. Implementation of the THICK utility, and the definition of the associated quantitative measures for the evaluation of ice shapes, represented a significant step forward in the icing community's

[‡] A detailed description of the IRT can be found on the Internet at <http://facilities.grc.nasa.gov/irt/index.html>.

[§] Each cited reference provides an example of the type of testing mentioned. Additional examples can be found elsewhere in the literature.

efforts to move from qualitative to quantitative ice prediction software validation techniques; and the first step in that process was compiling a standardized, consistently formatted data set for use both in formulating the technique and, later, in validating its implementation. With the development of the IceVal system, this data set has now been extended to an even broader range of ice shapes, airfoils, and conditions, and the format and content of the data are now rigorously controlled. Therefore, by combining this enhanced data set with the ability – through use of the GUI – to quickly view and output user-selected data subsets and associated ice shape images, the IceVal DatAssistant software system will facilitate the process of upgrading existing techniques and/or identifying new quantitative approaches.

3) Improved experimental testing

A required step in the process of developing an automated data management system is the definition of a standardized interface for data input and output. In the case of the IceVal system, the interface that has been defined is based on existing Icing Branch standards and practices for the collection of experimental data during IRT testing. By enforcing this standards-based interface, therefore, use of the IceVal system will help to encourage a more rigorous and consistent application of the existing data collection protocol – helping to ensure that the data specified in current Icing Branch procedures is consistently captured and retained. Furthermore, with a comprehensive database of testing data consolidated in one location, and system features permitting selective display of user-specified data subsets, the system can also be used to expose existing data inconsistencies, or gaps in the collected data, providing valuable input to the experimental test planning process. For example, while planning a NACA-0012 airfoil scaling test entry, one could use the IceVal system to display the subset consisting of all scaling tests performed on the NACA-0012, in order to highlight any gaps in the existing data. Missing conditions identified in this manner could then be included in the next test entry.

4) Increased efficiency during the aircraft design process

With the release of the initial version of the IceVal system, all publicly available IRT-generated experimental ice shapes to date with complete and verifiable conditions have been compiled into one electronically-searchable database. For the aircraft design community, which must certify aircraft for flight in icing conditions, combining this electronically-searchable database with the selective display capabilities of the GUI has the potential to provide significant benefit during the design certification process. By using the IceVal system's capability to display specific user-selected test cases, as described below in detail, it will often be possible to quickly locate ice shapes from existing test cases that are associated with an airfoil that falls into the same general classification as the new airfoil to be certified. By using the ice shape from one of these existing test cases as a starting point, the design engineer will then be able to expedite the process of determining an appropriate ice shape for use during certification testing, thereby increasing the efficiency of the aircraft design certification process.

5) New and/or improved analytical methods

The foundation of all scientific inquiry is data; and, certainly, a key factor in the increased pace of scientific progress in recent times is the improved access and data processing capabilities provided by computers. In a similar manner, the consolidation of the experimental icing data, combined with the ease of access and data processing capabilities provided by the IceVal GUI, will facilitate the process of performing statistical analyses or other scientific investigations that might ultimately lead to new and/or improved methods in the icing research field.

6) Ability to implement future enhancements to data management procedures and/or products

As with most organizational process improvement efforts, one of the primary benefits of implementing the IceVal system is the foundation that it will provide for future enhancements to Icing Branch data management practices and/or products. Having a system in place with a well-defined, consistent data interface will underscore for all who use the system precisely what data is and is not being routinely collected during experimental testing. This may then serve as the stimulus that will result in an upgrade to both the experimental testing protocol and the IceVal system itself, so as to capture additional, or different, data. For example, the original spreadsheets containing the experimental test data often list only the icing conditions and not the actual spray bar settings used during a test. As the equations for determining liquid water content and drop size have changed over time, the spray bar settings would be useful for resolving discrepancies in the data; but because this data is not generally documented on the spreadsheet, it is not included in the current version of the database. Likewise, the templates that the researchers use for tracing ice shapes often include notes about feather formation or anomalies witnessed during a test. While these templates are now digitally scanned, the additional information is not being

transferred to the spreadsheets, and thus is also not currently included in the database. With an initial version of the system in place, however, any of these enhancements can be readily implemented in the future.

In a similar fashion, if, in the future, a new area of icing research should require that different data be captured or additional computations be routinely performed, the currently existing system can be modified to incorporate this new capability. With most efforts such as this, the most difficult step is the first one; thus, the expectation is that any future enhancements to Icing Branch data management products and processes will now require only evolutionary, not revolutionary, change.

III. Legacy Data Overview

When an effort was first made to consolidate experimental data for use in ice shape prediction code validation, the data that had been collected during IRT test runs was discovered to be in a wide variety of formats. Each researcher was responsible for managing his or her own test data; and, although data collection and naming standards were in place at the time, the extent to which the standards were being followed, and the individual interpretations of those standards, varied. In addition, much of the documented data was stored as hardcopy information, and was thus not available electronically. The most commonly used methods for documenting data from experimental test runs were test matrices, run logs, and icing templates.

A test matrix is a chart, often in the form of an Excel file, showing the planned or actual conditions used for an entire test entry, which generally consists of multiple test runs, or sets of conditions. An example of a test matrix is shown in Fig. 1. The run log is a hand-written, hardcopy sheet containing information from a single test run - i.e., one set of conditions - and includes information as to the actual conditions run, observations made during the test, etc. A sample run log is shown in Fig. 2, below. An icing template is a rectangular piece of cardboard with the contour of the relevant airfoil cut into it and a hand-drawn ice tracing image documented onto it. The ice tracing is created by entering the tunnel following the test and manually tracing the ice shape that formed onto the cardboard. Aside from the actual ice tracing itself, the template might also contain miscellaneous notes made by the researcher during or after the test. A typical icing template is displayed in Fig. 3.

		Date:	8/12/2003												Page:	1		
		Config:	Twin Otter with flap												Engineers:	Potapczuk/Miller		
		Spray 1										Spray 2						
Start Time	Run #	Test Condition #	Airspeed (knots)	Total (degF)	Tstatic (degF)	AOA (degrees)	MVD, (um)	LWC, (g/m3)	Spray Pair (psi)	Spray DeIP (psi)	Spray Time, (min)	Nozzle	Spray 2 MVD, (um)	LWC, (g/m3)	Spray Pair (psi)	Spray DeIP (psi)	Spray 2 Time, (min)	Nozzle
17:20:49	1	1	130	29.0	25	2.5	20	1.34	19.5	12	1	STD	130	0.5	2	9	1	MOD
18:14:26	2	2	130	29.0	25	2.5	20	1.34	19.5	12	10	STD	130	0.5	2	9	10	MOD
19:03:34	3	3	130	29.0	25	2.5	130	0.5	2	9	10	MOD	20	1.34	19.5	12	10	STD
19:54:20	4	4	130	29.0	25	2.5	20	1.34	19.5	12	15	STD	130	0.5	2	9	5	MOD
20:41:20	5	5	130	29.0	25	2.5	20	1.34	19.5	12	5	STD	130	0.5	2	9	15	MOD
21:18:00	6	6	130	29.0	25	2.5	130	0.5	2	9	15	MOD	20	1.34	19.5	12	5	STD
22:06:28	7	7	130	29.0	25	2.5	130	0.5	2	9	5	MOD	20	1.34	19.5	12	15	STD
not run		8	130	29.0	25	2.5	130	0.5	2	9	20	MOD	-	-	-	-	-	
22:49	8	9	130	29.0	25	2.5	20	1.34	19.5	12	20	STD	-	-	-	-	-	
not run		10a	126	10.6	6.8	3.5	42	1.32	25.4	151.2	1	MOD						
		10b	126	10.6	6.8	3.5	20	1.2	37.7	136.5	2.7	MOD						
		10c	126	10.6	6.8	3.5	24	0.55	10	24.5	3.5	MOD						

Figure 1. Final test matrix, completed following a planned 10-run IRT test entry.

In addition to the test matrices, run logs, and hand-drawn icing templates, beginning in 1991, ice tracings taken following IRT test runs were, in some cases, digitized and stored electronically. Routine digitization of hand-drawn ice tracings began in the mid 1990s.

Other than the above, the primary source of experimental data from IRT test runs is the data taken by the IRT's Escort-D real-time data acquisition system. This system is set up prior to the beginning of a test, and records user-selected raw data, such as fan speed, air speed, spray bar pressure, air pressure, temperature, etc., and can be used by the researchers to verify information documented on the final test matrix.

While all of this data is available, in theory, for every test run, as a practical matter, this is often not the case. Because the data is managed by the individual researcher, and much of it is stored in hardcopy form, locating a specific data set may require a significant effort. Furthermore, because of differences in experimental technique from one researcher to the next, the extent to which the relevant conditions have been explicitly documented can vary widely. Moreover, since plans made prior to a test may change during the test based on interim results, it is not uncommon to find conflicting information as to the actual conditions run, complicating interpretation of the data.

In response to these difficulties, and in an effort to remedy the situation, the Icing Branch has chosen to implement the IceVal system. The architecture and principal features of this system, as well as the process used for its development, are described in the remaining sections of this paper.

SLD Mass Loss Experiment #5 - IRT Run Sheet

Date: <u>12/18/02</u>	LWC: <u>1.0</u> g/m ³	
Run No.: <u>17</u>	MVD: <u>1.00</u> μm	
Spray on: <u>20:39:49</u>	T _{air} : <u>5.2</u> °F	
Spray off: <u>20:46:50</u>	V _{air} : <u>115</u> mph	
Spray time: <u>7.0</u> min	P _{air} : <u>6</u> psig	
	ΔP: <u>30</u> psi	
Escort Rtg. No.: <u>18259</u>	Camera: <u>Kodak DCS 620</u>	
Tracing: <u>46-56-11e</u>	Roll No.: _____	
Ice Mass: <u>237.9</u> grams	Frames: <u>66-71</u>	
Model: <u>NACA0012</u>	Configuration: <u>Vertical</u>	AOA: <u>0.0°</u>
Chord: <u>21.36</u>		
Pat Weight: Empty <u>132.6g</u>	Full <u>471.0g</u>	Ice <u>338.9g</u>
Comments: Kodak shorts 66-67 were of the model installed Kodak shorts 68-71 were of the new model No problems ice came off easily.		
Research Engineer: <u>Mark Petroszek</u>		

Figure 2. Sample IRT run log.

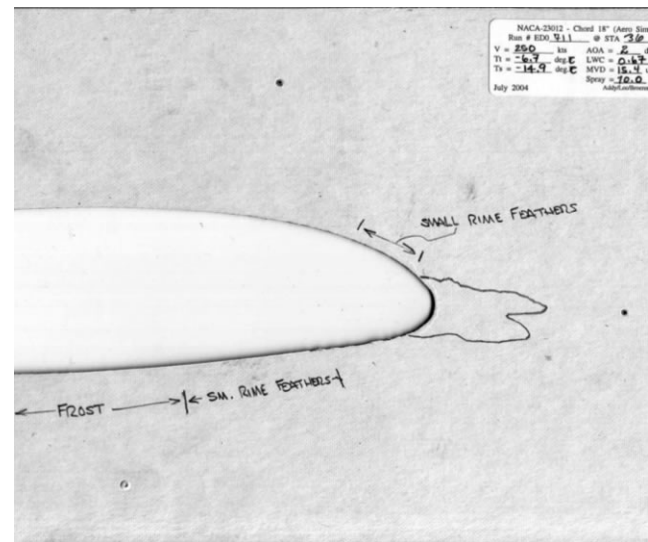


Figure 3. Typical icing template, showing hand-traced ice shape, and researcher notes documented onto the template.

IV. Software Development Process

The traditional software development process, utilized for many years by software practitioners, is referred to as the “waterfall model” for software development. This is a sequential process, beginning with requirements definition, and proceeding through design, implementation, verification, and maintenance in a linear fashion – completing each phase, and its associated documentation, before moving on to the next one. In recent years, this process has fallen into increasing disfavor, considered too inflexible and/or to involve too much overhead for today’s fast-paced, rapidly evolving, and highly competitive market. Instead, a variety of other lifecycle models have been proposed, from the “spiral model,”¹² which has much in common with the original waterfall model, to the so-called “agile development” models,¹³ such as Extreme Programming or Scrum, which have much less in common with the waterfall approach, tend to minimize documentation, and – as the name implies – emphasize rapid delivery of working software, and the ability to quickly respond to change.

For the development of the IceVal software system, the process used, best described as a “rapid” or “evolutionary prototyping”¹⁴ approach, falls somewhere in the middle of the range in terms of formality of both process and product – still incorporating the most essential forms of documentation, although less formally implemented, and utilizing a more user-centric, iterative approach to design and implementation than typically found with the more traditional methodologies.

To begin the process, the first step is to define and document an initial set of system requirements. For the IceVal system, this was accomplished through informal discussions with users, the results of which were then documented in detail and posted to the team web site for review. Once agreement had been reached as to the initial requirements, a preliminary version of the system was implemented, with emphasis on providing a working prototype as soon as possible. Once this prototype was available, and each time a relevant upgrade in functionality was implemented, the software was demonstrated for the users; and feedback from these sessions was then integrated into both the requirements documentation and the software itself.

From a design standpoint, an effort was made to create a system that was very modular in nature, consisting primarily of small, reusable, single-function routines with relatively transparent logic. In the few cases where longer, more complex routines were required, Program Design Language (PDL), or pseudocode – a structured, English language representation of the logic – was written, and was then included as a header in the associated routine.

As the IceVal system is an interactive system, testing was primarily function-based, and was likewise accomplished in an iterative fashion; i.e., when a particular system capability was added or updated, that portion of the system was rigorously tested to ensure that all features would perform as planned. In addition, wherever possible, “round-trip” testing was also performed; i.e., a file originally used as input to the system, was compared with the associated output file generated by the system using the data originally provided by the input file.

V. Database Description

The database component of the IceVal system consists of a Microsoft Access 2003 database file with nine individual database tables, as described below in Table 1. The specific parameter fields defined in each table, and the assigned data type for each of those fields, are as shown in Fig. 4. In general, the data type used for a parameter field is consistent with the native type of the data, as expected. However, the exception to this is in the case where a value that would normally be expected to be read in as a number is instead read in as a text value. Where this occurs, it is either due to the fact that the input parameter associated with that field may have a non-numeric character embedded in the input field along with the parameter value itself – e.g., a ‘SpanLocation’ value specified as ‘36”’ in the input file; or, in the alternative, the associated input parameter may not always have a numeric value at all – e.g., an ‘UpperHornAngle’ value may be displayed in the input file as ‘N/A.’

In terms of structure, the database tables are primarily defined along functional lines, i.e., the data is organized such that parameters are located in the same table with other parameters that have a similar purpose. Beyond this,

Table 1. IceVal Database Table Descriptions

Table Name	Table Index	Table Description
AirfoilCoordinates	AirfoilName	Nominal x and y coordinates for all airfoils contained in the database
GridlineCoordinates	Degrees	Start and end x and y coordinates, used by Excel to plot image reference points displayed on the ice shape image chart
IceShapeData	RunID	Airfoil and ice shape x and y coordinates for all ice shapes contained in the database
IceThicknessData	RunID	Ice thickness vs. wrap distance for all ice shapes contained in the database
RefConstants	Name	The set of reference constant values, and associated units, used in the calculation of run constants for each test case contained in the database
RunConstants	AltRunID	The set of run constant values associated with each test case contained in the database, calculated using data from the RunSpecs, SprayConditions, and RefConstants tables, utilizing equations found in the 2004 report by Anderson ¹⁵
RunSpecs	RunID	Basic run parameters with values that remain unchanged, such as test date, test objective, airfoil name, etc., for each individual test case contained in the database
SprayConditions	AltRunID	The set of run parameters with values that may vary during an individual test run for each test case contained in the database
ThickUtilityData	RunID	Ice shape parameter data computed by the THICK utility for each test case contained in the database

Field Name	Field Type
AirfoilName	Text
CoordIndex	Long Integer
AirfoilX	Double
AirfoilY	Double

a) AirfoilCoordinates

Field Name	Field Type
Degrees	Long Integer
Xstart	Single
Xend	Single
Ystart	Single
Yend	Single

b) GridlineCoordinates

Field Name	Field Type
RunID	Text
CoordIndex	Long Integer
AirfoilX	Double
AirfoilY	Double
IceShapeX	Double
IceShapeY	Double

c) IceShapeData

Field Name	Field Type
RunID	Text
CoordIndex	Long Integer
IceThickness	Double
WrapDistance	Double

d) IceThicknessData

Field Name	Field Type
Name	Text
Value	Single
Units	Text

e) RefConstants

Field Name	Field Type
AltRunID	Text
RunID	Text
LWC	Single
MVD	Single
SprayTime	Single

f) SprayConditions

Field Name	Field Type
AltRunID	Text
RunID	Text
StaticTempF	Single
StaticTempC	Single
StaticTempR	Single
StaticTempK	Single
FilmTempC	Single
FilmTempR	Single
TotalTempR	Single
MachNumber	Single
VelocitySI	Single
VelocityEng	Single
MVDft	Single
ChordFt	Single
StaticPressure	Single
AirDensity	Single
AirViscosity	Single
SpecHeatWater	Single
AirThConductivity	Single
ChordBasedRe	Single
MVDBasedRe	Single
LEDiameterBasedRe	Single
PrandtlNum	Single
HeatXfrCoeff	Single
InertiaParamK1	Single
Lambda	Single
ModInertiaParamK0	Single
StagnationBeta	Single
ModStagnationBeta	Single
LatentHeat	Single
ScalingE	Single
HeatOfVaporization	Single
VaporPressure	Single
MassDiffusivity	Single
SchmidtNum	Single
MassXfrCoeff	Single
EvapMassLoss	Single
RelHeatFactorb	Single
DropEnergyXfrPhi	Single
AirEnergyXfrTheta	Single
FreezingPotential	Single
FreezingFraction	Single
RelHeatFactorbMod	Single
FreezingPotentialMod	Single
FreezingFractionMod	Single
FreezingFractionPerCentDiff	Single
AccumParamByChord	Single
AccumParamByLEDiameter	Single

g) RunConstants

Field Name	Field Type
RunID	Text
Date	Text
Airfoil	Text
TestObjective	Text
SpanLocation	Text
LeadEngineer	Text
Chord	Single
AirSpeed	Single
AOA	Single
CorrectedAOA	Single
TotalTemperature	Single
IPSON	Text
SpanAngle	Text
LinkedRunID	Text
RepeatConditionType	Text
LEDiameterByChord	Single

h) RunSpecs

Field Name	Field Type
RunID	Text
IceStartIndex	Text
IceEndIndex	Text
LEcylinderXcenter	Text
LEcylinderYcenter	Text
AirfoilLEXstag	Text
AirfoilLEYstag	Text
IcingLimitXlow	Text
IcingLimitXhi	Text
IcingLimitYlow	Text
IcingLimitYhi	Text
IcingLimitSlow	Text
IcingLimitShi	Text
IceThicknessLowerMax	Text
IceThicknessLEMin	Text
IceThicknessUpperMax	Text
LowerIceArea	Text
UpperIceArea	Text
TotalIceArea	Text
MaxThicknessIceCoordXlower	Text
MaxThicknessIceCoordYlower	Text
MaxThicknessIceCoordXupper	Text
MaxThicknessIceCoordYupper	Text
MaxThicknessSurfCoordXlower	Text
MaxThicknessSurfCoordYlower	Text
MaxThicknessSurfCoordXupper	Text
MaxThicknessSurfCoordYupper	Text
LowerHornAngle	Text
UpperHornAngle	Text

i) ThickUtilityData

Figure 4. IceVal database table definitions, showing the fields included in each table and their associated data types.

parameters are further arranged, as needed, so as to normalize the data contained in the database, i.e., to prevent the unnecessary repetition of the same information in multiple database records. For example, the ‘RunSpecs’ and ‘SprayConditions’ tables both contain run parameters associated with the various test runs included in the database. However, for some of the runs in the database, several different settings are used during a single test run for the parameters contained in the ‘SprayConditions’ table; whereas the settings for parameters contained in the ‘RunSpecs’ table never change during a single run. The parameters have been separated into these two distinct tables in order to permit the storage of all values for the parameters that may have values that change, while preventing the unnecessary duplicate storage of the data that doesn’t change. Records from the two tables are then linked using a common value – in this case, the ‘RunID’ value.

VI. Software System Functionality

The IceVal DatAssistant software system is an interactive, Windows-based application developed using Microsoft Visual Basic 6.0. As is the case for other Windows applications, from the user’s perspective, operation of the system is controlled via a set of graphical forms – along with the various buttons and other graphical elements contained on those forms – that, together, constitute the system’s GUI. While this notion of the system’s mechanism of operation is not entirely accurate – in reality, operation of the system is accomplished via the logic contained in the forms’ “event-handlers”[¶] and supporting Visual Basic modules – it nevertheless serves as a useful frame of reference for an explanation of the system’s features.

The IceVal software system structure is shown below in Fig. 5. Modules whose names begin with the prefix, “frm,” are the system’s form modules; whereas the remaining modules represent the Visual Basic support modules. These support modules - with the exception of ‘MainModule,’ whose function is to initiate program execution - contain subroutines and functions utilized, either directly or indirectly, by one or more of the form modules’ event handlers.

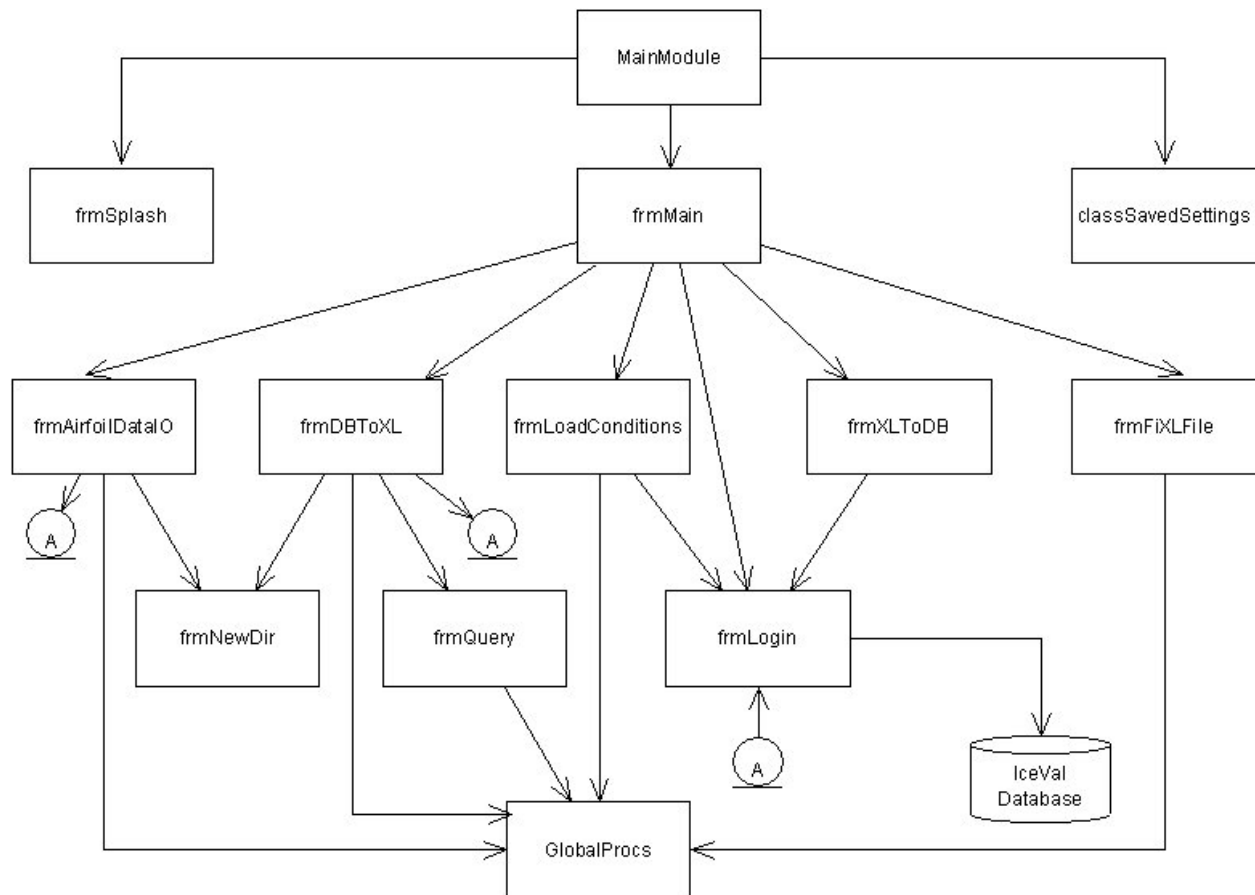


Figure 5. IceVal DatAssistant software system structure.

[¶] An “event handler” is a subroutine that is executed in response to the occurrence of a specific “event” that occurs in the environment, such as a user clicking the mouse or entering text in a textbox

The system's form modules, and the features implemented on each associated form, are as follows:

frmSplash

This module contains the logic required to display and operate the system splash screen, shown in Fig. 6. This screen appears when the user first starts the IceVal application.

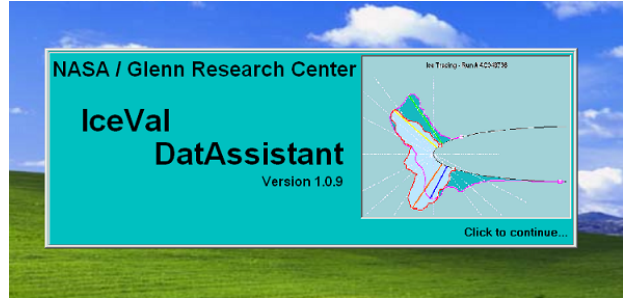


Figure 6. System splash screen, displayed at system startup.

frmMain

This module contains the logic used to display and operate the main user screen, shown in Fig. 7 below. From this screen, access can be obtained to all other forms in the system.

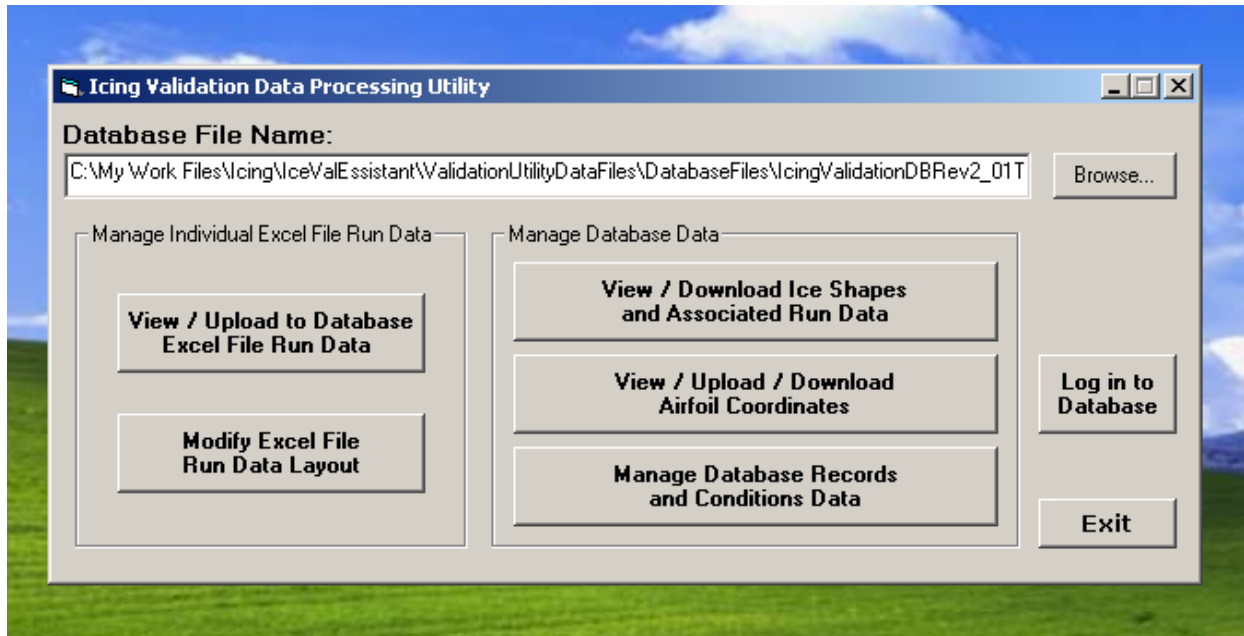


Figure 7. Screen shot showing the IceVal DatAssistant main user screen.

frmAirfoilDataIO

The form associated with this module (see Fig. 8) is used to upload, download, and/or display user-selected airfoil coordinates.

Airfoil ID:	Index	AirfoilX	AirfoilY
CYLINDER	1	1	0
GLC305	2	0.9981	-0.0435755
LTHS	3	0.9924	-0.08683
MS317	4	0.98295	-0.129405
N0012-21	5	0.96985	-0.171015
N4415MOD	6	0.95315	-0.211305
NACA0012	7	0.933	-0.250005
NACA0012_35	8	0.9096	-0.286785
NACA0015	9	0.883	-0.321395
NACA23012			
NACA23014MOD			
NACA4415MOD			
NLF0414			

Figure 8. The frmAirfoilDataIO form, used to process airfoil coordinate data.

frmDBToXL

The form rendered by this module (see Fig. 9) enables the user to view selected data from the database and, if desired, output that data to individual Excel files.

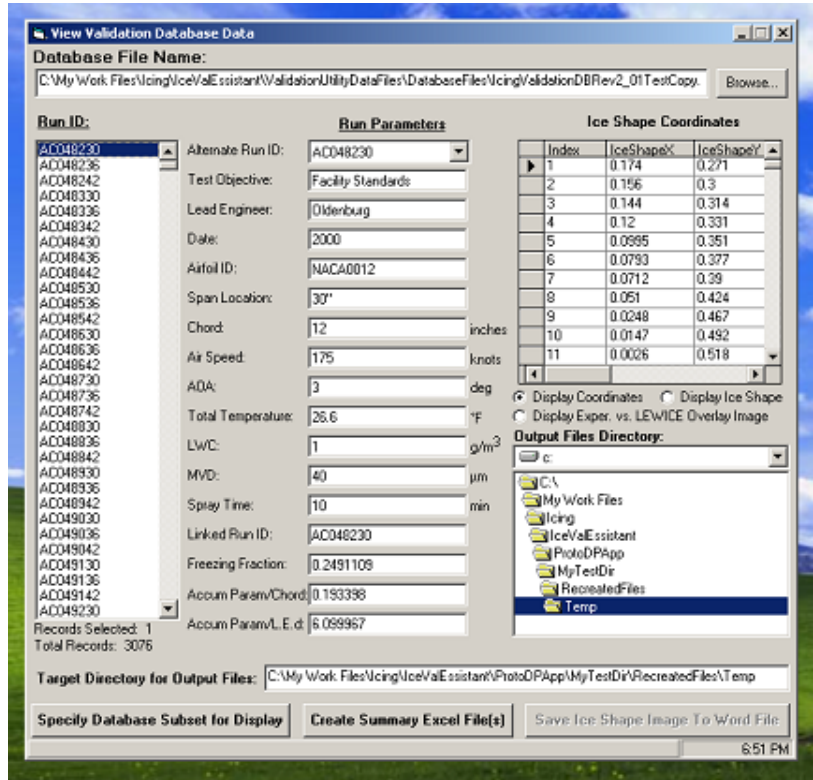


Figure 9. Screen shot of the frmDBToXL form, used to view and/or output user-selected database data.

frmLoadConditions

The form associated with this module, displayed in Fig. 10, can be used to upload/download icing conditions data to/from the database, as well as to add, delete and/or update selected database records.

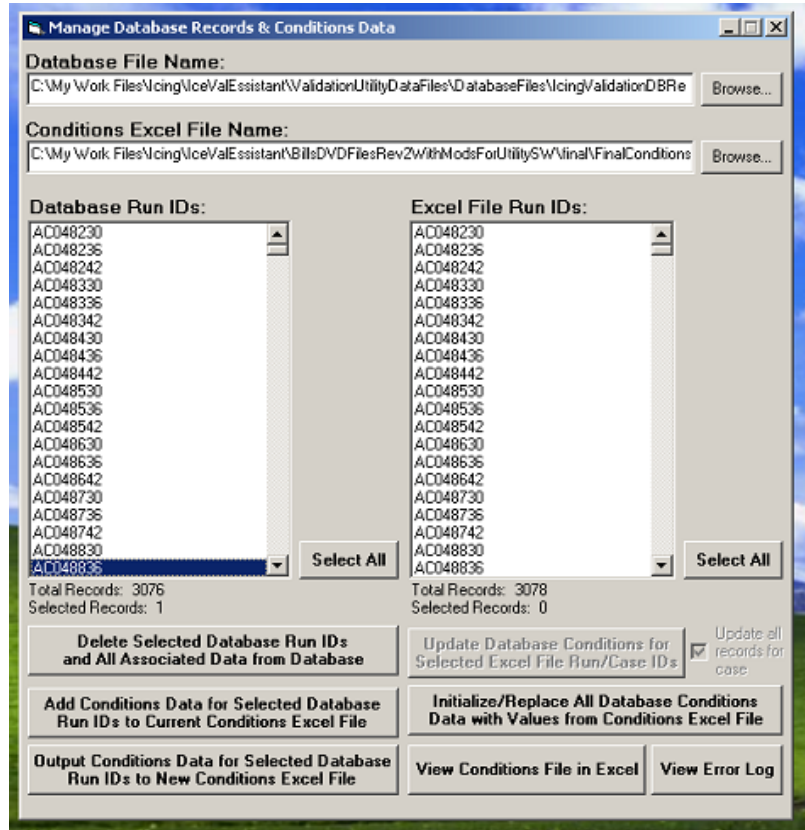


Figure 10. The frmLoadConditions form, used to manage database records in general, and to upload or output database conditions data.

frmXLToDB

Excel file data may be viewed and/or uploaded to the database via the interface provided by this form, shown in Fig. 11.

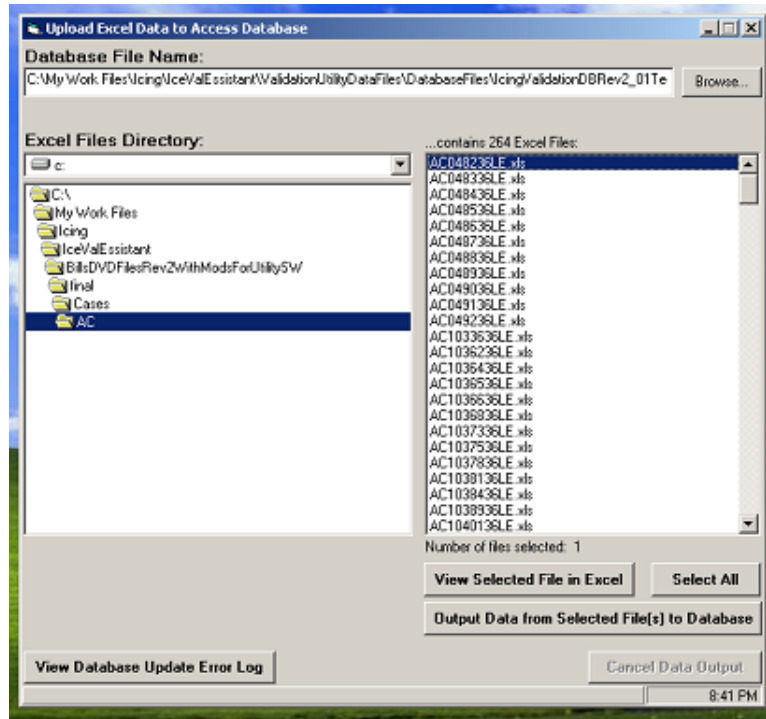


Figure 11. The frmXLToDB form, used to view and/or upload experimental run data from Excel input files.

frmFiXLFile

The frmFiXLFile form, displayed in Fig. 12, is used to correct the format of input files whose layout does not conform to the system's expected Excel file input format.

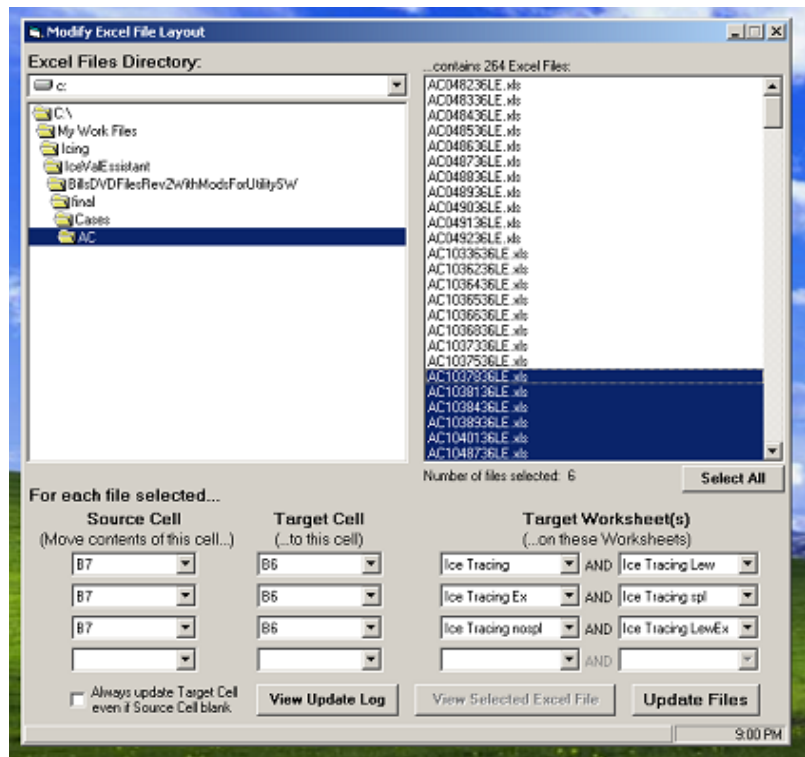


Figure 12. The frmFiXLFile form, used to modify the layout of input Excel files.

frmNewDir

This dialog, shown in Fig. 13, is available from either the frmAirfoilDataIO or the frmDBToXL form, and is displayed when the user chooses to create a new subdirectory.

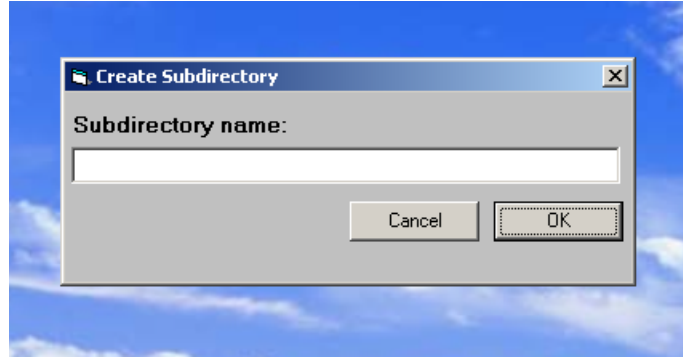


Figure 13. The frmNewDir dialog, which enables the user to create a new subdirectory.

frmQuery

The frmQuery form, shown in Fig. 14, and accessed by clicking on the frmDBToXL form's "Specify Database Subset for Display" CommandButton, provides the user with an easy-to-use, graphical mechanism for specifying database "queries," or requests to the database to provide a specific data subset for display.

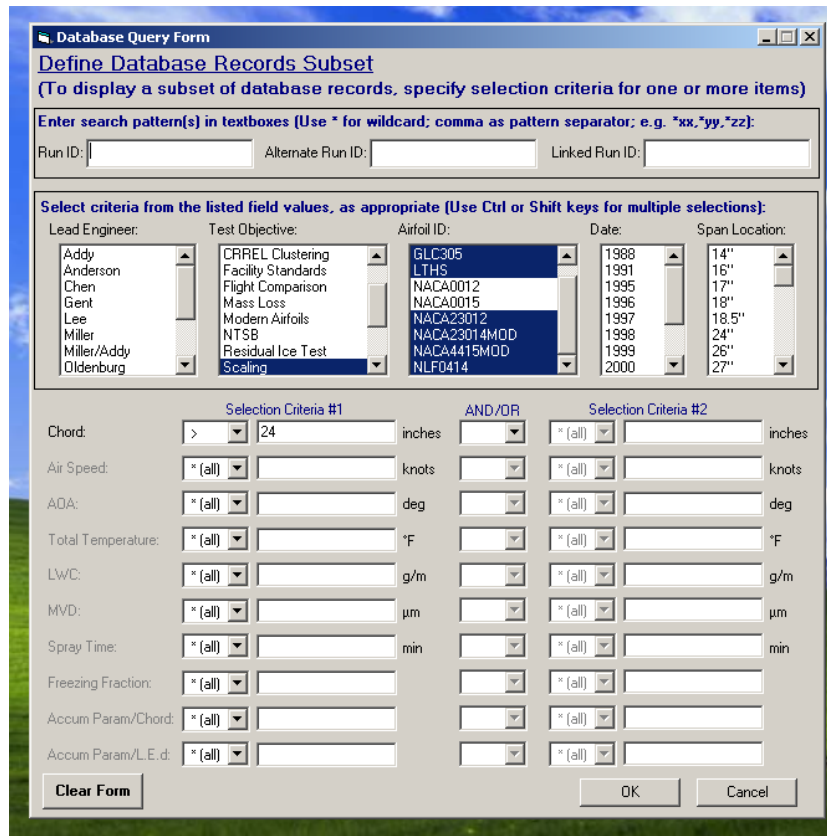


Figure 14. Screen shot of the frmQuery form, used to set up database queries which permit the display of user-selected data subsets.

frmLogin

The Database Login dialog, shown in Fig. 15, serves as the system interface to the Microsoft Access database, with the module containing the logic required to process a user's login and assign appropriate access privileges. The dialog can be accessed either directly, by clicking on the frmMain CommandButton labeled, "Log in to Database," (see Fig. 7, above) or indirectly, by performing an action which requires access to the database.

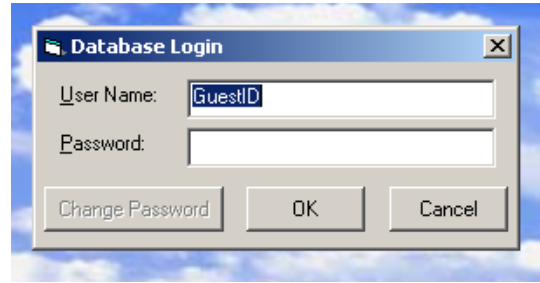


Figure 15. The frmLogin dialog, used to log in to the database and assign access privileges.



Figure 16. Extended form of the frmLogin dialog, used to change the specified user's password.

The extended form of the Database Login dialog, displayed in Fig. 16, is enabled by clicking on the dialog's "Change Password" CommandButton once the user has logged in to the database. This form of the dialog can be used to change the specified user's password.

VII. Known Issues / Future Work

While the currently existing system has met all the existing system requirements, there are nevertheless areas of future work which would ultimately result in an improved product. These include:

- 1) Migration to a higher capacity, more robust and/or public domain database, due to the limitations imposed by the use of Microsoft Access;
- 2) Enhanced error handling;
- 3) Enhanced database security;
- 4) Implementation of an improved help system, including an electronically-accessible User's Guide;
- 5) Incorporation of user-requested enhancements, such as the ability to overlay two user-selected images, or the addition of selected fields to the database; and
- 6) Integration of the IceVal system into the GlennICE framework.

VIII. Conclusions

With the release of the IceVal DatAssistant icing data management system, the icing community now has access to a comprehensive, electronically-searchable ice shape database via an easy-to-use, full-function, interactive GUI. This system currently provides the ease and reliability of access, as well as the consistency of data format, to enable more thorough, less costly icing software validation, while simultaneously enhancing the potential for future upgrades to existing techniques and/or the identification of new quantitative approaches. IceVal provides experimentalists with a tool that can be used to improve experimental testing, and aircraft design engineers with a mechanism to increase the efficiency of the aircraft design process. At

the same time, the IceVal system facilitates the process of performing the type of broad-based, all-encompassing investigation that might one day lead to new and/or improved analytical methods, and lays the foundation for future enhancements to icing data management procedures and products.

References

- ¹Olsen, W., Shaw, R., and Newton, J., "Ice Shapes and the Resulting Drag Increase for a NACA 0012 Airfoil," NASA TM-83556, Jan. 1984.
- ²Addy, H. E., Jr., "Ice Accretions and Icing Effects for Modern Airfoils," NASA TP-2000-210031, DOT/FAA/AR-99/89, April 2000.
- ³Anderson, D. N. and Reich, A. D., "Tests of the Performance of Coatings for Low Ice Adhesion," NASA TM-107399, AIAA-97-0303, Jan. 1997.
- ⁴Shin, J., Bond, T. H., and Mesander, G. A., "Results of a Low Power Ice Protection System Test and a New Method of Imaging Data Analysis," NASA TM-105745, June 1992.
- ⁵Ide, R. F. and Oldenburg, J. R., "Icing Cloud Calibration of the NASA Glenn Icing Research Tunnel," NASA TM-2001-210689, ARL-TR-2383, AIAA-2001-0234, March 2001.
- ⁶Wright, W. B., "Validation Results for LEWICE 3.0," NASA CR-2005-213561, AIAA-2005-1243, March 2005.
- ⁷Vargas, M. and Reshotko, E., "Physical Mechanisms of Glaze Ice Scallop Formations on Swept Wings," NASA TM-1998-206616, AIAA-98-0491, Jan. 1998.
- ⁸Miller, D. R., Potapczuk, M. G., and Langhals, T. J., "Additional Investigations of Ice Shape Sensitivity to Parameter Variations," NASA TM-2006-214227, AIAA-2006-0469, March 2006.
- ⁹Wright, W. B., "User Manual for the NASA Glenn Ice Accretion Code LEWICE Version 2.0," NASA CR-1999-209409, Sept. 1999.
- ¹⁰Wright, W. B. and Rutkowski, A., "Validation Results for LEWICE 2.0," NASA CR-1999-208690, Jan. 1999.
- ¹¹Wright, W. B., "Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 1.6", NASA CR-198355, June 1995.
- ¹²IEEE Standard Glossary of Software Engineering Terminology," IEEE Std 610.12-1990, IEEE Software Engineering Standards Collection, Sept. 1994.
- ¹³Highsmith, J.; Cockburn, A., "Agile Software Development: The Business of Innovation," IEEE Computer, Vol. 34, No. 9, Sept. 2001, pp. 120 - 122
- ¹⁴McConnell, Steve, *Rapid Development: Taming Wild Software Schedules*, Microsoft Press, Redmond, Washington, 1996, Chap. 21.
- ¹⁵Anderson, David N., "Manual of Scaling Methods," NASA CR-2004-212875, March 2004.