Proceedings of the ASME 2010 International Mechanical Engineering Congress & Exposition
IMECE2010
November 12-18, 2010, Vancouver, British Columbia, Canada

IMECE2010-39213

TOWARD UNDERSTANDING COLLABORATIVE DESIGN CENTER TRADE STUDY SOFTWARE UPGRADE AND MIGRATION RISKS

Douglas L. Van Bossuyt
Graduate Research Assistant
Complex Engineered Systems Design Laboratory
School of Mechanical, Industrial,
and Manufacturing Engineering
Oregon State University
Corvallis, Oregon, 97331
Douglas.VanBossuyt@gmail.com

Irem Y. Tumer*
Associate Professor
Complex Engineered Systems Design Laboratory
School of Mechanical, Industrial,
and Manufacturing Engineering
Oregon State University
Corvallis, Oregon, 97331
Irem.Tumer@oregonstate.edu

ABSTRACT

Collaborative design centers often employ software tools to conduct trade studies. Commonly, this takes the form of a software program to aggregate and pass data between multiple computer workstations. This allows multiple people to concurrently create a conceptual design. Trade study software continues to evolve to meet the demands of modern collaborative design centers. However, the risks associated with moving from one trade study software tool to another are not well understood. Additionally, little is known about the software preferences of Collaborative Design Center (CDC) staff. This paper determines software preferences of two user groups consisting of graduate and undergraduate mechanical engineering students. This paper then explores the risks in deploying new trade study software in a collaborative design center. A method for estimating and mitigating risks with changing trade study software is presented. Recommendations for a smooth transition between software packages are given. The risk model developed in this paper offers a quick way of estimating and mitigating conversion risk for collaborative design centers.

1 INTRODUCTION

This paper explores the cost and risks associated with migrating from one trade study software program to another in a Collaborative Design Center (CDC) environment. Our hypothesis is that the risks associated with migrating between trade study software programs can be determined, modeled, and partially mitigated. This paper specifically evaluates transitioning from the ICEMaker [1] software package to ModelCenter [2] in a CDC environment.

This paper presents one possible way to define and calculate the risks involved in transitioning between trade study software packages. The opinions and work products of two populations of engineering student users – engineering graduate students and engineering undergraduate students – are analyzed to determine risks inherent in the transition and implementation processes. Software package preference is also determined from the opinions and work products of the user populations. The preferences of the user groups provide motivation for CDC practitioners and operators to transition away from ICEMaker to ModelCenter.

In the following, Section 2 presents a brief background on trade studies, CDCs, and trade study software. Section 3 describes the experimental setup and methodology to observe participants' reactions to, observations of, and work products from conducting trade studies using ICEMaker and ModelCenter. The models used in the trade studies were developed from Wertz and Larson's Space Mission Analysis and Design textbook [3]. The results of the study are presented and analyzed in Section 4, followed by a discussion of contributions and future work in Section 5.

*Address all correspondence to this author.

2 BACKGROUND

Several concepts, methodologies, and software packages are briefly reviewed in this section. These include a review of trade study methodology, CDCs, and the ICEMaker and ModelCenter software packages.

2.1 Trade Studies

Trade studies are used in the conceptual complex system design process to generate a set of designs that can be compared against one another. Trade studies can be performed manually by teams of people or automatically using software packages. While computer-generated trade studies can generate thousands of design points in a short time, human-generated trade study results are often more readily accepted because they are seen as having higher fidelity. This paper focuses on manual trade studies.

The goal of trade studies is to find a design with maximum utility for the system objectives and design constraints. To achieve this, trade-offs are made between system-level variables such as mass, power, cost, and other parameters [4]. At the start of a trade study, each subsystem is allocated a specific amount of the system-level variables. During the course of the trade study, multiple subsystems will be identified as lacking sufficient quantities of one or more system-level variables while having an overabundance of other variables. These variables are then traded between subsystems. Based upon the needs of the individual subsystems, the variables contain varying degrees of intrinsic value for the subsystem designers [5–7]. The conceptual designs that result can then be ranked according to appropriate selection rules [8–11].

2.2 Collaborative Design Centers

CDCs are an integral part of the conceptual design process at many companies and institutions. Trade studies are often conducted in CDCs as part of the early stages of a complex conceptual design. A great example of a successful CDC can be found at National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL)'s Project Design Center (PDC) and the associated design team, which is commonly referred to as Team-X [12].

The design team Team-X includes engineers and scientists from all major spacecraft mission subsystems. They are co-located in the PDC which is outfitted with the latest technology to aid in spacecraft mission development and concurrent design. The equipment, the CDC environment, and the personnel allow Team-X to complete spacecraft architecture, mission, and instrument design trade studies very rapidly [13]. Most studies conducted by Team-X are currently completed within 2 to 3 days, whereas, prior to Team-X, the process took 3 to 9 months [14]. In addition, the cost of concept-level spacecraft mission design has also been decreased by a factor of 5 compared to pre-Team-X design processes [14].

The success of Team-X led to the development of CDC environments using the methods of Team-X at other NASA centers. These include the Langley Research Center (LaRC) [15], two other groups at JPL known as Team-Z [16] and Team-I [17], NASA Goddard [18], and the Johnson Space Center [12]. Additionally, the European Space Agency (ESA) has replicated the methods used by Team-X [19]. Several academic institutions now house CDCs including Stanford, CalTech, MIT, the Technical University of Munich, Georgia Tech, and others [12, 20–22]. Several private companies have also adopted the Team-X approach to conducting trade studies, including Aerospace Corporation, Boeing, and the former TRW Corporation [12, 23].

2.3 Trade Study Software

Trade study software can be used to conduct automated or manual studies. Several different commercial and academic trade study software packages are available. Additionally, some CDCs have developed proprietary software. These include ICEMaker [1], ModelCenter [2], Advanced Trade Space Visualization (ATSV) [24, 25], and NeXSys [26], among others [27,28]. This research specifically explores ModelCenter and ICEMaker, thus warranting further background on the two software packages.

ICEMaker was originally developed by a team of graduate students at CalTech and was later commercialized. It was developed to allow CDC teams to conduct real-time trade studies. This is done using Microsoft Excel spreadsheets as the medium to transmit system-level variables between subsystems via a network server interface.

For close to a decade, ICEMaker has been the De facto standard software for CDCs. Team-X and several other aerospace organizations employ or employed ICEMaker as their primary trade study software package [1]. Recently however, ICEMaker was replaced at Team-X by NeXSys [26]. NeXSys maintains the same general Graphical User interface (GUI) and functionality as ICEMaker. The major upgrade of NeXSys over ICEMaker is a robust MySQL back-end where all system-level and pertinent subsystem variables are captured and stored. NeXSys, like ICEMaker, only interfaces with Microsoft Excel spreadsheets. In light of Team-X's switch and the discontinuation of development of ICEMaker, other CDCs can soon be expected to change the software packages they use as well.

To conduct more advanced analyses, several corporations such as Boeing, Raytheon, Ford, Samsung, BAE Systems, and others have introduced the use of ModelCenter [29] during automated trade studies, particularly for visualization. However, it is unclear in what capacity, if at all, ModelCenter is used in corporate CDC environments to conduct manual trade studies. The purpose of the experimental study conducted and described in this paper was to assess the cost and risks associated with Team-X migrating to ModelCenter as the software of choice in their model-based design sessions for manual trade studies. At present, Team-X does not use ModelCenter. However, some within JPL have expressed interest in its introduction [30], which was the impetus for this study.

Phoenix Integration's ModelCenter is a design space exploration, optimization, and trade study software package designed around the concepts of wrappers and plugins. Wrappers and plugins are used to interface design tools such as Microsoft Excel, EES, CATIA, MatLab, and any other software program that has an Application Programming Interface (API). The wrappers and plugins allow variables from models implemented in other software programs to be linked together so that variables from one model can interact with variables from another.

Many pre-packaged wrappers and plugins are available to link common software packages with ModelCenter. Additionally, "scriptWrappers" can be used to program an interface with external software packages. This is especially useful when communication with a custom-designed software package or database is desired.

A host of visualization techniques and trade study tools are included in ModelCenter. These tools allow a design team to quickly find weaknesses in a design, find optimal solution sets using multi-variable Trade Studies, and in general aid the design process [2]. Several of the advanced visualization techniques were adopted from ATSV including glyph plots and parallel axis plots. Visualization of Pareto frontiers and Pareto sets was also borrowed from ATSV.

This section reviewed several concepts, methodologies and software packages including trade study methodology, CDCs, and trade study software. Each topic is important to the remainder of this paper. These topics will be used in future sections to develop and conduct the experimental study.

3 METHODOLOGY

In order to understand and model the cost risks associated with migrating between the ICEMaker and ModelCenter software packages, an experiment was conducted. To conduct the experiment, several steps were completed including the development of simplified spacecraft models from Wertz and Larson [3], the implementation of the spacecraft models in ICEMaker and ModelCenter, and the creation of a simulated CDC environment.

Specifically, for the design trade studies, three groups of graduate and undergraduate engineering students were put through two simulated trade study sessions using ICEMaker and ModelCenter. Following each trade study session, work products were created, questionnaires were given, and a group discussion was held in order to identify and understand the risks associated with trade study software migration and methodology implementation and the software and methodology preferences of the study population. This section details the steps taken to conduct the trade study session experiments and collect necessary data.

3.1 Simplified Spacecraft Models

A simplified spacecraft model was developed from Wertz and Larson [3] using Microsoft Excel for typical satellite missions. Four representative subsystems were chosen to represent the spacecraft including Communication, Data Handling, Attitude Control, and Power. Each subsystem model was programmed to have two user inputs and three function or component-driven outputs. The inputs were specific to each subsystem. They consisted of either a drop-down menu where several component options could be chosen or an input box where bounded numeric values could be input to drive function-based models.

To replicate actual CDC trade studies, three outputs were chosen to represent spacecraft output data from all of the subsystems, namely, Subsystem Power Requirements, Subsystem Mass, and Subsystem Cost. All values including user-selectable inputs, internal variables, and outputs had all units intentionally removed. Additionally, all formulas and other numeric information was altered to only generally correspond to real-world spacecraft systems. This is exemplified by the subsystem cost parameter that generally ranged between a unitless figure of 1 and 30.

The Communication subsystem is a function-based model that accepts user input for the Antenna Size and Frequency Downlink variables. Antenna size can range from 1 to 4 and Frequency downlink can rage from 1 to 18, including decimal values. Both of the user input fields have corresponding instructions for the user to maintain input values between the allowable ranges. The Communication Subsystem Power Requirements, Mass, and Cost output variables were computed using the formulas shown in 1, 2, and 3, respectively.

$$\text{Power} = -\text{Antenna Size} + 0.6 \times \text{Frequency Downlink} + 3 \quad (1)$$

$$\text{Mass} = \text{Antenna Size} \times 2.5 + 2 \quad (2)$$

$$\text{Cost} = \text{Antenna Size} \times 0.75 + \text{Frequency Downlink} \times 0.1 \quad (3)$$

The Data Handling subsystem is a component-based model that contains two user inputs in the form of drop-down selection boxes. The first user input, System Complexity, has the options of "simple," "typical," and "complex." The other user input is Spacecraft Bus Configuration which allows the user to select either "one unit," "two unit," or "integrated" where one unit, two unit, and integrated refer to the spacecraft having one or two primary computing units and distributed subsystem computers, or an integrated unit that handles all command and data handling functionality. The resulting Data Handling subsystem outputs are shown in Table 1.

The Attitude Control subsystem is a component-based model that gives the user control over two inputs via drop-down

TABLE 1. Data Handling Subsystem Input and Output Variables

Input Vars		Output Vars		
System Complexity	Bus Config	Power	Mass	Cost
Simple	One Unit	7.5	4.8	0.9
Typical	One Unit	11.25	6.6	1.35
Complex	One Unit	15	12	1.8
Simple	Two Unit	11.25	3.6	1.575
Typical	Two Unit	16.875	4.95	2.3625
Complex	Two Unit	22.5	9	3.15
Simple	Integrated	6	2.8	1.35
Typical	Integrated	9	3.85	2.025
Complex	Integrated	12	7	2.7

TABLE 2. Attitude Control Subsystem Input and Output Variables

Input Vars		Output Vars		
Spin method	Pointing Method	Power	Mass	Cost
Gravity Gradient	Nadir Pointing	4.5	1.05	0.99
Gravity Gradient	Scanning	6	2.55	1.485
Gravity Gradient	Off-Nadir Pointing	3	1.05	1.188
Spin	Nadir Pointing	9	4.2	3.3
Spin	Scanning	12	10.2	4.95
Spin	Off-Nadir Pointing	6	4.2	3.96
3-Axis	Nadir Pointing	13.5	2.8	2.53
3-Axis	Scanning	18	6.8	3.795
3-Axis	Off-Nadir Pointing	9	2.8	3.036

selection boxes. The inputs are "Stability Method" and "Pointing Method." Table 2 displays the full range of user-selectable components and the corresponding output variable values.

The Power Subsystem is driven by a component-based model that has two inputs, namely, "Power Source" and "Energy Source," which are controllable via drop-down selection boxes. Table 3 presents the range of possible user-selectable input variable combinations and their corresponding output variables. Unlike the other three subsystems, the Power output variable for the Power subsystem indicates how much power is available to the entire spacecraft system from the power produced within the Power subsystem.

In addition to the four participant-controlled subsystems, a

TABLE 3. Power Subsystem Input and Output Variables

Input Vars		Output Vars		
Power Source	Pointing Method	Power	Mass	Cost
Photovoltaic	Primary Battery Only	41.25	3.8	1.9
Photovoltaic	Primary and Secondary	70.125	7.6	3.8
Static	Primary Battery Only	27.5	6.65	20
Static	Primary and Secondary	46.75	13.3	40
Dynamic	Primary Battery Only	82.5	13.3	1.4
Dynamic	Primary and Secondary	140.25	26.6	2.8

TABLE 4. Payload Subsystem Input and Output Variables

	Navigation	Astronomical
Power	50	20
Mass	2	2
Cost	6	6

Payload subsystem was also developed from Wertz and Larson [3]. It is used only to set the mission objectives and requirements. The two possible payloads consisted of a navigation and an astronomical package. Only one payload package was selectable at any given time. The Payload subsystem outputs power, mass, and cost variables. It also produces data on system constraints due to the payload. Table 4 presents the two payload choices and corresponding output data.

The simplified spacecraft models developed from Wertz and Larson [3] presented in this section were used to simulate the conceptual spacecraft design trade study process. Each subsystem takes user input either in the form of selection driving component-based models or numeric inputs driving function-based models. All unit information was intentionally expunged from the models. Constants used in the functional equations and output numbers from component models were altered to keep from closely resembling any real conceptual spacecraft designs. The subsystem models developed here are the basis of the experiments described below.

3.2 ICEMaker Implementation

The subsystem models were first implemented in ICEMaker Version 2.3.1 by copying the subsystem models built in stand-alone workbooks into the client workbooks created by the ICEMaker server. The pertinent output and input variables in each ICEMaker subsystem client spreadsheet were linked to the Output and Input sheets, respectively, of each ICEMaker subsystem workbook. As outlined in Section 2.3, data is sent and received by the ICEMaker subsystem client workbooks by clicking the Send and Request buttons.

3.3 ModelCenter Implementation

The same models were implemented in ModelCenter Version 9 next. Two options were considered to implement the subsystem models: 1) a central ModelCenter instance that would interact with individual subsystem models contained in Excel workbooks accessed via Analysis Server instances running on several computers, and, 2) individual instances of ModelCenter linked together using a series of custom scriptWrappers running on individual instances of Analysis Server, an analysis program connected to ModelCenter, designed to connect ModelCenter to a MySQL database. The second option was selected because it allowed for a more direct comparison of ModelCenter to ICEMaker, it allowed individual subsystem engineers to have access to the full suite of ModelCenter tools, and individual subsystem engineers were able to make decisions and submit design revisions as they desired rather than be forced to submit data at specific times by a regimented data collection process that a centralized instance of ModelCenter would require.

The custom scriptWrappers were written to send and receive data to a MySQL database. The database tracks each subsystem input and output variable, the time each subsystem sends data, and the order in which data is received and stored by the server. Within ModelCenter, the custom scriptWrapper provided the required input and output variables for each subsystem model.

The subsystem models were brought into the ModelCenter environment using the built-in Excel plug-in. The choice to implement the subsystem models in Excel over other more powerful and custom solutions was made in order to focus on the change from ICEMaker to ModelCenter. In this instance, continuity in interaction with the subsystem models helped to isolate the causes of satisfaction or dissatisfaction in conducting simulated trade studies to the trade study software being used.

3.4 CDC Room Setup

The Complex Engineered System Design (CESD) Lab at Oregon State University was used for the experiments, and was set up to replicate a typical CDC environment. Four user-controlled workstations were setup around the perimeter of the room. The center of the room was occupied by three researcher-controlled machines. One provided data visualization in the form of system-level and subsystem information in numeric and textual form on Display 1. The other researcher-controlled machines acted as the server and monitored the participant workstations. Figure 1 shows an overview of the room design.

During the ICEMaker and ModelCenter sessions, only Display 1 was used to project system-level and subsystem information in numeric and textual form. The other two displays were

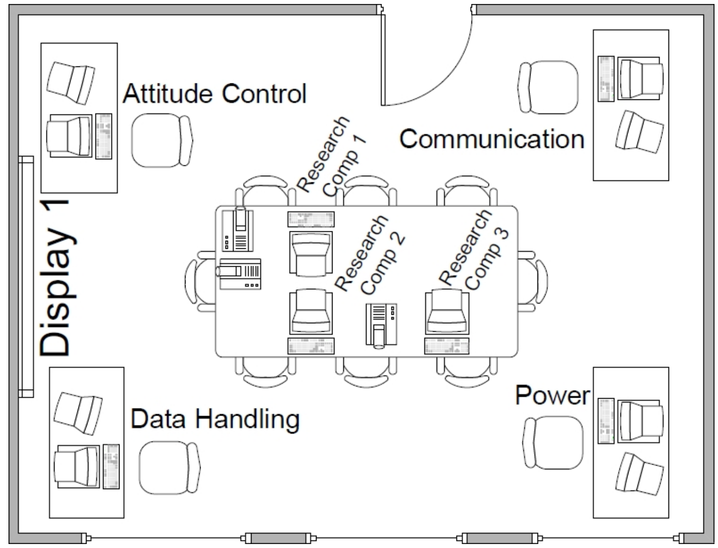

FIGURE 1. CDC Room Layout

powered off and unused (not shown in the Figure). Display 1 was controlled from Research Computer 1 and projected system-level and subsystem information.

Study participants sat at the four workstations around the perimeter of the room. The participants were free to get up and move about the room to interact with their colleagues running the other subsystem stations. The researchers generally stayed at the three researcher-controlled machines but would also go to the participants when they had questions. The conference table at the middle of the room was used to hold group discussions after each trade study session.

3.5 Study Population

Two distinct populations participated in the research study. One group of graduate mechanical engineering students, and two groups of undergraduate junior and senior level mechanical engineering students participated. Each group of participants was composed of four people.

The graduate mechanical engineering student group consisted of four people specializing in areas related to complex design, conceptual design, and collaborative design including trade studies. All have had experience with CDC environments, and are familiar with the general concepts of trade studies. All four participants have taken graduate level coursework at Oregon State University in state-of-the-art risk and model-based design methods. Two of the participants had previously interned with NASA, and hence also possessed general knowledge of conceptual satellite design. Several of the participants had been exposed to ModelCenter in the past but are not proficient in ModelCenter. None had seen ICEMaker before although several were aware of

TABLE 5. Mission Constraints

	Navigation	Astronomical
Energy Storage:	N/A	N/A
Power Source:	Photovoltaic	N/A
Spacecraft Bus:	N/A	N/A
Stability Method:	N/A	Gravity Gradient
Required Processing:	140	70
Maximum Mass:	45	35
Maximum Cost:	15	15

the program. This group of participants can be viewed as an expert user group. A analogous group of people in an established CDC would be people with one or two years of experience within the CDC.

Two groups of undergraduate mechanical engineering students participated in the experiment. Each group was a mix of junior-level and senior-level students who had satisfactorily completed junior level design courses that contained material on the mechanical design process, and had a collaborative design project. The undergraduate mechanical engineers did not have prior knowledge of trade studies, or the software used in the experiment. This group of participants can be considered a general user group. A similar group in a CDC might be engineers and scientists who are just being introduced to the CDC.

3.6 Mission Scenarios

Two mission scenarios were used for the two phases of the experiment including a navigation satellite, and an astronomical satellite. The missions were all earth-orbiting satellites that consisted of a series of design constraints and requirements. All constraints, requirements, and mission data were based upon information from Wertz and Larson [3] but were intentionally modified so that information used in this experiment did not closely resemble real-world satellite design information.

Both missions contained payload power, mass, and cost output variable data. Constraints placed upon subsystem design decisions were also provided. Table 4 details the payload requirements and Table 5 reviews the design constraints of both missions.

In addition to payload output variables and subsystem design constraints, both missions also demanded that cost and mass be minimized. At the same time, the missions required that a positive power balance be achieved. Additional general information about the function of a particular payload was provided to several groups who requested more details on the purpose of the missions and the scientific goals.

3.7 Questionnaires, Work Products, and Group Discussions

Four methods of data collection were used during the experiments. One method which was invisible to the participants was subsystem and system-level passively collected data from ICEMaker and ModelCenter. The other three methods required user input and interaction. Those methods requiring user input and interaction including work products, questionnaires, and group discussions which are detailed in this section.

At the end of both trade study sessions, the participants were asked to fill out a "System Design Report" document. The document asked the participants to write down all design decisions they made, the rationale behind those design decisions, and any comments that they had about the session. Participants were instructed to concentrate on their own individual subsystems but also record pertinent information on decisions and rationale of other subsystems with which they interacted.

Following the completion of the System Design Report, a questionnaire was administered to the participants. Table 8 in Appendix A details the questions used after the trade studies. All of the survey questions were designed to answer the questions posed in this research.

Group discussion followed completion of the System Design Report and the questionnaire in both trade study sessions. Two questions were repeated throughout the three sessions while other questions were tailored to each session. Table 9 in Appendix A lists the common and unique questions.

This section outlined the development of simplified spacecraft models from Wertz and Larson [3]. The implementation of these models into ICEMaker and ModelCenter was reviewed. The CESD Lab at OSU was configured to be a simulated CDC environment. Mission scenarios and user feedback instruments were created. And finally, a study population was recruited. The following section presents the result of the experimental study.

4 RESULTS

Analysis of the data collected during the experiment sessions shows that there are several potential risks in migrating from one trade study software package to another. These risks and a method to estimate them for CDC environments of varying size and trade studies of varying complexity are presented in the following section. The data indicates a clear participant preference for ModelCenter over ICEMaker to conduct people-in-the-loop trade studies in CDC environments. Details of these results and their implications are presented below.

4.1 Software Preferences

ModelCenter emerged as the nearly universally preferred software package to conduct trade studies. The study participants had distinct reasons for generally preferring ModelCenter over

ICEMaker. A few specific instances where ICEMaker would be the preferred software program were also identified. Their reasons for preferring one software over the other are detailed below.

The one graduate and two undergraduate student participant groups found ModelCenter to be a useful and preferred program for complex trade studies. However, the participants found that the simplistic models used in this experiment were equally usable with both ICEMaker and ModelCenter. Indeed, the graduate participants generally had a small preference for ICEMaker in the context of the models used in this experiment but were quick to point out that ModelCenter was the much more preferred option for all but the simplest of real-world studies.

The participants found that ModelCenter instilled a greater confidence in the results of the trade studies compared to ICEMaker. Results typical of all three participant groups are presented in Table 10 in Appendix B. Interestingly, the undergraduate student participants were more confident in the ModelCenter trade study results than the graduate students. All participants agreed that ModelCenter looks and feels more professional while ICEMaker appeared to them to be an outdated interface. They also found the optimization and Design of Experiments (DoE) tools, and the ability to interface with many different software programs to be of great benefit. The participants felt that presenting a result from ModelCenter would elicit a more favorable response from a boss or customer than presenting a result created using ICEMaker. One respondent wondered why ICEMaker would be used when the same functionality exists in Google Docs, an Internet-based document collaboration service [31].

No great difference in user interface or significant learning curve was identified in transitioning the participant groups from ICEMaker to ModelCenter. The participants found that the interface implemented in ModelCenter was easy to use and similar enough to ICEMaker to not cause any undue burden transferring from one software package to the other. In both instances, only one button needed to be clicked in order to send data to the server. Figure 2 shows the ModelCenter interface while Figure 3 shows the ICEMaker interface. Note that the Excel spreadsheet for ModelCenter is not shown and is identical to the spreadsheet shown in ICEMaker. Sending data to the server was done by pressing the “Send” button in ICEMaker while in ModelCenter this functionality was achieved by pressing the green play button. One participant indicated that it was very easy to learn how to use ModelCenter after having experience with ICEMaker. However, many participants found the titillating advanced features of ModelCenter to be too much of a temptation and spent some time investigating them. The time required to complete the ModelCenter session was approximately 15% or five minutes longer than the ICEMaker session. This can be attributed to the participants exploring the advanced functionality of ModelCenter not strictly required to complete the trade study.

All participants indicated that the brief 30 minute training sessions on ModelCenter and ICEMaker were not long enough to

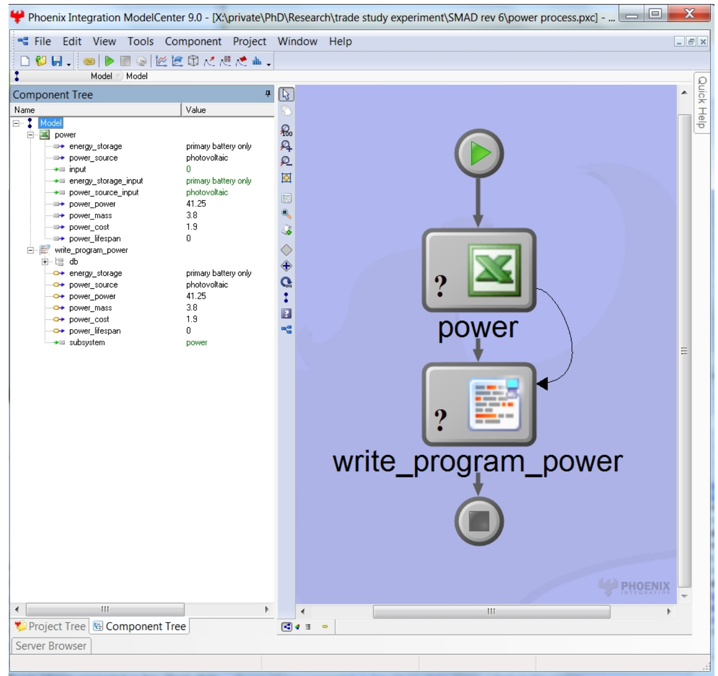

FIGURE 2. ModelCenter Interface

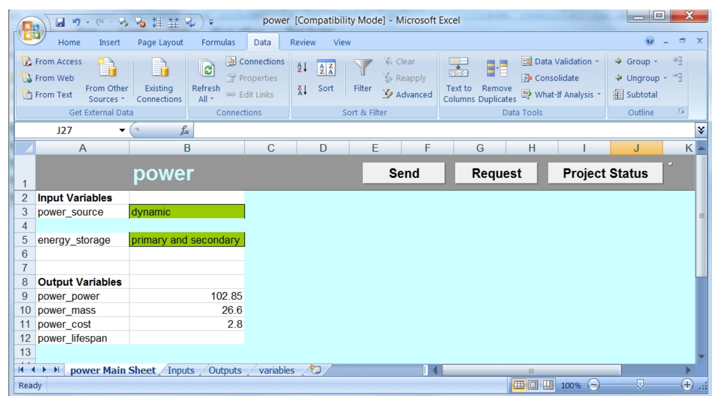

FIGURE 3. ICEMaker Interface

feel completely comfortable with the software. They found that having the wide array of enhanced functionality in ModelCenter available to use but without sufficient training to use it was a frustration. While the participants satisfactorily completed their work, they believed that the trade study sessions could have been completed much faster, more efficiently, and with less cognitive burden had they been more well-versed in ModelCenter or been told to not worry about the enhanced functionality. However, the participants noted that the minimal training they received coupled with minor support from the researchers allowed them to adequately complete the studies. The participants suggested that, rather than static training being presented via presentation slides, a hands-on demo would be much more effective.

4.2 Software Migration Risks

Several software migration risks were identified during the course of preparing for and conducting the experiments. Firstly, training resurfaced several times as lacking. Second, one of the stations running ModelCenter experienced several minor glitches. FInally, the time required to migrate the models from ICEMaker to ModelCenter was much greater than initially expected.

During the experiment sessions, participants were given very brief introductions to both ModelCenter and ICEMaker. While a brief introduction sufficed for ICEMaker, participants found that they wanted to know more about ModelCenter. Participants could easily use the basic functionality of ModelCenter to complete the trade studies but consistently became curious about the advanced features. They also had a strong interest in understanding how the back-end of ModelCenter works compared to an almost non-existent desire to understand how ICEMaker works. This resulted in the more curious participants attempting to tinker with the more advanced functions of ModelCenter and in one case caused their instance of ModelCenter to crash. The problem was recovered from quickly.

The researchers developed the subsystem models in Excel intentionally considering an implementation in ICEMaker and ignoring an implementation in ModelCenter. After the models had been implemented in ICEMaker, the researchers then implemented them in ModelCenter. What the research team had assumed would only take a few hours to complete turned into a 60 hour project to develop and debug scriptWrappers that would allow multiple ModelCenter instances to interact with a MySQL database. Once the scriptWrappers were developed, bringing the subsystem models into ModelCenter and testing the setup took only a few hours.

In summary, data analysis points toward several potential risks that CDC operators must be aware of when migrating between trade study software packages. In spite of the risks that CDCs face in changing trade study software packages, participant feedback clearly indicated that ModelCenter is preferred over ICEMaker. In order to satisfy the software preferences of the users while not causing undue hardship on CDCs, characterization and mitigation of the risks involved with migrating software packages must be completed.

5 DISCUSSION

In this section a risk model is developed to aid CDC operators in determining if transitioning from ICEMaker or similar trade study software packages to ModelCenter or similar programs is a worthwhile investment of time and resources. Risk in this case is defined as the time required to make the transition between software packages. Recommendations are presented to decrease the amount of time and thus the amount of risk to a CDC transitioning to a new trade study software package.

5.1 Modeling the Risks Associated with Migrating Between Software Packages

The results of the study show that the participants preferred ModelCenter over ICEMaker in real-world CDC environments. The primary source of risks associated with migrating between software packages is the time required to migrate from one software package to another. This is due to the tight timelines and budgets under which many CDCs operate. This section develops a model of the risks associated with migrating a CDC from ICEMaker to ModelCenter.

All steps of the migration process require a time commitment by part or all of the CDC staff. The Information Technology (IT) department will require time to learn ModelCenter and MySQL. Following that, the IT department will need time to develop the scriptWrappers required to connect ModelCenter to a database back end, and to implement the subsystem models in ModelCenter. Time should then be devoted to debug and test the software as well. The end users must spend some time being trained on the software. Finally, a learning curve and period of adjustment to the software will manifest as extra time required to complete trade studies. Equation 4 lays out the calculation.

$$T_S = T_{IT} + T_D + T_{SU} + T_{DT} + T_{TU} + LC \tag{4}$$

where T_S is defined as the time to switch from ICEMaker to ModelCenter, T_{IT} is the time for IT Personnel Training, T_D is the scriptWrapper Development Time, T_{SU} is the Setup Time, T_{DT} is the Debug and Test Time, T_{TU} is the Time to Train Users, and LC is the learning curve.

Training time for both the IT staff and end users can be estimated from the training courses offered by Phoenix Integration [32]. A basic course in using ModelCenter requires two days of time. Advanced training runs one to two days depending upon the skill level of the participants. Training in the design exploration and optimization tools included in ModelCenter requires one day. Based upon this information and the experiences of the researchers in preparing for the experiment, the researchers believe that the IT staff will benefit from taking the basic course, advanced course, and the design exploration and optimization course. A reasonable estimate of the time required to complete these courses is five business days.

Based upon the experiences of the researchers in developing the scriptWrappers, at least two weeks of time should be budgeted to develop the scriptWrappers. This will provide enough time to develop a scriptWapper that robustly and securely interacts with a MySQL database. If a complex permission-based interface is desired within the ModelCenter environment to restrict database access, additional time will be required.

The time required to integrate models into ModelCenter is determined by how many variables are in each model and how many models have been successfully integrated by the IT staff. Equation 5 shows a means to calculate a reasonable estimate for time in hours to complete the integration of trade study models into ModelCenter. The time to complete model integration will vary as a factor of IT experience. Models that do not have pre-built wrappers or plug-ins will require additional setup time.

$$T_{SU} = \sum_{n=1}^{N_M} \left(N_V \times \sum_{m=1}^{N_M} (m \times LC) \right), \quad (5)$$

$$LC = \begin{cases} .3\,hours & \text{When IT staff has worked with less than three models in ModelCenter} \\ .1\,hour & \text{When IT staff has worked with three or more models in ModelCenter} \end{cases}$$

where T_{SU} is setup time, N_M is the Number of Models, N_V is the Number of Model Variables, LC is the learning curve, and where each model has five or more variables.

The time required to debugging and testing models implemented into ModelCenter can be expected to vary as a function of the experience level of the IT staff assigned to carry out the task, the number of model variables, and the number of models to be debugged and tested. The researchers experienced both highly competent IT professionals and junior personnel. Equation 6 shows a means to estimate the time required to debug and test model implementation.

$$T_{DT} = \frac{N_V}{3} \times \sum_{n=1}^{N_M} n \times \text{IT Staff}, \quad (6)$$

$$\text{IT Staff} = \begin{cases} 0.5 & \text{Inexperienced IT Staff} \\ 0.3 & \text{Experienced IT Staff} \\ 0.1 & \text{Expert IT Staff} \end{cases}$$

where T_{DT} is the time to debug and test, N_M is the Number of Models, N_V is the Number of Model Variables.

The time necessary to train users will vary depending upon the desired level of user interaction with ModelCenter. Based upon the results of the study, a user can be trained in under 30 minutes to use ModelCenter in the same way as ICEMaker is used. However, many users will quickly want to learn more about ModelCenter. Additionally, the more useful portions of ModelCenter require further training. Table 6 lists reasonable amounts of time required to train a user for various levels of ModelCenter proficiency.

TABLE 6. ModelCenter Training for Various Levels of Proficiency

Proficiency Level	Hours
ICEMaker Equivalent	0.5
Basic Optimization and Design Exploration	4
General ModelCenter Training	16
Advanced ModelCenter Training	16
Advanced Optimization and Design Exploration	16

Determining the learning curve of a user group in a CDC environment is beyond the ability of this research to predict with any level of accuracy. Based upon the experiments, the researchers believe several trade study sessions will need to be completed before a CDC is as efficient at completing trade studies with ModelCenter as they were with ICEMaker. All three participant groups took approximately 15% longer to conduct a trade study using ModelCenter for the first time as compared to conducting a trade study using ICEMaker for the first time. Because long-term testing was beyond the abilities of this research, this paper does not provide any guidance on the long-term learning curve of trade study software.

The risk model developed above applies well to the experiment detailed in this paper. It took approximately one week to conduct IT training for the one IT staff member who worked on the experiment. Setup time ran two person-weeks to develop a complete scriptWrapper package. Model integration time, as calculated by Equation 5, should have been 50.6 hours. In reality, this took about 47 hours. Debug and test time calculated from Equation 6 took 37.5 hours while in reality it took 40 hours. User training took approximately 30 minutes to conduct in a group setting. As stated previously, a learning curve is not estimable at this time. However, the ModelCenter trade study that was conducted took 5 minutes longer than the ICEMaker trade study. Summing these numbers provides a total theoretical time to switch from ICEMaker to ModelCenter of approximately 205 hours while in reality it took approximately 210 hours. The researchers originally estimated only 40 hours was required to switch the experiment CDC from ICEMaker to ModelCenter. Had this risk model been available prior to the start of the experiment, the researchers could have more accurately budgeted for the amount of time required to complete the implementation. The risk model developed in this section will help future CDC operators from making the same mistake.

TABLE 7. Time to Switch from ICEMaker to ModelCenter for the Experiment CDC

Time (hr)	T_{IT}	T_D	T_{SU}	T_{DT}	T_{TU}	LC	T_S
Theoretical	40	80	50.6	37.5	0.5	0.08	210
Actual	40	80	47	40	0.5	0.08	210

5.2 Methods for Mitigating Software Migration Risks

Several potential strategies have been identified to mitigate some of the risks involved with transitioning to a new piece of trade study software. The strategies presented here will not be applicable in all cases. A degree of engineering and management judgment must be applied when selecting and implementing these strategies.

In order to satisfy the curiosity of some users while not overwhelming other users, multiple levels of ModelCenter training can be offered. ModelCenter can be used only as a simple interface to link a subsystem model with other models. Less than half an hour of training is required to teach a user how to properly operate ModelCenter in this fashion. A complex and deep understanding of all that ModelCenter has to offer requires at least a week of training plus additional time to practice and become comfortable with ModelCenter.

In a CDC where a MySQL or other database back end is already used to link subsystem models together, a phased or targeted roll-out of ModelCenter is possible. Subsystems staffed by people who wish to switch to ModelCenter or subsystems identified as having a high level of benefit from adopting ModelCenter can be transitioned over first while other subsystems can be transitioned at a later date or indefinitely use old but still functional software.

In order to cut out some of the development and testing time required for scriptWrapper development, outside consultants can be hired. This will free up valuable IT time for other uses. However, consultants should not be used to replace acquiring internal expert knowledge of ModelCenter.

In this section, a risk model was developed that allows CDCs operators to understand the risks that a CDC faces in migrating from one trade study software package to another. This risk model characterizes the risk to CDCs based on the amount of time that will be required to complete the implementation and transition process. Several methods for reducing the risk associated with trade study software migration were presented also presented. Equipped with this information, CDC operators can decide whether it is an appropriate time to transition to ModelCenter.

6 CONCLUSION

This paper presents an experimental study to determine the risks associated with migrating a CDC conducting trade studies from one trade study software package to another. This paper specifically evaluates the transition between the ICEMaker and ModelCenter software packages in CDC environments. Methods are developed in this paper to analyze the risks associated with trade study software package transition. Recommendations are presented to reduce the identified risks.

Through several data collection methods, it was determined that the participants in the research study preferred ModelCenter over ICEMaker to conduct trade studies for a variety of reasons. Those reasons included greater confidence in the results, and belief that the results from ModelCenter would be more readily accepted by superiors and customers as compared to results from ICEMaker. Participants also found that the transition from ICEMaker to ModelCenter was not particularly difficult although most participants wished for extended training in ModelCenter in order to use the software to its full potential.

The benefits of transitioning from ICEMaker to ModelCenter outweigh the risks in some CDC environments. This paper presents methods of examining the risks associated with this transition. With strong evidence of support for transitioning to ModelCenter, it is up to CDC operators to judge if the time is right for their operations to make the switch. The risk model and risk reductions methods developed and presented in this paper are good starting point for CDC professionals to make the decision to adopt ModelCenter and the risk trading methodology.

ACKNOWLEDGMENT

This research was carried out in part for JPL, Caltech, under contract with NASA. Thanks goes to Scott Ragon, Taurik Elgabrowny, and others at Phoenix Integration for providing software donations and technical support, and to Kevin Parkin, who was one of the original authors of ICEMaker, for providing software donations. Special thanks goes to Leanne Lai at the Oregon State University College of Engineering Computing Support group for her efforts. The authors wish to specifically acknowledge the efforts of Josh Wilcox who was instrumental in preparing and conducting the experiments. Thanks also goes to Sarah Oman and Michael Koch for their assistance in conducting the experiment. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators.

REFERENCES

[1] Parkin, K. L., Sercel, J. C., Liu, M. J., and Thunnissen, D. P., 2003. "Icemaker: An excel-based environment for collaborative design". In Aerospace Conference.

[2] Phoenix Integration Inc., 2008. PHX Model Center, Nov. `http://www.phoenix-int.com/software/phx_modelcenter.php`.

[3] Wertz, J. R., and Larson, W. J., 1999. *Space Mission Analysis and Design*. Springer.

[4] Papalambros, P. Y., and Wilde, D. J., 2000. *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press.

[5] NASA, 1995. *NASA Systems Engineering Handbook*. NASA.

[6] FAA, 2006. *National Airspace System System Engineering Manual*, 3.1 ed. Federal Aviation Administration ATO Operations Planning.

[7] Ross, A. M., Hastings, D. E., Warmkessel, J. M., and Diller, N. P., 2004. "Multi-attribute tradespace exploration as front end for effective space system design". *Journal of Spacecraft and Rockets,* ***41***(1), pp. 20–29.

[8] Ravi, B., and Srinivasan, M. N., 1990. "Decision criteria for computer-aided parting surface design". *Journal of Computer Aided Design,* ***22***.

[9] Russell, J. S., and Skibniewski, M. J., 1988. "Decision criteria in contractor prequalification". *Journal of Management in Engineering,* ***4***(2), pp. 148–164.

[10] Bacon, C. J., 1992. "The use of decision criteria in selecting information systems/technology investments". *MIS Quarterly,* ***16***(3), pp. 335–353.

[11] Ji, H., Yang, M. C., and Honda, T., 2007. "A probabalistic approach for extracting design preferences from design team discussion". In Proceedings of the ASME Design Engineering Technical Conferences; Computers and Information in Engineering Conference, IDETC/CIE.

[12] Shishko, R., 2000. "The proliferation of pdc-type environments in industry and universities". In Proceedings of the 2nd European Systems Engineering Conference, EuSEC.

[13] Deutsch, M.-J., and Nichols, J. S., 2000. "Advanced approach to concept and design studies for space missions". *Astrophysics and Space Science,* ***273***, pp. 201–206.

[14] Oberto, R. E., Nilsen, E., Cohen, R., Wheeler, R., DeFlorio, P., and Borden, C., 2005. "The NASA exploration design team: Blueprint for a new design paradigm". In Proceedings of the 2005 Aerospace Conference, no. 8957662 in IEEE Conferences, IEEE, pp. 4398–4405.

[15] Gough, K. M., Allen, B. D., and Amundsen, R. M., 2005. "Collaborative mission design at NASA Langley Research Center". In Space Systems Engineering Conference.

[16] Edelson, R. E., 2000. "Team Z, a rapid reaction approach to mission operations system design and costing". In The Proceedings of SpaceOps, AIAA.

[17] Oxnevad, K. I., 1998. "A concurrent design approach for designing space telescopes and instruments". In Space Telescopes and Instruments.

[18] Karpati, G., Martin., J., and Steiner, M., 2003. "The integrated mission design center (IMDC) at NASA Goddard Space Flight Center". In Aerospace Conference.

[19] Bandecchi, M., Melton, B., and Ongaro, F., 1999. "Concurrent engineering applied to space mission assessment and design". *ESA Bulletin,* ***99***.

[20] Osburg, J., and Mavris, D., 2005. "A collaborative design environment to support multidisciplinary conceptual systems design". *SAE Transactions,* ***114***, pp. 1508–1516.

[21] Finkel, S., Wilke, M., Metzger, H., and Wahnfried, M., 2002. "Design centers - transfering experience from astronautics to aeronautics". In Proceedings of the 12th Annual Symposium of the International Council on Systems Engineering, INCOSE.

[22] Masher, T. J., and Kwong, J., 2004. "The space systems analysis laboratory: Utah State Universitys new concurrent engineering facility". In Aerospace Conference.

[23] Aguilar, J. A., Dawdy, A. B., and Law, G. W., 1998. "The aerospace corporations concept design center". In Proceedings of the 8th Annual International Symposium of the International Council on Systems Engineering.

[24] Stump, G. M., Yukish, M., Simpson, T. W., and O'Hara, J. J., 2004. "Trade space exploration of satellite datasets using a design by shopping paradigm". In Aerospace Conference.

[25] Simpson, T. W., Carlsen, D. E., Congdon, C. D., Stump, G., and Yukish, M. A., 2008. "Trade space exploration of a wing design problem using visual steering and multidimensional data visualization". In 4th AIAA Multidisciplinary Design Optimization Specialist Conference.

[26] Warfield, K., and Hihn, J., 2009. "Spreadsheets in Team X: Preserving order in an inherently chaotic environment". In Proceedings of the 42nd Hawaii International Conference on System Sciences.

[27] Volk, S. K., Wheeler, R., Wilkinson, B., Jones, M., and Birgel, S., 2000. "Concurrent real time engineering via the fredrik work environment: Helping engineers produce their products by structuring their access to relevant information". In Proceedings of the International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE.

[28] Meshkat, L., Weiss, K. A., Luna, M., and Leveson, N., 2006. "Supporting concurrent engineering in JPLs advanced project design team using a systems engineering development environment". In Proceedings of Virtual Concept.

[29] Phoenix Integration, 2010. Phoenix integration customer list. `http://www.phoenix-int.com/customers/customer_list.php`.

[30] Morse, E., Leavens, T., Cohanim, B., Harmon, C., Mahr, E., and Lewis, B., 2006. "Next-generation concurrent engineering: Developing models to complement point designs". In Aerospace Conference.

[31] Google, 2010. Google docs. http://docs.google.com/.
[32] Phoenix Integration, 2010. Software training. http://www.phoenix-int.com/software/software_training.php.

Nomenclature

ATSV Advanced Trade Space Visualization
CATIA Computer Aided Tridimensional Interactive Application
CDC Collaborative Design Center
EES Engineering Equation Solver
ESA European Space Agency
JPL Jet Propulsion Laboratory
LaRC Langley Research Center
NASA National Aeronautics and Space Administration
PDC Project Design Center
GUI Graphical User interface
API Application Programming Interface
DoE Design of Experiments
IT Information Technology

APPENDIX A: QUESTIONNAIRE QUESTIONS AND GROUP DISCUSSION PROMPTS

TABLE 8. Trade Study Session Questions

Session 1: ICEMaker						
Describe any difficulties you encountered with understanding and using the ICEMaker software:						
Session 2: ModelCenter						
Describe any difficulties you encountered with understanding and using the ModelCenter software:						
Indicate your preference of software platform for conducting trade studies:						
Prefer ICEMaker						Prefer ModelCenter
High	Medium	Low	No Preference	Low	Medium	High
Comments:						
How did you find the transition from ICEMaker to ModelCenter?						
Easy	1	2	3	4	5	Hard
Indicate which software produced results in which you feel more confident:						
Confident in ModelCenter results						Confident in ICEMaker results
Very Confident		Somewhat Confident	Neutral	Somewhat Confident		Very Confident
Comments:						
Is there anything that should have been done differently when transitioning from ICEMaker to ModelCenter?						

TABLE 9. Group Discussion Questions

Common Questions:	
Question 1:	Were any of the subsystem models hard to understand and use? Were any particularly easy?
Question 2:	Did you prefer component-based or function-based subsystem models?
ICEMaker Session Questions:	
Question 1:	Did you encounter any difficulties in using ICEMaker?
Question 2:	Was there anything that could have been done differently to help you more quickly and efficiently complete the trade study?
ModelCenter Session Questions:	
Question 1:	Did you encounter any difficulties in using ModelCenter?
Question 2:	Were there any difficulties in making the transition between ICEMaker and ModelCenter?
Question 3:	Which software tool do you think produced a better result? Which would you be more comfortable to show your boss?
Question 4:	Was there anything that could have been done differently to help you more quickly and efficiently transition from using ICEMaker to ModelCenter?

APPENDIX B: TRADE STUDY USER INPUTS AND RESULTS

TABLE 10. Trade Study User Inputs and Results

ICEMaker User Inputs and Results						
	Power	Communication	Data Handling	Attitude	Payload	
	Power Source	Antenna Size	System Complexity	Stability Method	Mission	
	Dynamic Energy Storage Primary and Secondary	1 Frequency Downlink 4	Typical Spacecraft Bus 1 Unit	Gravity Gradient Pointing Method Nadir Pointing	Astronomical	
						TOTAL
Power	102.85	4.4	11.25	4.5	60	22.7
Mass	26.6	4.5	6.6	1.05	2	40.75
Cost	2.8	1.15	1.35	0.99	6	12.29
ModelCenter User Inputs and Results						
	Power	Communication	Data Handling	Attitude	Payload	
	Power Source	Antenna Size	System Complexity	Stability Method	Mission	
	Primary and Secondary Energy Storage Photovoltaic	2 Frequency Downlink 18	Typical Spacecraft Bus 2 Unit	Nadir Pointing Pointing Method Gravity Gradient		
						TOTAL
Power	70.13	11.8	16.88	4.5	35	1.95
Mass	7.6	7	4.95	1.05	2	22.6
Cost	3.8	3.3	2.36	0.99	6	16.45