NASA/TM—2010-216909

GLobal Integrated Design Environment (GLIDE): A Concurrent Engineering Application

Melissa L. McGuire
Glenn Research Center, Cleveland, Ohio

Matthew R. Kunkel and David A. Smith
Wyle Information Systems, Cleveland, Ohio

December 2010

NASA STI Program . . . in Profile

NASA/TM—2010-216909

GLobal Integrated Design Environment (GLIDE): A Concurrent Engineering Application

Melissa L. McGuire
Glenn Research Center, Cleveland, Ohio

Matthew R. Kunkel and David A. Smith
Wyle Information Systems, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

December 2010

Acknowledgments

The authors wish to express their deep thanks to Leon Gefert for creating GLIDE 1.0 in response to engineering analysis team need to transfer data in a clean and organized fashion. The authors also wish to thank Maria Babula, Branch Chief of the Mission Design and Analysis Branch, for supporting the development of GLIDE 2.0 and championing its use within GRC. Special thanks to Steven Oleson, the technical lead of the COMPASS team, for supporting the development of and use of GLIDE 2.0 in the ongoing studies of the COMPASS Team.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://gltrs.grc.nasa.gov

GLobal Integrated Design Environment (GLIDE): A Concurrent Engineering Application

Melissa L. McGuire
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Matthew R. Kunkel and David A. Smith
Wyle Information Systems
Cleveland, Ohio 44135

Abstract

The GLobal Integrated Design Environment (GLIDE) is a client-server software application purpose-built to mitigate issues associated with real time data sharing in concurrent engineering environments and to facilitate discipline-to-discipline interaction between multiple engineers and researchers. GLIDE is implemented in multiple programming languages utilizing standardized web protocols to enable secure parameter data sharing between engineers and researchers across the Internet in closed and/or widely distributed working environments. A well defined, HyperText Transfer Protocol (HTTP) based Application Programming Interface (API) to the GLIDE client/server environment enables users to interact with GLIDE, and each other, within common and familiar tools. One such common tool, Microsoft Excel (Microsoft Corporation), paired with its add-in API for GLIDE, is discussed in this paper. The top-level examples given demonstrate how this interface improves the efficiency of the design process of a concurrent engineering study while reducing potential errors associated with manually sharing information between study participants.

Introduction

Concurrent engineering is a process that involves multiple engineers and/or researchers designing a product (Ref. 1 to 6). Concurrent engineering studies offer advantages over non-parallel engineering strategies in that every aspect of the overall product design is considered simultaneously. When done correctly, this simultaneous approach reduces overall product design time. Since numerous engineers are engaged in the design cycle at the same time, concurrent engineering fosters an overall collaborative development where multiple experts (or teams of experts) are focused on the same product at the same time. While concurrent engineering is a very useful technique when used properly, a central concern for the process is how key information and data are shared between participants in a design cycle. The issue is of central importance when data and calculations from one discipline are dependent on data and calculations from another discipline. Sharing data parameters and information between participations in a concurrent engineering process must be quick, reliable and often must be secure.

The GLobal Integrated Design (GLIDE), Figure 1, is a client/server/API engineering application designed to facilitate multi-discipline, concurrent engineering study sessions by providing a convenient, reliable, fast (near real-time) and secure environment for researchers and engineers to share data and the results of calculations with each other. GLIDE utilizes common web technologies (Apache, HTTP, php) for client-server interaction making it relatively easy to install, use and maintain. The API for GLIDE requires applications to pass data into the environment with HTTP "put" commands and receive data through HTTP "get" commands. Given the web server-like appearance for the API, GLIDE's user interface is common, versatile and affords users the ability to interact with GLIDE through common front ends. GLIDE case studies in this paper focus on one particular API consisting of a Visual Basic for Applications (VBA) plug-in for Excel. This plug-in essentially makes Excel the user front end to GLIDE.

Figure 1.—The GLIDE Logo and acronym.

The purpose of this paper is to introduce GLIDE as a concurrent engineering tool and to detail its technical composition. A brief history of GLIDE is presented as a motivation for the tool development. Once introduced, case studies from real world studies using GLIDE at NASA Glenn Research Center (GRC) are presented to illustrate its usefulness and its user interface through Excel.

Nomenclature

API	Application Programming Interface
APL	Applied Physics Lab
ARC	NASA Ames Research Center
CAD	computer aided design
COMPASS	COncurrent Multidisciplinary Preliminary Assessment of Space Systems
DLL	Dynamic Link Libraries
DOE	Department of Energy
GLIDE	GLobal Integrated Design Environment
GRC	NASA Glenn Research Center
GSFC	NASA Goddard Space Flight Center
GUI	graphic user interface
Gzip	GNU zip
HTTP	HyperText Transfer Protocol
ICP	Intramural Call for Proposals
JPL	NASA Jet Propulsion Laboratory
JSC	NASA Johnson Space Center
LAN	local area network
LaRC	NASA Langley Research Center
LSMD	Laboratory for Spacecraft and Mission Design
MEL	Master Equipment List
MySQL	My Structure Query Language
NASA	National Aeronautics and Space Administration
NEDT	NASA Engineering Design Team
OS X	Apple's Operating System "X"
PHP	Hypertext Preprocessor (originally personal home page)
SEP	Solar Electric Propulsion
SQL	Structured Query Language
SSL	secure sockets layer
TTPO	Technology Transfer and Partnership Office
VBA	Visual Basic for Applications
VM	virtual machine
XML	Extensible Markup Language

Background and GLIDE History

GLIDE was originally developed at GRC for the In-Space Propulsion program to support multicenter, multidiscipline studies.

In 2005, as part of an Intramural Call for Proposals (ICP)-award project, GRC participated with many NASA centers—including the NASA Ames Research Center (ARC), Goddard Space Flight Center (GSFC), Jet Propulsion Laboratory (JPL), Johnson Space Center (JSC), and Langley Research Center (LaRC)—and the Aerospace Corporation, an industry partner, to form the NASA Engineering Design Team (NEDT) to examine designs for the 2022 Mars Sample Return Mission (Refs. 7 and 8). As a concurrent study, NEDT was faced with the challenge of exchanging information in a near real-time. To solve NEDT's issues with real-time information exchange, several existing technologies were investigated (including Modelcenter (Refs. 9 and 10) by Phoenix Integration, ICEMaker (Ref. 11 and 12) developed by LSMD). While these technologies were found to be useful, they were limited. After visiting Team X, JPL's concurrent engineering team, and attempting to integrate the Team X tools into a collaborative design team across the agency, the need for a tool that could operate both synchronously and asynchronously across firewalls was identified. Issues identified and solved using the GLIDE application were limited access due to firewalls (NASA, DOE, and other partners), lack of asynchronous data availability provided by tools available at the time, difficult NASA wide scalability, both cost and technological, cross platform (Macintosh (Apple, Inc.) and Microsoft Windows (Microsoft Corporation)) capability and ease of use for integration to current engineering tools.

To address these issues, a prototype for GLIDE was developed and used to share data directly from within Excel workbooks across disciplines. Figure 2 shows the infrastructure data flow and protocol used by the NEDT. GLIDE was selected and used as the preferred parameter-exchange solution by ARC, GRC, GSFC, and JSC. LaRC also demonstrated parameter exchange using GLIDE, but LaRC's role within the Mars sample return study did not require parameter exchange. The green dotted lines show where the GLIDE application via HTTPS was used, and the red lines showed where the file sharing for data exchange methodology was used.

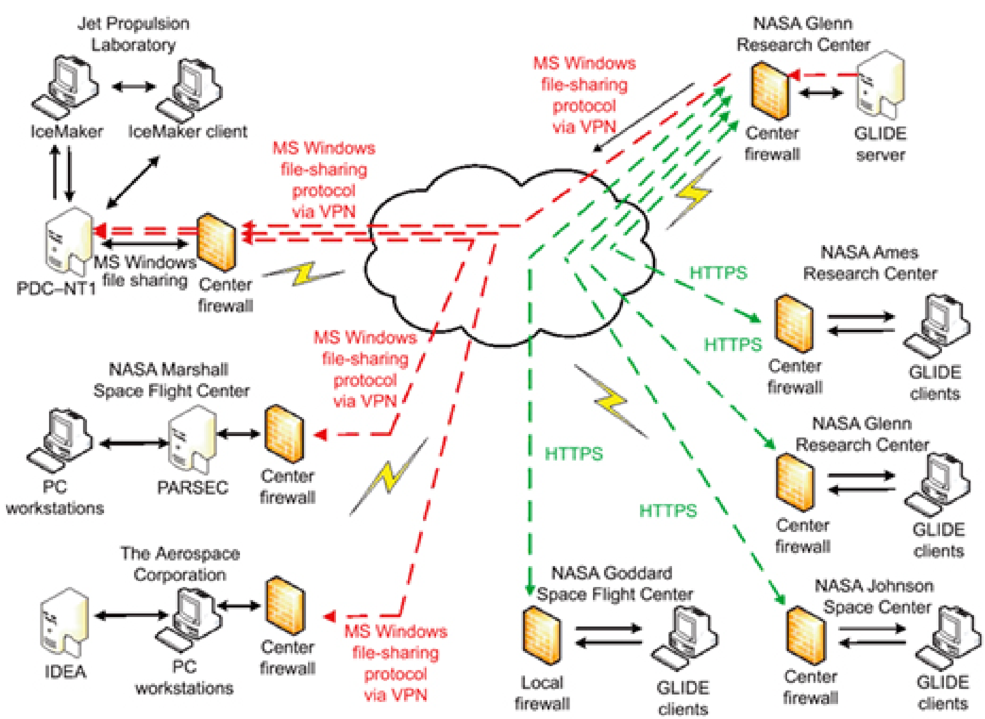


Figure 2.—The GLIDE Client-Server paradigm as applied to NEDT.

Since its inception in 2005, GLIDE, and its Excel API interface have been refined and now function as the primary concurrent engineering application tool for the GRC concurrent engineering design team, Concurrent Multidisciplinary Preliminary Assessment of Space Systems (COMPASS). The COMPASS team typically designs complex aerospace systems requiring specific engineering expertise from multiple disciplines and operates in a fashion similar to JPL's Team X (Ref. 13) or GSFCs IDC (Ref. 14). These disciplines, such as Avionics, Communications, Power, Propulsion, Structures, and Thermal are called upon in near real-time to contribute their expertise to size their subsystem. These subsystems are then tied together at the system integration level to build the entire system in answer to a customer request. Some of these customers include Exploration Systems Architecture Study designs for Ares V, In-space designs using electric propulsion thrusters, lunar communications satellites and ground assets, and recent studies for the committee conducting the Review of U.S. Human Space Flight Plans (aka Augustine Committee). The COMPASS team has used GLIDE to pass data in order to enable fast, complex, concurrent design studies. The customer is often present during the design session. The interconnectedness of each discipline's role to the overall system design results in an iterative concurrent engineering process.

What GLIDE Is

GLIDE is data exchange tool that works by populating a MySQL database through a publish and subscribe methodology. It is secure in that Data is sent via secure SSL encryption protocol, and as such is capable of securely crossing firewalls. It is capable of asynchronous and synchronous operation. GLIDE was designed to be a tool that includes engineers and analysts and their discipline tools in the design process. Because most engineering discipline experts are familiar with Excel, GLIDE uses Excel as a front end graphic user interface (GUI) to request/subscribe Name/Value pairs-based data in a database to facilitate an organized passing of data between disciplines. As an Excel aware application, GLIDE can take advantage of Solver in Excel or tailored optimization routines written in Visual Basic. It can also utilize Excel's ability to import Dynamic Link Libraries (DLLs) or read in Fortran and C files (on both Macintosh and Microsoft Windows platforms) to tie data from text input/output files through GLIDE to the database for data transfer. It was written as a cross platform compatible application on both Windows and Macintosh OS X platforms. GLIDE enables the identification of data holes and the assignment of required data generation. Data can be generated with outside tools, codes, etc and fed back through GLIDE via the Excel interface or with cell-based arithmetic or VBA based analysis within Excel.

What GLIDE Is Not

GLIDE is not an analysis tool with any inherent analytical expressions or engineering optimization capabilities. It is not tied to any specific type of modeling or methodology, except the Master Equipment List (MEL) process used by COMPASS. GLIDE is not limited to types of systems in a design such a spacecraft or other space related designs such as done by the COMPASS team. GLIDE does not require additional software installation on most engineers' machines, other than the client and Excel add-in, as most engineers have Microsoft products as part of the standard suite of tools installed on their computers. GLIDE does not have steep learning curve since most engineers are familiar with Excel.

GLIDE Technical Description

GLIDE is written as a compilation of several programming languages, including REALbasic developed by REAL Software, Inc. (Ref. 15), PHP (Ref. 16), and VBA. GLIDE client installers are available to download for both Microsoft Windows and Macintosh systems. The GLIDE client software is compatible with Excel 2000 or later on Windows systems and with Excel X and 2004 on Macintosh systems. Currently GLIDE is only available to users at GRC, where it is being beta tested, but is in the process for release through the GRC Technology Transfer and Partnership Office (TTPO).

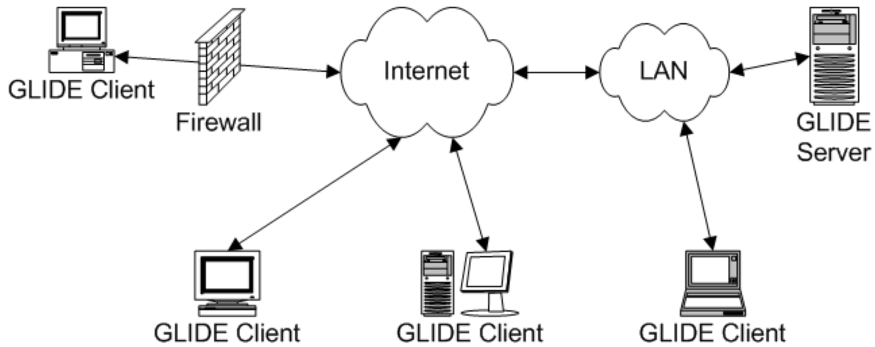


Figure 3.—The GLIDE Client-Server paradigm.

GLIDE follows the Client-Server paradigm as shown graphically in Figure 3, transferring encrypted and compressed data via standard web protocols. Currently, engineers use Excel as a front-end to the GLIDE Client as many of their custom tools run in Excel. As such, GLIDE is not a stand-alone engineering model but an interface to various subsystem engineering models, each of which is appropriately chosen on a case-by-case basis. Consequently, the inputs linked by GLIDE change with every design.

GLIDE Server

The GLIDE Server was designed to run on a standard web-server configuration, specifically: Linux, Apache (The Apache Software Foundation (Ref. 16)), MySQL (MySQL AB), and PHP. Both Linux and Apache are often utilized together to provide a reliable applications base. PHP: Hypertext Preprocessor, is an interpreted scripting language that operates well with MySQL, an open-source SQL database server. On the GLIDE Server, PHP scripts handle all GLIDE Client communication, storing datasets in the MySQL database. A Jabber server is also part of the GLIDE Server's configuration, which allows for the dynamic creation of chat rooms for each design session to facilitate the design process.

Server Background and Description

The GLIDE Server is based on Apache and uses PHP scripts to process requests from the Client. This design allows users to connect from any computer with Internet access, either directly or through a proxy. The Client to Server protocol consists of HTTP-POST data sent over SSL. A POST message is a type of HTTP request message. The Server responds with GZip compressed data wrapped in Extensible Markup Language (XML). The parameter datasets are formatted in new-line separated rows of tab-delimited fields, a format chosen for its low overhead and native compatibility with Excel. The Server generates the returned dataset using the most recently submitted set of values, however, all values submitted are retained, thereby providing version support. The ability to revert to a previous point can save time in a design session in which the dataset diverges continually and the final answer is not reaching convergence. The engineering lead can decide to choose a different dataset of trade space iteration in order to converge on a closed solution to the design problem. It also allows for spin-off studies based on previous work to be started very quickly, in as short as minutes.

The Server Side API has been written such that any application capable of submitting HTTP Forms using POST variables over SSL, which is also able to unzip and parse XML, can be configured to act as a GLIDE Client with a minimum amount of programming required. In the event that an application is not capable of the encryption, decompression, or XML parsing requirements, that application might still be able to connect to the server via the Client Side API. Figure 4 shows the server and client API concept in graphical representation, highlighting the ability for the GLIDE client data transfer to pass securely through firewalls.

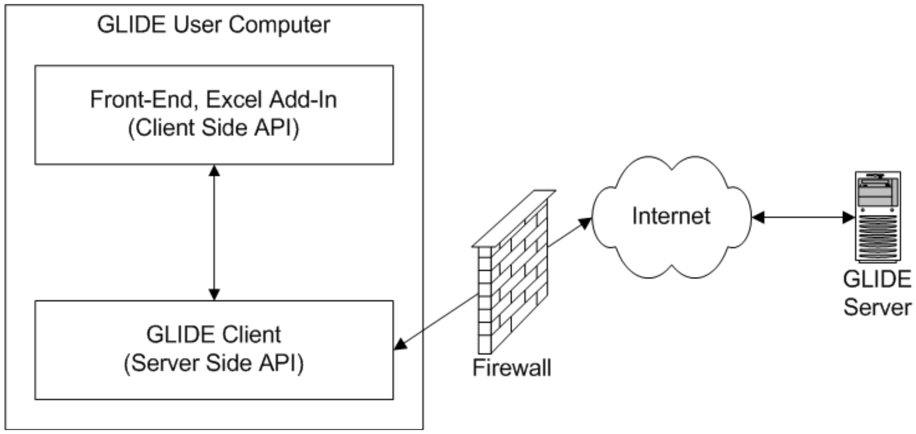


Figure 4.—GLIDE server and client API illustration.

GLIDE Client

The GLIDE Client was implemented in REALBasic to provide support for Windows, OS X, and Linux through the compilation of platform-specific binaries from a universal set of source code. The GLIDE Client Tool is a client side application with the primary job of acting as data conduit between the GLIDE server and GLIDE enabled Excel workbooks. The Client tool performs this job by providing: user authentication, data publishing, and data subscribing and available data queries. Additional services provided by the Client Tool include: data management, reporting queries history of datasets, and data age reporting. In addition to passing engineering datasets between the GLIDE Server and the Front-End, the Client also incorporates a proprietary Jabber client and other tools to view design session information and status.

Tracking Progress With the Client

A data progress graph can be defined on a per-session basis to enable real-time auditing or tracking of key system parameters from within the Client application window. Custom graphs can be created, as shown in Figure 5, to track changes in one or more system parameters. There is also the beta testing of an internal to the client file sharing mechanism to allow for file transfers to and from the Server without having to allow for WebDAV mounted shares on the server itself. This method will allow for more secure transfer of files across a network and across firewalls. The client provides additional monitoring of the dataset information, such as, data age and user activity. This functionality enables the iterative process to continue, in the case where not all participants are located in the same facility. Authentication to the Server is handled from the Client via an SSL socket over which all data is transferred.

Client Window Display

The Client maintains responsibility for all server communication including encryption, decompression, and XML parsing. Excel, or any front-end, must configure design session specific settings and then can make standard HTTP GET requests to the Client’s internal web-server. The Client Side API has been written to allow Excel to send and receive data as efficiently as possible; Excel Macros running VBScript are sometimes slow in execution time. This implementation gives Excel the ability to use the built-in External Data tool to perform a Web Query and insert the returned data directly into cells on a spreadsheet. The Server responds with the data format that Excel expects, thereby eliminating any need for dataset processing on the Excel side. Figure 6 shows some of the data available via the GLIDE Client window. In this case, the user can see which parameters have been input and by which users, as well as additional data about those parameters. As the GLIDE Client evolved based on user requests, additional information has and will be incorporated.

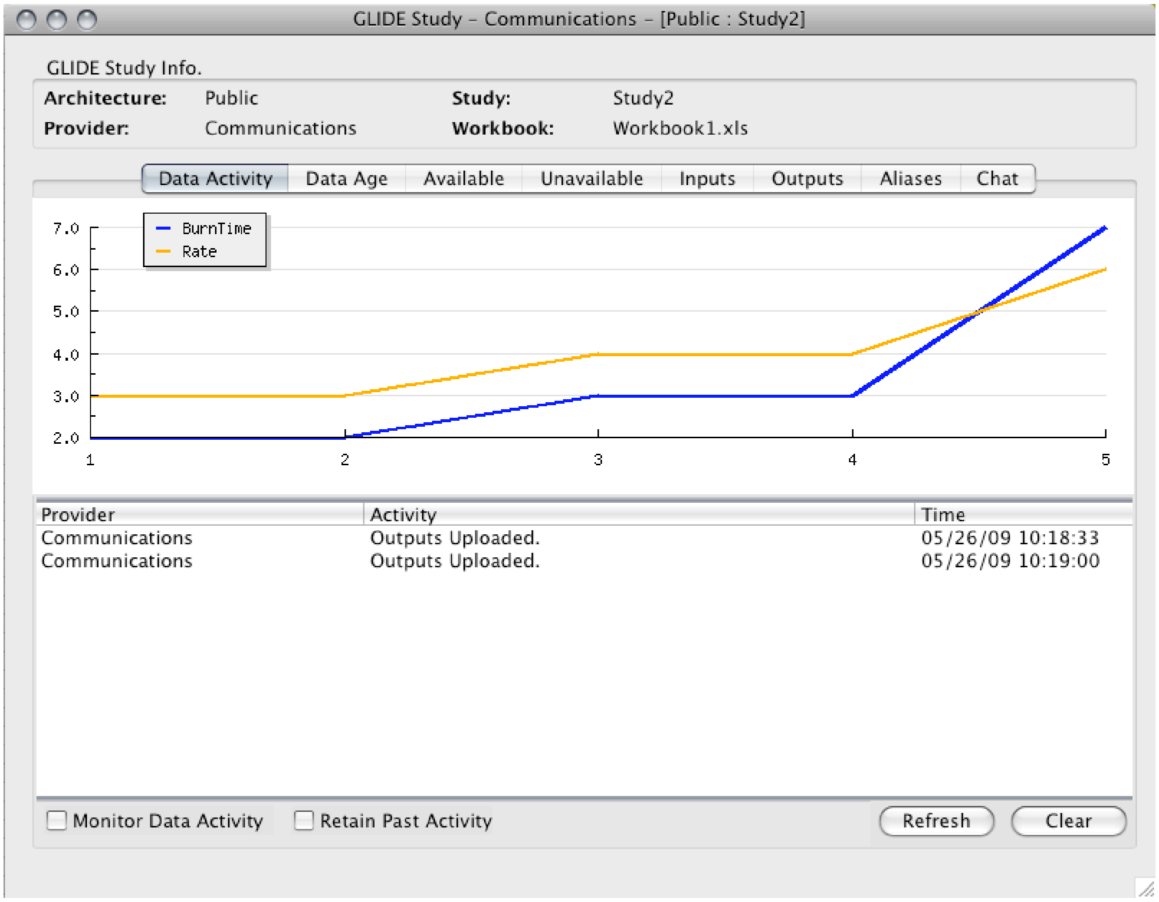


Figure 5.—GLIDE Client Data Tracking Custom Graph Function example.

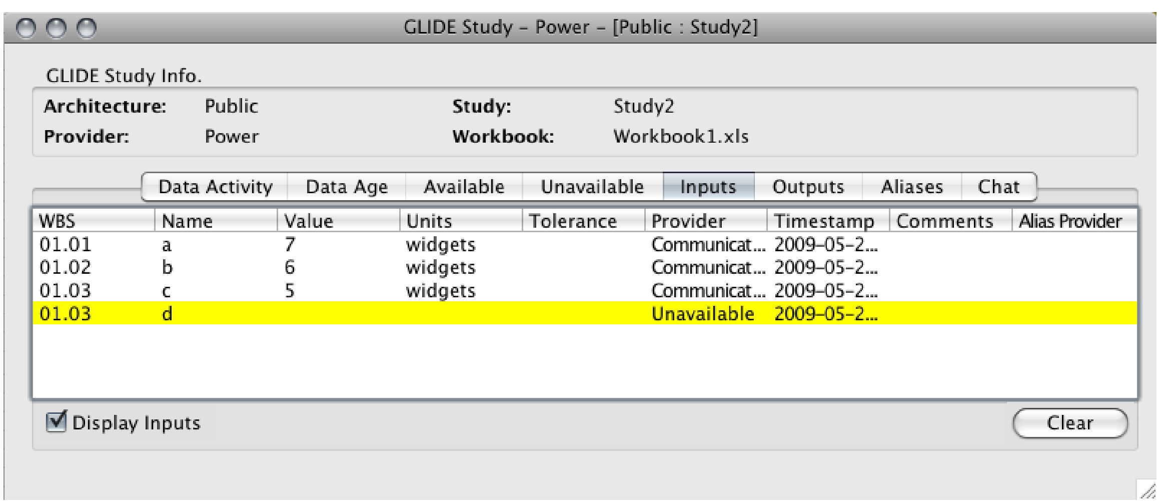


Figure 6.—GLIDE Client Window Display example.

GLIDE and Its Use in the Design Process

Generally, a concurrent design session consists of two or more engineers representing each discipline meeting together in a single location, or remotely in separate locations via teleconference interface. The lead discipline engineers exchange parameters and iterate through their respective processes to converge on an acceptable dataset. In cases in which the engineers are unable to meet, their parameters are passed via email, telephone, facsimile, or even postal mail. The result of this slow process of data exchange would elongate a design session to weeks or even months. Often one subsystem provides designs based on out-dated information on the other subsystems.

With the rise of ubiquitous Internet access, it is a logical step forward to utilize technology to real-time and streamline this process. While the iterative process remains in place, software can now exchange parameters securely and efficiently, while at the same time allowing for much more information about a design session to be made available.

Most Engineers are familiar with and have detailed top-level design tools created in Microsoft's Excel application. In addition, since Excel is available on both the Windows and Macintosh platform, rather than recreate the front end GUI with which to view the data to be passed, the developers decided to build an interface between the GLIDE server and GLIDE database and Excel. This involved the creation of an Excel add-in. Figure 7 illustrates how the COMPASS concurrent engineering design team currently uses the GLIDE server, the GLIDE client, and the GLIDE WebDAV shares in study sessions. The GLIDE server uses a virtual machine (VM) set up to allow for WebDAV shares on one VM and the database on a second VM. The GLIDE client tool and GLIDE enabled Excel workbooks on the user's computer interface with this server.

A typical design session uses the GLIDE application to tie together various subsystem engineering experts. In a COMPASS session, those experts include Propulsion systems, power system, communications systems, structures and mechanisms, Thermal protection and avionics systems engineers as well as trajectory and other mission analysis experts. This iterative and concurrent engineering relationship is shown in Figure 8. The subsystem leads use the GLIDE application and framework to tie together their analysis of a single design. The system integration lead engineer iterates with that data until a design has reached closure (for example, propellant tanks are sized to hold propellant necessary to perform a mission) during an iteration loop.

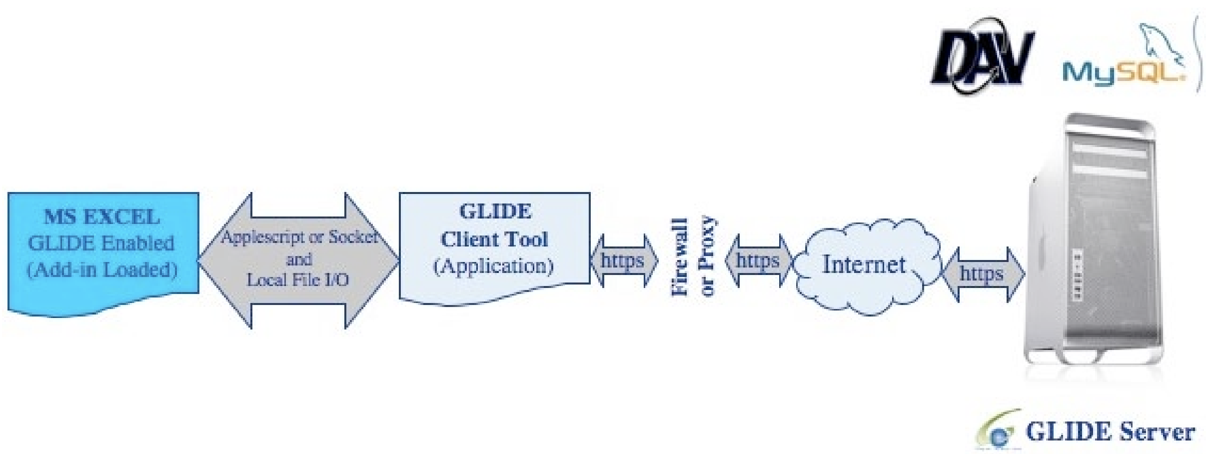


Figure 7.—GLIDE Server, Client, and Share structure.

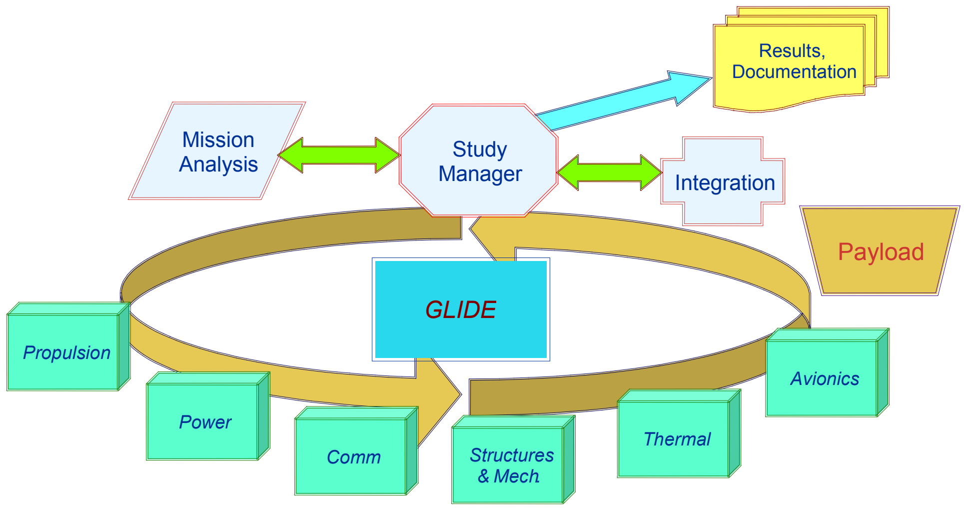


Figure 8.—COMPASS team structure and GLIDE application usage.

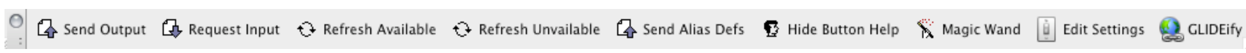


Figure 9.—GLIDE application Microsoft Excel toolbar.

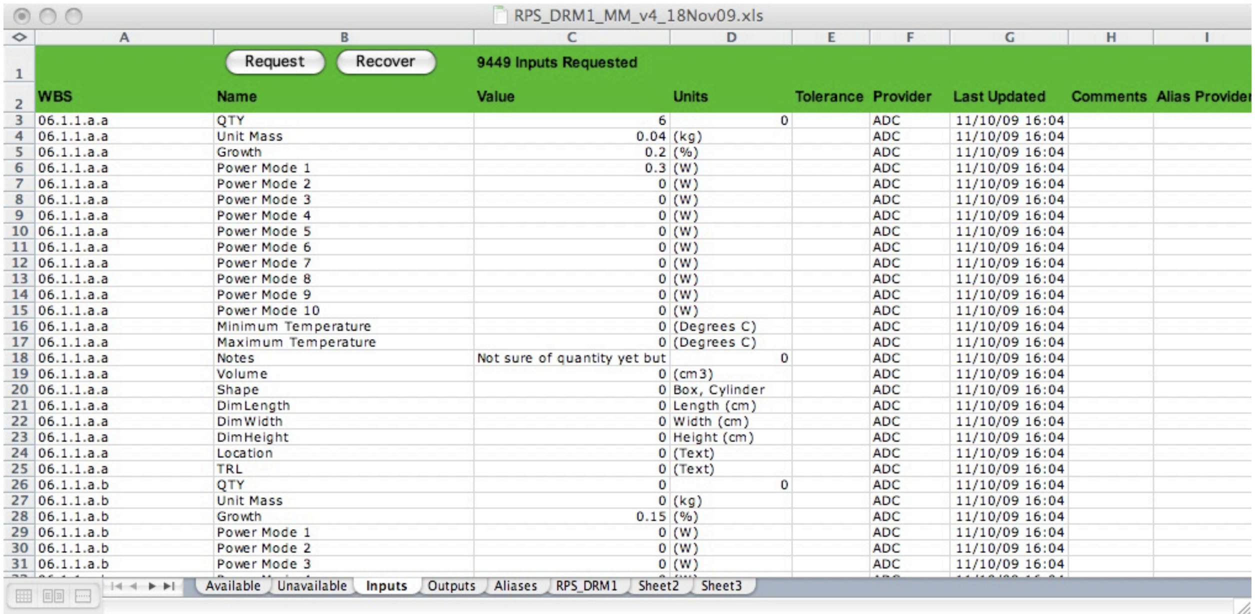
Microsoft Excel Add-In

Most users prefer to use software with which they are already familiar to participate in GLIDE design sessions. For this reason, the current front-end to GLIDE is Excel with the help of an add-in. Once the add-in has been installed to the user's Excel application, the toolbar and its functionality will become available. The toolbar, shown in Figure 9, provides GLIDE's core functionality to Excel: the ability for the user to view parameters, send output, and refresh input.

Upon a user's request to view available parameters in Excel, the add-in requests data from the Client, the Client then requests the data from the Server, and responds to the Excel add-in with the result returned from the Server. Custom Application Programming Interfaces between both the Front-End and the Client, and the Client and the Server, allow integration to other engineering applications as needed. The Front-End API can be implemented in an application without the need to account for authentication to the server or the ability to decrypt or decompress the data received from the Server.

Parameter Inputs and Outputs

GLIDE-ifying an Excel workbook adds in five additional worksheets to the workbook: Available, Unavailable, Inputs, Outputs and Aliases. The two worksheets that will be used most often by the engineer to transfer and receive data are the Inputs and Ouputs sheets. As should be obvious, the Inputs tab (sheet as shown in Fig. 10) is a worksheet set up to gather all of the variable names for which the engineer needs the associated data. The Outputs tab (or sheet) is where the discipline engineer places the variables for which they are publishing the associated values.



RPS_DRM1_MM_v4_18Nov09.xls

	A	B	C	D	E	F	G	H	I
1		Request Recover	9449 Inputs Requested						
2	WBS	Name	Value	Units	Tolerance	Provider	Last Updated	Comments	Alias Provider
3	06.1.1.a.a	QTY	6	0		ADC	11/10/09 16:04		
4	06.1.1.a.a	Unit Mass	0.04	(kg)		ADC	11/10/09 16:04		
5	06.1.1.a.a	Growth	0.2	(%)		ADC	11/10/09 16:04		
6	06.1.1.a.a	Power Mode 1	0.3	(W)		ADC	11/10/09 16:04		
7	06.1.1.a.a	Power Mode 2	0	(W)		ADC	11/10/09 16:04		
8	06.1.1.a.a	Power Mode 3	0	(W)		ADC	11/10/09 16:04		
9	06.1.1.a.a	Power Mode 4	0	(W)		ADC	11/10/09 16:04		
10	06.1.1.a.a	Power Mode 5	0	(W)		ADC	11/10/09 16:04		
11	06.1.1.a.a	Power Mode 6	0	(W)		ADC	11/10/09 16:04		
12	06.1.1.a.a	Power Mode 7	0	(W)		ADC	11/10/09 16:04		
13	06.1.1.a.a	Power Mode 8	0	(W)		ADC	11/10/09 16:04		
14	06.1.1.a.a	Power Mode 9	0	(W)		ADC	11/10/09 16:04		
15	06.1.1.a.a	Power Mode 10	0	(W)		ADC	11/10/09 16:04		
16	06.1.1.a.a	Minimum Temperature	0	(Degrees C)		ADC	11/10/09 16:04		
17	06.1.1.a.a	Maximum Temperature	0	(Degrees C)		ADC	11/10/09 16:04		
18	06.1.1.a.a	Notes	Not sure of quantity yet but	0		ADC	11/10/09 16:04		
19	06.1.1.a.a	Volume	0	(cm3)		ADC	11/10/09 16:04		
20	06.1.1.a.a	Shape	0	Box, Cylinder		ADC	11/10/09 16:04		
21	06.1.1.a.a	DimLength	0	Length (cm)		ADC	11/10/09 16:04		
22	06.1.1.a.a	DimWidth	0	Width (cm)		ADC	11/10/09 16:04		
23	06.1.1.a.a	DimHeight	0	Height (cm)		ADC	11/10/09 16:04		
24	06.1.1.a.a	Location	0	(Text)		ADC	11/10/09 16:04		
25	06.1.1.a.a	TRL	0	(Text)		ADC	11/10/09 16:04		
26	06.1.1.a.b	QTY	0	0		ADC	11/10/09 16:04		
27	06.1.1.a.b	Unit Mass	0	(kg)		ADC	11/10/09 16:04		
28	06.1.1.a.b	Growth	0.15	(%)		ADC	11/10/09 16:04		
29	06.1.1.a.b	Power Mode 1	0	(W)		ADC	11/10/09 16:04		
30	06.1.1.a.b	Power Mode 2	0	(W)		ADC	11/10/09 16:04		
31	06.1.1.a.b	Power Mode 3	0	(W)		ADC	11/10/09 16:04		

Available Unavailable Inputs Outputs Aliases RPS_DRM1 Sheet2 Sheet3

Figure 10.—GLIDE-ified Microsoft Excel workbook input sheet.

Example Case Study: Mercury SEP Stage

At the time of the writing of this paper, the COMPASS team is currently designing a Solar Electric Propulsion (SEP) stage to send a yet to be designed Lander to Mercury as part of the Decadal survey planning done by NASA's Science Mission Directorate. The COMPASS design team consists of engineers from multiple disciplines (Power, Propulsion, Mission design, Thermal, Communications, computer aided design (CAD), etc) and organization codes across GRC. The team members meet in the COMPASS lab during the thrice-weekly study sessions, or tie in from remote locations via a teleconference number. The GLIDE application allows the engineers to publish and subscribe to the data from their desks. Even though they cannot be present, the portable and cross-internet nature of GLIDE allows engineers to actively participate in a live design session.

The ACE lab at the Applied Physics Lab (APL) (Ref. 17) is designing the Lander. The APL and COMPASS teams are interfacing their two designs via the sharing of requirements; design decisions and top level Master Equipment Lists (MELs) generated using file exchange, teleconferences and videoconferences. It is planned in the future to link the two teams together through the use of the GLIDE application and sever in order to provide an end-to-end design in real time from two physically separate engineering design teams.

A past example of a COMPASS design, which used the GLIDE server and client, is documented in Reference 18 and involved the design of a Near Earth Asteroid sample return spacecraft. The customer brought the requirements of the study, and the COMPASS team tied together the relevant discipline leads via a MEL interface built in Excel to design the spacecraft. This design and the flexibility of GLIDE was used to allow the team to perform trades through multiple cases in which different propulsion system options were tested and compared.

Other Uses of the GLIDE Application

Other potential users of the GLIDE application are any concurrent design engineering or other similar entity, which would have the need to link together, multiple mathematical or logical data sets (Ref. 19). There are a number of concurrent engineering teams like COMPASS at NASA and any one of them could apply the GLIDE model to data passing (GSFC IDC (Ref. 14), Team X (Ref. 13), LaRC IDC (Refs. 20 to 22)). GLIDE could easily be integrated either inside the current operating practices of these teams or be used to link the teams remotely to perform concurrent engineering design sessions across these multiple centers.

Up to now, GLIDE has been used within the framework of designing a spacecraft or other space related system. While this has proven a very efficient way to pass data for the spacecraft design teams, there is no reason that the data has to be spacecraft in nature. GLIDE could pass any data that is set up using a parameter/value methodology. Current applications under study include automation of management practices and automation of detailed communications system design.

Limitations of the Current GLIDE API

The current API used for GLIDE is Excel and therefore any and all limitations inherent in Excel also limit the client side usage of GLIDE. One limitation on the transfer of data is also tied to Excels use of a 255 characters per cell limit, as well as limit in number of lines and columns that Excel can handle in any workbook. In addition, as Microsoft changes versions of Excel, the add-ins and formatting of the Excel connection to GLIDE have to be revisited. The GLIDE developers have to be cognizant of how changes in the VBA implementation will affect the ability of the GLIDE add-in to perform its functions. Recently, Microsoft has dropped support of Virtual Basic in the Macintosh OS X version of Excel 2008, and has no official plans to replace it. For this reason, all Macintosh users have to continue to use either Office X or 2004 in order to run GLIDE. Future versions of GLIDE will continue to support both the Macintosh and the Windows platforms, so work is ongoing to find a workaround for the Excel add-in API on the Macintosh solution.

Since GLIDE was written as a parameter passing solution, it is also not suited to the transfer of files such as CAD or other engineering files. It could be used to pass data about those files but not the files themselves. There are no plans to change GLIDE into a file passing application at this time.

Conclusion

The ability of GLIDE to tie together multiple participants who are both collocated and/or dispersed in remote duty stations, has given the COMPASS team the ability to quickly perform designs of relatively complex space systems. Removing potential human error in data passing by automating the process has increased the reliability and accuracy of the data produced in the COMPASS design sessions. Without the GLIDE application, these design sessions would take much more time making multiple trade space iterations impossible in the limited amount of time currently allocated to such tasks.

GLIDE uses a common Web-based MySQL server that is accessed using PHP scripts (the PHP Group). MySQL is a relational database management system, available as an open source system. GLIDE's Web-served database allows secure and controlled access to design data by using firewall-friendly secure sockets-layer- (SSL-) based user authentication. Data can be queried and published to the GLIDE database via a Excel interface. The training of GLIDE users is relatively effortless since most are familiar with Excel.

Availability: The GLIDE development team is currently working with the GRC TTPO to release GLIDE to other government and academic institutions. It is hoped that this process will be completed in the near future.

References

1. Smith, Robert P., "The Historical Roots of Concurrent Engineering Fundamentals," IEEE Transactions on Engineering Management, Vol. 44, No. 1, Feb. 1997.
2. Mark, Gloria; Abrams, Steve; and Nassif, Nayla: "Group-to-Group Distance Collaboration: Examining the "Space Between"," Proceedings of the 8th European Conference of Computer-supported Cooperative Work (ECSCW'03), Sep. 14–18, 2003, Helsinki, Finland.

3. Fulvio E. Oliveto, “Concurrent Engineering: Evolution and Application,” 0-7803-6262-4/00 © 2000 IEEE.
4. “A New Way of Doing Business—Concurrent Engineering,” Design News, Vol. 53, No. 17, p. 34, Sep. 7, 2998.
5. Oxnevad, K.I., “The NPDT—The Next Generation Concurrent Design Approach,” Proceedings of 2nd European Systems Engineering Conference (EuSEC 2000), Munich, Germany, Sep. 13–15, 2000.
6. DelRosario, Ruben; Davis, Jose M.; Keys, L. Ken, “Concurrent and Collaborative Engineering Implementation in an R and D Organization,” 2003 IEEE International Engineering Management Conference: Managing Technologically-Driven Organizations; Nov. 1–3, 2003; Albany, NY.
7. *Glenn Research Center at Lewis Field 2005 Research and Technology*, NASA/TM—2006-214016, pp. 6–7.
8. Falck, Robert D.; and Burke, Laura M.: “XLerator Software Released.” Glenn Research Center at Lewis Field 2005 Research and Technology , NASA/TM—2006-214016, 2006, pp. 5–6. http://www.grc.nasa.gov/WWW/RT/2005/PB/PBM-falck.html.
9. Modelcenter developed by Phoneix Integration. http://www.phoenix-int.com/software/phx_model center.php .
10. Taylor, J.L.; Neely, M.A.; Curran, F.M.; Christensen, E.R.; Escher, D.; Lovell, N.; Morris, Charles (Technical Monitor), “Integrated Technology Assessment Center (ITAC) Update,” AIAA–2002–3550, 8th AIAA/ASME/SAE/ASEE Joint Propulsion Conference; Jul. 7–10, 2002; Indianapolis, IN.
11. LSMD (Laboratory for Spacecraft and Mission Design): ICEMaker, http://www.lsmd.caltech.edu/tools/icemaker/icemaker.php.
12. Schuman, Todd; de Weck, Olivier, L.; Sobieski, Jaroslaw: “Integrated System-Level Optimization for Concurrent Engineering with Parametric Subsystem Modeling,” 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference; Apr. 18–21, 2005; Austin, TX.
13. Jet Propulsion Laboratory Team X homepage: http://jplteamx.jpl.nasa.gov/.
14. NASA Goddard Spaceflight Center Integrated Design Center homepage: http://idc.nasa.gov/.
15. REALBasic software: http://www.realsoftware.com/realbasic/.
16. PHP reference: http://en.wikipedia.org/wiki/PHP.
17. The Apache Software Foundation: http://www.apache.org/.
18. The Johns Hopkins University Applied Physics Laboratory: http://www.jhuapl.edu/.
19. Oleson, Steven R.; and McGuire, Melissa L., “COMPASS Final Report: Near Earth Asteroids Rendezvous and Sample Earth Returns (NEARER),” NASA/TM—2009-215825; CD–2008–28.
20. GLIDE software release website: http://www.grc.nasa.gov/WWW/RT/2005/PB/PBM-gefert.html.
21. NASA Langley Research Center Integrated Design Center homepage. http://idc.larc.nasa.gov/.
22. Gough, Kerry M., Allen, B. Danette; and Amundsen, Ruth M.: “Collaborative Mission Design at NASA Langley Research Center,” GT–SSEC.A.3.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
01-12-2010	Technical Memorandum	

4. TITLE AND SUBTITLE
GLobal Integrated Design Environment (GLIDE): A Concurrent Engineering Application

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
McGuire, Melissa, L.; Kunkel, Matthew, R.; Smith, David, A.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
WBS 432938.11.01.03.01.02.39

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-17183

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TM-2010-216909

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Categories: 61, 66, and 62
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 443-757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The GLobal Integrated Design Environment (GLIDE) is a client-server software application purpose-built to mitigate issues associated with real time data sharing in concurrent engineering environments and to facilitate discipline-to-discipline interaction between multiple engineers and researchers. GLIDE is implemented in multiple programming languages utilizing standardized web protocols to enable secure parameter data sharing between engineers and researchers across the Internet in closed and/or widely distributed working environments. A well defined, HyperText Transfer Protocol (HTTP) based Application Programming Interface (API) to the GLIDE client/server environment enables users to interact with GLIDE, and each other, within common and familiar tools. One such common tool, Microsoft Excel (Microsoft Corporation), paired with its add-in API for GLIDE, is discussed in this paper. The top-level examples given demonstrate how this interface improves the efficiency of the design process of a concurrent engineering study while reducing potential errors associated with manually sharing information between study participants.

15. SUBJECT TERMS
Concurrent engineering; Real-time operation; Programming language; Application programming interface

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk (email:help@sti.nasa.gov)
U	U	U	UU	18	**19b. TELEPHONE NUMBER** *(include area code)* 443-757-5802

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18