DrawCraft: A Spacecraft Design Tool For Integrated Concurrent Engineering[1]

Shahram Moez Ardalan
Undergraduate in Chemistry
California Institute of Technology
MSC #147, Caltech
Pasadena, CA 91126
(626) 395-1713
sardalan@caltech.edu

Abstract—DrawCraft is a design productivity tool that quickly enables engineers to render preliminary spacecraft within the Computer Aided Drafting suite SolidWorks. User-friendly and easy to learn, DrawCraft assists the entire design process by using templates of the most commonly used spacecraft components, requiring minimal information from the user. All of the information necessary to fully define a spacecraft of some number (n) parts is kept in a datamatrix of n rows by 21 columns. The n by 21 datamatrix can be exported to a tab-delimited text file called the Shared Mechanical Control Sheet or the Products Attributes Database. DrawCraft can also import data from the PAD or an SMCS and then render the corresponding spacecraft. The importance of the SMCS is that many elements of the datamatrix are either common engineering parameters or are simply related to such parameters. Hence, DrawCraft can import design parameters from other tools such as trades models or design databases.

TABLE OF CONTENTS

1. INTRODUCTION

The Need

The process of developing a project from conception to launch is a long sequence of numerous steps. Normally, after the design team has determined which components and assemblies a spacecraft will require, designers must use Computer Aided Drafting (CAD) tools to draft each component of the spacecraft and put them together, even for relatively preliminary models. Any changes that must be made to the model have to be made within the CAD program by someone with sufficient knowledge of the usually complex software. This system is costly and time consuming.

Another major source of inefficiency in a mission design environment is the transfer of information among different teams. The actual process of recording data, delivering it to another team, and explaining it is redundant and time consuming. A powerful design tool should thus be able to communicate well over different media and import/export data from/to a simple format which each team can individually use equally well.

Previous Work

DrawCraft builds upon previous work conducted in the Laboratory for Spacecraft and Mission Design (LSMD). The task that the original programs needed to accomplish was to write a spreadsheet of parameters describing a spacecraft and then use the spreadsheet to render spacecraft in a CAD program. The first program SCDTWiz (Spacecraft Design Tool Wizard) was a Visual Basic macro for Microsoft Excel. It presented the user with a series of dialog boxes that prompted the user for dimensions and parameters describing a spacecraft. SCDTWiz then collected all of the inputted data and created a tab-delimited text file called the Shared Mechanical Control Sheet (SMCS). The second program Redraw was a macro for the low-end CAD suite MiniCAD [1]. Written in MiniCAD's personal programming language MiniPascal, Redraw was designed to use the data supplied by the SMCS and to render the corresponding spacecraft in MiniCAD. After rendering the spacecraft, the program terminated and the user was left in the MiniCAD environment to work with the open CAD file.

The combination of SCDTWiz and Redraw had a number of benefits and drawbacks. Both MiniCAD and Microsoft Excel are cross-platform, thus allowing a large market to take advantage of both SCDTWiz and Redraw. Unfortunately, MiniCAD is used by a relatively small market and is thus less practical for software development. Also, in order to use both programs in conjunction, the user had to run the SCDTWiz macro in MS Excel, copy the spreadsheet from Excel into a MiniCAD spreadsheet, and then manually launch the Redraw macro. While Redraw accomplished its original purpose of translating the SMCS into a MiniCAD model, any further development of the program was limited by the simplicity of MiniPascal. If the user made a change to the spacecraft model in MiniCAD, no changes would be shown in the SMCS. Furthermore, creating a spacecraft on the fly was impossible because each time the SMCS was altered, the Redraw macro had to be started from scratch. The functionality of MiniCAD and the programming capacity of MiniPascal were both too limiting to allow further development of the program. Without the ability to improve and expand, it was unrealistic to expect Redraw to become a powerful tool.

The Approach

The topic of this paper, DrawCraft, is an attempt to simplify and expedite the process of creating preliminary models of spacecraft. Although the models produced by DrawCraft are not detailed enough for fabrication or quantitative analysis, they are intended to be accurate enough for a wide variety of uses. DrawCraft also accomplishes the tasks of both of its predecessors, SCDTWiz and Redraw. Like SCDTWiz, DrawCraft presents a series of dialog boxes querying the user for spacecraft dimensions and attributes, and then creates an SMCS, organizing all of the provided data. Like Redraw, DrawCraft can interpret the data in an SMCS and render a spacecraft based on the data. However, DrawCraft can also utilize the data directly from the user or from the Products Attributes Database (PAD) [2]. The PAD is a single, network-accessible repository of all of the parametric information that describes a project at any phase of implementation.

In order to keep DrawCraft up to date and versatile, DrawCraft was written for use with the CAD suite SolidWorks [3]. SolidWorks is a powerful mid-ranged program and among the most widely used CAD suites on the market. SolidWorks is built on the Standard for the Exchange of Product (STEP) specifications and is therefore compatible with high-level desktop tools. SolidWorks also comes equipped with a detailed Application Programming Interface (API) for cooperative use with Microsoft Visual Basic. Also the language in which DrawCraft is written, Microsoft Visual Basic [4] provides an excellent programming environment. Both SolidWorks and Visual Basic are still being improved upon and undergo regular upgrades. As these two progress and add new features, so will DrawCraft, thus preventing DrawCraft from suffering its predecessors' fates. With three methods of inputting data and three methods of exporting data, DrawCraft will indeed become a useful tool in the design room.

This Paper

This paper provides a detailed summary and description of DrawCraft beginning with introductory material in Section 1. Section 2 describes the Shared Mechanical Control Sheet, the key to linking DrawCraft to other design and analysis packages. This is the data structure around which the program is written and organized. Section 3 is a short overview of the effective use of DrawCraft. A description of the internal structure and design of the program is included in Section 4. Section 5 discusses NetCraft, one of the further developments of DrawCraft for use on the World Wide Web. Finally, the current status and future plans for DrawCraft are outlined in Section 6.

2. THE SHARED MECHANICAL CONTROL SHEET

The Shared Mechanical Control Sheet (SMCS) is the basic data structure around which DrawCraft is written and organized. The SMCS takes on two forms in the DrawCraft environment. The first form is as a two-dimensional array of strings called DataMatrix. DataMatrix is loaded in DrawCraft's memory throughout the duration of the program and is referenced constantly during the rendering process. The exported instantiation of DataMatrix is a tab-delimited text file, which is referred to generically as the SMCS. While the program is running, the DataMatrix stores all of the specifications for each component in the current spacecraft. As a tab-delimited text file, the SMCS is the primary method of importing and exporting work in the DrawCraft environment and stores all of the necessary data to render a given spacecraft one component at a time. For each component, the SMCS holds four types of information: identification, geometric properties, location, and material properties. Using all of the data presented in these four categories, a component can be designed uniquely and placed without ambiguity. Although the current SMCS is only twenty-one columns wide, the SMCS may later be extended to 25 columns for added features without requiring serious reconstruction of the program.

Hierarchy of Parts

Every line of the SMCS describes a specific part within the spacecraft. Each line of the SMCS is independent of its position with respect to other lines, and hence, no particular order is necessary. However, in order to make the SMCS easier to use, DrawCraft reorganizes the lines into a certain order called the Standard Hierarchy of Parts. This hierarchy depicts the arrangement of the spacecraft's components in an outline-like data structure.

Main Bus
- Parts on the Main Bus
- Secondary Bus
  - Parts on the Secondary Bus
  - Tertiary Buses and their Parts
- Other Secondary Buses and their Parts

Figure 1 shows the format of the SMCS and the contents of each column. The first line of the SMCS will always be the Main Bus[2], and it is identified as the only line in the SMCS with Wall Number 0 and Part ID# 0. In SolidWorks, the Main Bus is the Main Assembly and has ownership of all other components in the spacecraft. The lines immediately

[2] Bus is synonymous with Chassis

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	**Component Name**	**Part ID#**	**Name of Parent Bus**	**Wall # on Parent Bus**	**Part Specific Parameters**									**Translation Vector**			**Orientation Vector**			**Mass**	**Fill Color**
2	Master Bus	0	-	0										-	-	-	-	-	-	Mass	Fill Color
3	Parts on the Master Bus	10-100	Master Bus	Wall # on Master Bus										x	y	z	x	y	z	Mass	Fill Color
4	Secondary Bus	0	Master Bus	Wall # on Master Bus										x	y	z	x	y	z	Mass	Fill Color
5	Parts on the Secondary Bus	10-100	Secondary Bus	Wall # on Secondary Bus										x	y	z	x	y	z	Mass	Fill Color
6	Other Buses	0	Any Bus	Wall # on Bus										x	y	z	x	y	z	Mass	Fill Color

Figure 1: General format of the SMCS

following the first line are for the non-bus parts that are directly attached to the Main Bus. These lines are recognized as having the name of the Main Bus as their Parent Bus, as shown in the third column of Figure 1. SolidWorks is given these parts as components of the Main Assembly. After these rows, the SMCS contains Secondary Buses that are attached to the Main Bus. While still having Part ID# 0, they must have non-zero Wall Numbers because DrawCraft must be told where on the Main Bus to place the Secondary Bus. In SolidWorks the Secondary Buses are Sub-Assemblies of the Main Assembly. Since they are the only other assemblies used by DrawCraft, buses are the only parts to which components may be attached.

For the purpose of flexibility, DrawCraft allows the spacecraft to have any combination of buses in series or parallel. This means that the user could create a spacecraft of just buses linked together in a chain, or the Main Bus could have twenty sides with a Secondary Bus attached to each one.

Reading a Line

Identification—The first four columns of the SMCS are for identification of the component. The first column contains the name of the component. This name, however, does not need to be indicative of what kind of part it is: e.g. Bob is as

Part	**Part ID#**	**Geometric Properties**					
Master Bus	0	Number of sides	Side length	Depth of bus	wall thickness	No ends, Top, Bottom, or Both (0,1,2,3)	-
Black Box	10	Length	Width	Depth	-	-	-
Tank	20	Radius	Length of cyl. section	Wall thickness	-	-	-
Solar Array	30	Boom length	Boom diameter	Array length	Array width	Thickness	Stowed config.
Thruster	40	Chamber length	Chamber radius	Throat length	Throat radius	Nozzle length	Exit radius
Antenna	50	Radius	Focal length	# of supports (0 for no boom)	Support thickness	Support angle (from axis)	-
Thruster Cluster	60	Cube side length	Throat diameter	Exit diameter	Nozzle length	Config type	-
Star Tracker	70	Side length	Depth of box	Cone length	Field of view angle	-	-
Cylinder	80	Radius	Length	-	-	-	-
Plate	90	Number of sides	Side length	Plate thickness	-	-	-
Strut	100	Radius	Length	-	-	-	-

Table 1: Geometric Properties for predefined parts

valid a name for a component as High Gain Antenna. The second column holds the component's Part ID#. The Part ID# is predefined for each kind of part and indicates to DrawCraft what kind of part the line contains. The Parent Bus Name and Wall Number immediately follow the Part ID#. Together, Parent Bus Name and Wall Number tell DrawCraft to which bus the component is attached and on which side of the bus it should be placed. The name of the Parent Bus must be entered exactly as it is typed in the first column of its own line. The Wall Numbers start with 1 on the front of the Bus in the SolidWorks Coordinate system and increase counterclockwise; but "Top" and "Bottom" are also valid Wall Numbers.

Geometric Properties—The next nine columns are allocated for the geometric attributes of each part. DrawCraft is capable of constructing eleven predefined parts based solely on these attributes. Table 1 is a table of the attributes for each predefined part. While currently no part requires more than six attributes, DrawCraft has been left with room for adding new parts, and future parts will have the option of using up to nine attributes. In order to maintain simplicity, the only attributes required for each part are currently essential to the part's basic structure.

Because the user may not be fully proficient in the writing of a valid SMCS, DrawCraft is equipped with an SMCSChecker. The SMCSChecker ensures that each entry in the SMCS is a valid parameter for the part it describes. If the user attempts to make a bus with 1.7 sides or a thruster with a larger throat diameter than chamber diameter, DrawCraft will open the SMCS in the default text editor and tell the user in what row and column the error occurs. DrawCraft is also capable of integrating custom objects into the spacecraft. These custom parts can be any SolidWorks object from either the user or a third party program. To make use of this feature, the user simply uses 1000 for the Part ID# and enters the custom object filename as the first geometric attribute. The user can then even place equations describing the geometry of the custom object in the other eight geometric attribute columns of the SMCS. For example, to set the SolidWorks dimension D1@Sketch1 in the custom object to 12 cm, the user would enter "Length=D1@Sketch1=.12" as a geometric property.

Location—The next six columns of the SMCS are designated for location data to properly place the component in position with respect to its Parent Bus. The first three entries combine to form the translation vector, and the second three entries combine to form the orientation vector. Both of these vectors are taken with respect to a unique coordinate system on the face of the wall to which the part is attached as shown in Figure 3. The origin of this coordinate system is the geometric center of the outer surface of the wall. The z-axis is defined to be normal to the wall and pointing outward from the Bus. The y-axis is vertical and perpendicular to the top of the Bus (except for the coordinate systems on the top and bottom), and the x-axis is given from the other two.

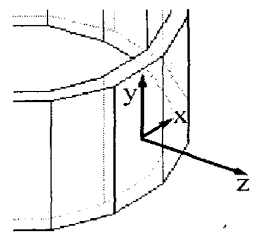

Figure 3: The coordinate system on the wall of a bus

Thus, the default translation vector [0,0,0] places the origin of the component at the center of the outer surface of the wall to which it is attached. The default orientation vector [0,0,1] lines up the main axis of the component with the z-axis of the wall: i.e. the component sticks straight out of the bus by default. In addition, one can place an object inside of the bus by using a translation vector with a negative z-component, and one can point any object into the bus by using a negative z-component in the orientation vector. While the orientation vector need not be a unit vector on input, it will always be outputted so.

Material Properties—The last two entries in each line of the SMCS are reserved for material properties. The first column contains the masses of each part in kilograms. While the density of a part may in some cases make more sense to use, few actual components have one continuous density. Also, the templates that DrawCraft uses to represent each of the components are not specifically realistic in volume, thus making the density meaningless. However, supplying the mass of each component individually does approximate the mass properties of the entire spacecraft with an appreciable degree of accuracy. The last column of the SMCS is simply the color that SolidWorks should use when rendering each component. This attribute allows for practical purposes such as color-coding separate systems as different colors or representing each material with a distinct color. On the other hand, the user can also simply paint the spacecraft pretty colors for the sake of aesthetics and self-satisfaction.

3. USE OF DRAWCRAFT

Getting Started

DrawCraft can be launched from within SolidWorks or as a stand-alone application, in which case DrawCraft automatically loads SolidWorks. Within DrawCraft, the user can either open an existing SMCS or start from scratch. When starting from scratch, DrawCraft allows the user to

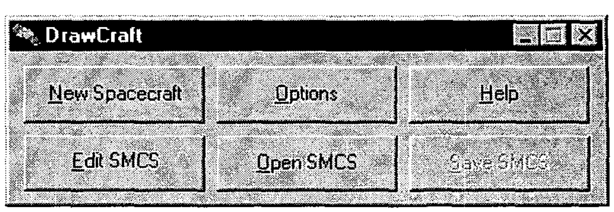

Figure 4: DrawCraft's Opening Menu

build a complicated spacecraft with little to no previous knowledge of DrawCraft or SolidWorks. While the user builds the model through a combination of the DrawCraft Tool Palette and the Part Options Dialog Boxes, DrawCraft automatically generates an SMCS and updates it with each addition or change. Figure 4 shows DrawCraft's Opening Menu. Using this menu, the user decides to either start a spacecraft from scratch or load a pre-existing SMCS.

If the user chooses to start a new spacecraft, he or she is presented with the Bus Options Dialog Box, Figure 6. These dimensions and parameters will be used to generate the Main Bus, the first part of every spacecraft and the only part that the program requires to be made. Once the user has finished entering the necessary data, DrawCraft renders the Main Bus in SolidWorks. At this point, DrawCraft and SolidWorks consider the spacecraft to be complete, and any additions or changes to the spacecraft are optional because a Main Bus by itself is considered to be a valid spacecraft. The user now has full artistic license to edit the spacecraft as he or she wishes via the Tools Palette.

To load an SMCS when starting DrawCraft, the user simply chooses "Open SMCS" from the Opening Menu. DrawCraft then queries the user for the SMCS to load and verifies the validity of the SMCS through the SMCSChecker. If the SMCS is approved, DrawCraft renders the corresponding spacecraft in SolidWorks. At this point, DrawCraft cannot distinguish whether the current spacecraft were made from scratch or built from an imported SMCS. As before, the user has full control of the program and may add on to or edit the spacecraft via DrawCraft or SolidWorks as he or she wishes.

The Tools Palette

The Tools Palette is DrawCraft's primary interface and houses most of the program's functionality. Shown in Figure 5, the palette is split into three sections with a total of twenty-two buttons. The first section contains the buttons for adding components, the second section of the palette consists of the buttons used for editing, and the third section holds the import/export/analyze buttons.

Adding Components—The first section houses buttons for adding new components to the spacecraft. DrawCraft comes equipped with the ability to build eleven predefined parts.

After clicking on a "New Part" button, the user is presented with a dialog box of options. The dialog box for creating each new part is nearly identical to the Bus Options Dialog Box in Figure 6. The dialog boxes differ only in the

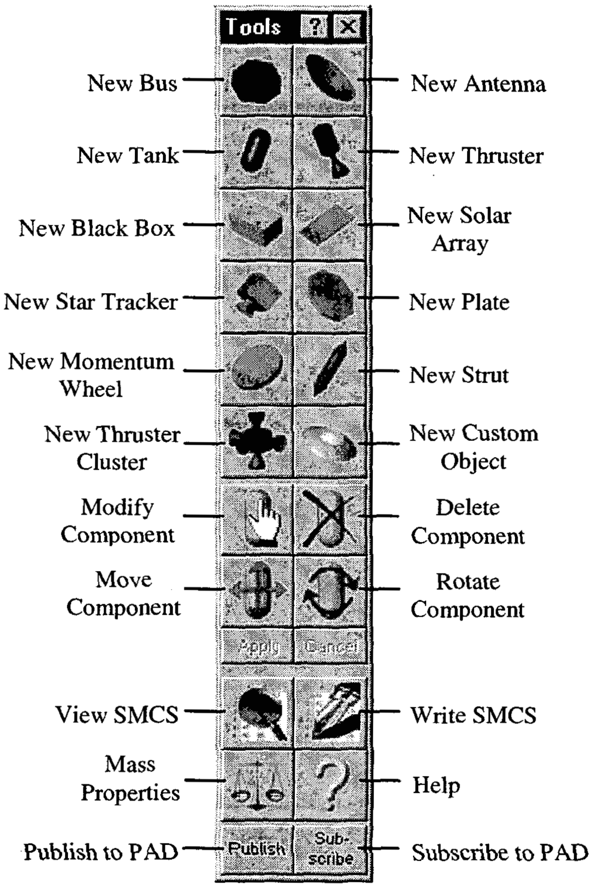

Figure 5: The Tools Palette

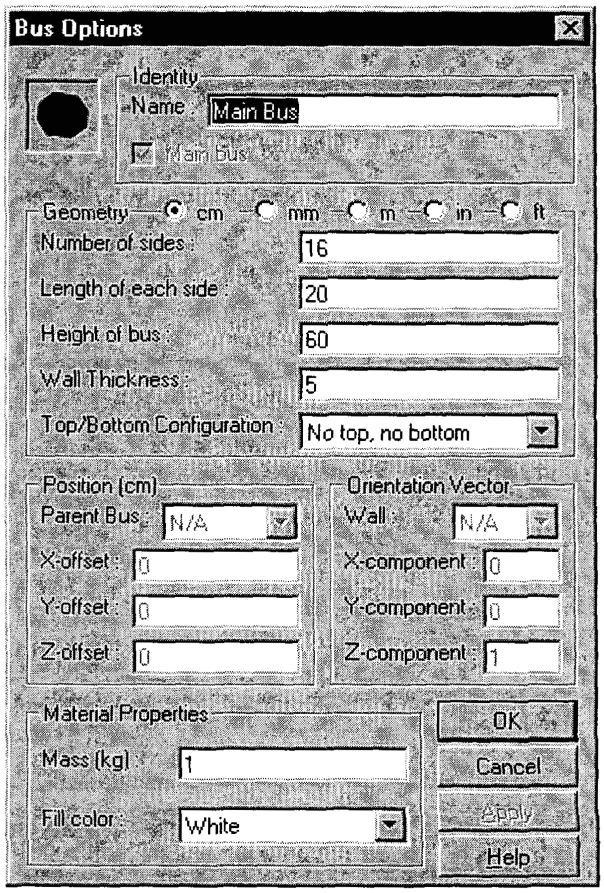

Figure 6: The Bus Options Dialog Box

geometric properties, which are unique to every part. In the dialog box, the user enters all of the data that are needed to fully define the part. Each attribute that one would find in a line of the SMCS can also be found on the corresponding options dialog box. After the user enters the identification, geometric properties, location, and material properties[3] of the new component, DrawCraft uses the SMCSChecker to verify that each entry is valid. If the dimensions are approved, DrawCraft then renders the new part, attaches it to the appropriate Bus, and adds the corresponding line to the SMCS.

The button immediately following the eleven predefined parts is for adding Custom Objects. When the Custom Object button is selected, the Custom Object Options Dialog Box appears, querying the user for the location of the SolidWorks Document to use. The user enters all of the identification, location, and material properties as with the other parts, but the geometric properties section allows the user to enter equations to specify certain dimensions within the Custom Object file. As with all parts, Custom Objects may only be attached to Buses.

Editing Components—The next section of the palette contains the buttons for editing components of the spacecraft. The first button is the Modify Component button. Clicking on this button brings up a list of the components of the spacecraft. After selecting the component that the user wants to modify, the same Options Dialog Box appears in which the user originally entered the part's attributes. Changes can be applied to every field except the Name and Parent Bus fields. Once changes to the dialog box are complete, DrawCraft changes the values in the SMCS and updates the parameters in SolidWorks. The adjacent button is for deleting components from the spacecraft, which consequently deletes them from both the SMCS and the SolidWorks model. Out of precaution, deleting a Bus is only allowed if it has no children.

Translation and Orientation vectors can also be changed manually by using the Move and Rotate buttons in the same section of the Tools Palette. Using this method, the user can freely move or rotate any component directly in SolidWorks, and DrawCraft will track the changes and update the DataMatrix in memory. When one moves or rotates a bus, all of the bus's children move with it such that their locations with respect to the Bus go unchanged. Thus, only the location information of the bus is altered unless the user specifically moves the parts individually.

For those who are more comfortable in the SolidWorks environment, the user can also edit dimensions and locations of parts within SolidWorks without using the DrawCraft Tools Palette. The next time DrawCraft is called upon to update the SMCS for saving or exporting, DrawCraft will update all of the parameters for each part from the SolidWorks model and save them to the DataMatrix.

Importing/Exporting Data—After the spacecraft is complete, the user has several options for what to do with the information. The SolidWorks rendering itself is independent of DrawCraft and can be saved and recalled on any computer on which SolidWorks is installed. DrawCraft can also export data to two different formats of information storage. The first two buttons of this section on the Tools Palette are "View SMCS" and "Write SMCS." Choosing either forces DrawCraft to recollect all of the geometry and location properties from the SolidWorks model of the spacecraft. "View SMCS" then simply displays the updated SMCS in a DrawCraft table while "Write SMCS" saves the SMCS as a tab-delimited text file for future use by DrawCraft or any third party software. The other method DrawCraft uses to export data is via the Products Attributes Database (PAD) at the Jet Propulsion Laboratory (JPL). The "Publish" button at the bottom of the Tools Palette forces DrawCraft to recollect all of the data in the SMCS from the SolidWorks model and then submits each entry of the SMCS to the PAD. The second button fills an SMCS with data from the PAD and renders the SolidWorks spacecraft accordingly. This method of importing and exporting data also eliminates the need for storing the SMCS on a disk or transferring it over the network.

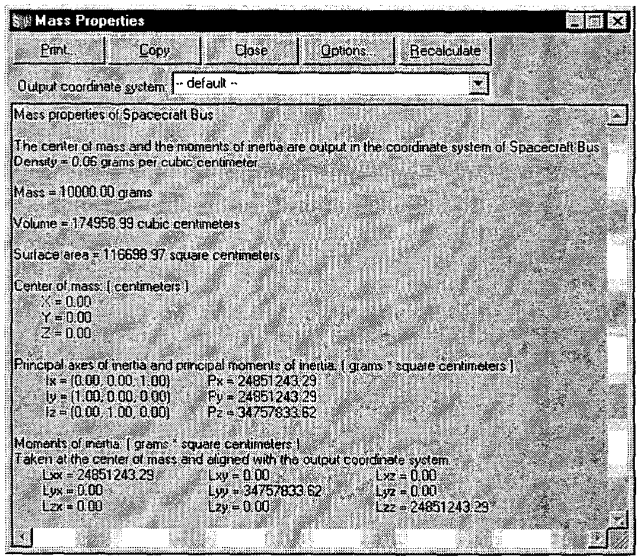

Figure 7: The Mass Properties feature of DrawCraft

Mass Properties—The last feature of DrawCraft is the Mass Properties function. This button actually calls a single command in SolidWorks, which calculates several physical properties of a part or assembly based on densities assigned by the user. When DrawCraft is building each component of the assembly it uses the mass stored in the DataMatrix and the volume given by SolidWorks to calculate the component's density and assigns the density to the SolidWorks model. When the user calls upon the Mass Properties function, SolidWorks uses several built-in

[3] See section 3 for a detailed description of each entry.

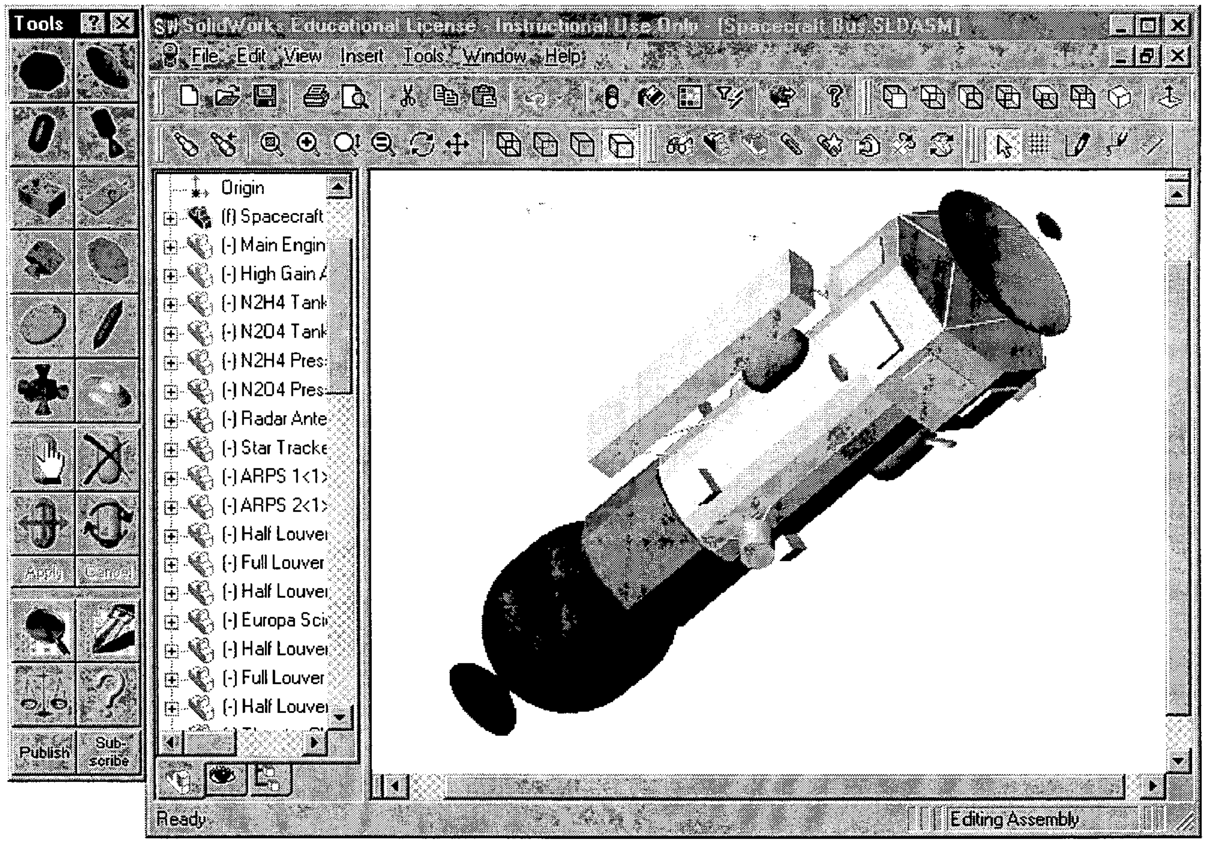

Figure 8: DrawCraft rendering of the Europa Orbitor

mathematical algorithms to calculate a number of useful properties of the spacecraft: total mass, volume, surface area, the center of mass, the principal axes of inertia, moments of inertia, and products of inertia. These properties are delivered to the user in the Mass Properties Window shown in Figure 7.

Online Help

DrawCraft also comes with complete online help. When the user pauses the cursor over one of the buttons on the Tools Palette, ToolTips automatically appear with a brief description of the button. If the user is still unsure as to the use of a button or what should go into a textbox, he or she can also click on the "What's This" question mark in the title bar and click on an item. This action loads DrawCraft's HTML Help file and opens the page that describes the button or textbox in question. The HTML Help file can also be opened by clicking on the Help Button at the bottom of the Tools Palette or any of the Options Dialog Boxes. Within the Help file are full descriptions of each Graphical User Interface and instructions on how to use the program. The Help also contains images of the DrawCraft's eleven predefined parts and drawings of what each geometric property represents for each part.

4. CODE IMPLEMENTATION

DrawCraft was written using Microsoft Visual Basic 6.0 with the SolidWorks Application Programming Interface (API). The best way to explain how DrawCraft was written is to describe the processes involved in loading a complete SMCS and rendering the corresponding spacecraft in SolidWorks. This basic functionality of SolidWorks can be organized into four small procedures and one large procedure.

Preparing the SMCS

After selecting the SMCS to load, DrawCraft reads each line of the tab-delimited text file into memory and converts it into a two-dimensional array of strings called the DataMatrix in which each line of the SMCS is a row. The next step is to check that each line of the DataMatrix describes a valid part and that the DataMatrix as a whole has no contradictions or violations. DrawCraft executes this step by running an SMCSChecker that goes through every entry of every line in the DataMatrix. When an error is found, DrawCraft tells the user which entry is invalid and what parameter that entry represents. DrawCraft then loads the SMCS in Notepad so that the user may correct the error and try again. While each line of the SMCS must represent a valid part, the lines of the SMCS need not be organized in the standard hierarchy of parts at this stage. This way, when the user changes the connectivity of components directly in the SMCS, he or she will not be forced to figure out how to rearrange the lines accordingly. After the DataMatrix passes the SMCSChecker, the DataMatrix undergoes complete reorganization. DrawCraft analyzes each line of the DataMatrix and places the lines in the order determined by the hierarchy of parts described in Section 3. The standard hierarchy of parts is crucial to the design process because it also tells DrawCraft in what order to build the components in. With the DataMatrix now ready to use, DrawCraft loads SolidWorks and initializes the global variables.

Building the Spacecraft

The longest procedure is the actual rendering of the spacecraft. The first step is to build the Main Bus, onto which most else will be attached. DrawCraft begins by opening a new assembly in SolidWorks. DrawCraft then opens the template for the Bus in SolidWorks and saves it under the name given for the Main Bus. The parameters of the bus are then altered to match those given in the SMCS and then the Main Bus is placed in the center of the Main Assembly. The Main Bus remains fixed here forever and cannot be moved or rotated.

The next step is to build and attach all of the parts that belong to the Main Bus. For a non-bus part, the template file is first opened and saved under a different name so as not to alter the template. The parameters of the new part are then replaced with those specified in the SMCS, and the part is saved again so that it may be added to the Main Assembly. By default, SolidWorks places all components in the center of the assembly to which they are added. DrawCraft in turn uses the translation and orientation vectors, which are given with respect to the wall, and runs a complex algorithm that converts the given data into coordinates for the component and its rotation matrix, with respect to the Main Assembly coordinate system. SolidWorks can use this information to place the component on the wall it belongs in the appropriate position. Although the part is not actually attached to the bus with a mate, the user can add mates once the design phase is finished. After all of the non-bus parts are built and added, DrawCraft proceeds to building the secondary buses.

DrawCraft calls upon the same procedure to build a secondary bus that it uses to build the Main Bus; thus, all buses are built in the exact same fashion. The procedure uses a recursive method, calling upon itself from within its own code, where each level of recursion ends when a bus is completed. The following steps repeat:

- Open new assembly
- Open bus template
- Make alterations
- Add bus to assembly
- Open part templates
- Make alterations
- Add parts to assembly
- Add children buses

Once there are no more children buses to add to a bus, DrawCraft adds the completed bus to its parent bus assembly and places it appropriately. After the last bus has been added to the Main Bus Assembly, DrawCraft resaves the assembly and shows the Tools Palette. The user now has full control of the program and may add or edit as he or she wishes.

Starting from Scratch

If the user chooses to start from scratch rather than load an SMCS or subscribe to the PAD, DrawCraft simply starts with a blank SMCS and forces the user to build a Main Bus. After the Main Bus is created, the spacecraft is officially complete and all other actions are executed via the Tools Palette. The rest of the spacecraft can be built by adding individual components.

Adding Components

If the user decides to add a new part, DrawCraft shows the Options Dialog Box for that part. After the user fills the Options Dialog Box with the attributes of the new part, DrawCraft uses the SMCSChecker to validate all of the entries; and if the dimensions are valid, DrawCraft adds the new line to the SMCS in memory. The procedures for actually building the spacecraft and adding it to the spacecraft are the same procedures that were used when building the spacecraft from the SMCS. DrawCraft opens the template for the part and alters the dimensions accordingly; the component is then added to its Parent Bus Assembly.

Exporting Data

When either "View SMCS," "Write SMCS," or "Publish to PAD" are selected, DrawCraft queries SolidWorks for each part's location and geometric properties. DrawCraft then converts the location information back into the coordinate systems of the buses' walls and compares the location and geometric properties received from SolidWorks with those stored in the DataMatrix. If the user has made any changes to the model with the SolidWorks interface, these changes are recognized and the DataMatrix is updated. DrawCraft then converts the DataMatrix into a tab-delimited text file and saves it to the user-inputted filename for the SMCS. Or in the case of publishing to the PAD, DrawCraft connects to the PAD and uploads each entry in the SMCS to its proper place in the database.

5. NETCRAFT

DrawCraft has many possibilities for expansion and development. The Laboratory for Spacecraft and Mission Design (LSMD) has developed a version of DrawCraft for use over the World Wide Web. NetCraft [5] allows visitors from the Web to enter an SMCS and produce a three-dimensional rendering of a spacecraft without requiring the user to have either DrawCraft or SolidWorks installed on his or her machine. NetCraft produces files in the Virtual Reality Modeling Language (VRML) because most users on the web are unable to view SolidWorks files.

NetCraft consists of four programs working together: DrawCraft, SolidWorks, NetCraft Server, and the NetCraft Applet. The first three programs mentioned are installed on the LSMD web-server, and the last is a Java applet within the LSMD website. When the user visits the NetCraft page[4], the NetCraft Applet loads and the user is given instructions. The user then types, pastes, or inserts a valid SMCS in tab-delimited form into the applet and clicks the "Render" button. The applet opens a TCP socket connection to the NetCraft Server, which is constantly running on the LSMD web-server. The NetCraft Server is a multi-threaded Java application that manages and coordinates the NetCraft system. One thread listens for new connections and places them in a queue for the second thread. The second thread takes a connection from the queue, saves the SMCS from the NetCraft Applet, and launches DrawCraft. DrawCraft reads the path of the SMCS file and the path of the output VRML file as command-line parameters. Once the rendering in SolidWorks is complete, the model is saved as a VRML file on the LSMD web server. The NetCraft Server informs the NetCraft Applet on the user's browser that the rendering is complete and the Applet redirects the browser to the VRML file. The NetCraft Server then closes the connection with the NetCraft Applet and takes the next connection from the queue. Using a VRML viewer, the user can then open the three-dimensional model of the spacecraft and manipulate it in space.

6. STATUS AND FUTURE PLANS

The need for an automated CAD modeler has long existed but has gone unrecognized by software developers until now. In response to this need, the Laboratory for Spacecraft and Mission Design (LSMD) has created DrawCraft, a powerful program for spacecraft modeling and visualization. DrawCraft is presently in its later stages of beta testing and is already being put to use at the LSMD. The leader of DrawCraft's original software development team is currently passing the software code on to two separate teams at the Jet Propulsion Laboratory and California Institute of Technology. These teams will take over the beta testing process, and once complete, both teams will collaborate, adding new features, more parts, and enhancing DrawCraft's functionality.

[4] www.lsmd.caltech.edu/designTools/design/netCraft/

7. REFERENCES

[1] MiniCAD: http://www.diehlgraphsoft.com/

[2] Joel Sercel, "The Product Attributes Database (PAD): First of a New Class of Productivity Tools for Product Development," *1997 IEEE Aerospace Applications Conference Proceedings.*

[3] SolidWorks: http://www.solidworks.com

[4] Microsoft Visual Basic: http://msdn.microsoft.com/vbasic/

[5] Craig Vieregg, "Ae125 TIDE Technical Report," *1999 Caltech TIDE*

8. ACKNOWLEDGEMENTS

The author of this paper would like to show his gratitude to the rest of the DrawCraft Development Team for their hard work and for making the programming go so smoothly. Geraud Krawezik is acknowledged for having designed the graphical user interfaces and dialog boxes, and even more so for actually making them work. Jackie Yeung is acknowledged for designing the online help and contributing the entire aesthetic portion of DrawCraft's interface. Dr. Joel Sercel is acknowledged for his vision of DrawCraft and his guidance throughout the design process. Dr. Sercel is also acknowledged for providing access to the laboratory in which DrawCraft's development took place. Michael Liu is acknowledged for his insight and advice on the code and also for teaching one of the team members how to program. Tammy Roust is acknowledged for her assistance with the Products Attributes Database at JPL. Scott Proper is acknowledged for his work on bug fixing and enhancing DrawCraft in its later stages of development. Dr. Felicia Dixon is acknowledged for her extensive assistance with writing, revising, and proofreading this article. The author would like to acknowledge Dr. Mathew Musick and Dr. Jonathon Daniels for their deep insight and support throughout the evolution of DrawCraft and this article.

9. BIOGRAPHY

Figure 9

Shahram "Bob" Ardalan, figure 9, is a second year undergraduate at the California Institute of Technology and is currently enrolled in both the Chemistry and Aeronautics Options. His primary work has been accomplished in the Laboratory for Spacecraft and Mission Design under Dr. Joel Sercel. At the LSMD, Shahram was a Lab Administrator and a programmer for the SolidWorks API. He coordinated the software development team for DrawCraft during the summer of 1999 and is currently training others to take over his work. Shahram plans to complete his Bachelors in Chemistry with a concentration in Physics and also hopes to pursue a doctorate in either Chemistry or Medicine. Shahram expects to begin work in the research group of Dr. Harry Gray and hopefully achieve a career in the Space Program.