

**A STUDY OF DATABASE MIGRATION:
UNDERSTANDING THE USER EXPERIENCE**

A Thesis Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Scott Eugene Shumate
December 2019

Copyright © 2019 by Scott E. Shumate
All rights reserved.

ACKNOWLEDGEMENTS

I would like to thank my family and friends for their boundless support and understanding as I completed this thesis. Special thanks to my mentor, Gina Garber for her support and encouragement. I am also indebted to my committee members, Dr. Diane Kelly and Dr. Awa Zhu, for serving on my thesis committee and providing constructive comments and criticisms on my manuscript. To my adviser, Peiling Wang, for her guidance throughout my Master's program and thesis research.

I would also like to thank Christina Chester-Fangman and Katie Pendleton for their assistance in editing and preparing this thesis.

Last but not least, I wish to thank the participants for sharing their insights and thoughts into the database migration experiences.

ABSTRACT

This thesis reports on a participatory field study of a database migration project. The purpose of the study is to understand the user experiences in the database migration. Both the field observations during the process and the post-migration user interviews provided valuable data to gain insight into what challenges the users faced and how they adapted to the new database. The findings suggest that the users were generally positive about the new database, although they were still not entirely confident about the system's outputs or certain functions. They developed strategies to ensure data accuracy: keeping paper archives to check outputs and postponing implementation of certain modules. Some of the challenges were due to the missing vital steps in data migration, such as data migration without performing data cleaning first. The year-long learning and interaction with the database have not resulted in a coherent mental model of the database; the contributing factors include 1) the complexity of the back-end database schema, 2) the inclusive all-in-one front-end interface design, and 3) the disadvantage of the generic trainer-based training sessions held before the data migration. From this study, we provide five recommendations for a robust data migration process: 1) data cleaning before migration, 2) incorporating elements to help build adequate mental models, 3) training on the migrated real database (never on a demo database), 4) customizability to hide unused tabs and functions, 5) creating task-based user guides.

TABLE OF CONTENTS

Chapter One. Introduction and Research Questions	1
Chapter Two. Literature.....	4
Database Migration.....	4
The Process	4
Impact on Users During and After Migration.....	8
User Experiences in System Migration.....	9
User Tasks.....	9
Mental Models	9
How People Adapt to New Technologies	11
Chapter Three. Methodology	14
Organization and Participants	15
Timeline	16
Data Collection Instrument.....	19
Interviews.....	21
Field Observation.....	22
Coding Schemas and Data Analysis	26
Chapter Four. Findings	31
Interview Findings	31
Use of Database	31
Favorable Experiences	35
User Challenges	40

Mental Models	45
User Adaptation Strategies	47
Field Observation Findings.....	49
Database Back-end (Data Schema).....	49
Database Front-end (Interface)	51
Migration Process and User Training	55
Chapter Five. Discussion	61
What are the users' experiences in a database migration?.....	61
How do the users adapt to the new database?.....	69
What are the users' mental models of the new database?.....	71
How do users' mental models impact the use of the new database?	72
Limitations and Contributions	74
Chapter Six. Conclusions and Recommendations	76
List of References	79
Appendix.....	84
Appendix A – Interview Guide.....	85
Vita.....	88

LIST OF TABLES

Table 3.1. Question mapping between research questions and interview questions	20
--	----

LIST OF FIGURES

Figure 3.1 Example of Field Note Database Display	25
Figure 4.1 User Interface: Communications Center	33
Figure 4.2 User Interface: Reports Management Center	34
Figure 4.3 User Interface: Event Management and Attendance Reporting	36
Figure 4.4 User Interface: Example Member Record	37
Figure 4.5 Example of Save Interface.....	43
Figure 4.6 User Interface: Advanced Feature for SQL Query Builder	52
Figure 4.7 Query Builder	56
Figure 4.8 User Guide/ “Cheat Sheet” Created by Database Volunteers	59
Figure 4.9 User Guide/“Cheat Sheets” Created by Database Volunteers.....	60
Figure 4.10 Knowledge Base / Help Center	70

CHAPTER ONE.

INTRODUCTION AND RESEARCH QUESTIONS

The digital revolution is the legacy of the last seventy years. Information technologies grew in capabilities, shrank in size, and became integrated into the daily lives of more and more people every day. Organizations the world over have left behind their paper records and filing cabinets, and now make use of sophisticated information systems for both data storage and business operations. These systems are utilized by the organization to make use of member data, events, programs, donations, reporting data, and other information essential to the organization's operations. Because of this, employees interact with these systems frequently. Each system is built on a different framework with different functionality, and inevitably these organizations are required to migrate from an old system to a new one. There are many reasons an organization might choose to undergo these migrations. Over time, the needs of an organization shift as it grows and expands, which can result in an existing system no longer meeting those needs, and no change to the existing system can accommodate them. Additionally, the financial situation at the organization may change, whether it be from a reduction in available funding, a lull in business, or simply a change in the value proposition where new technologies can offer more and better services for a similar or reduced price. Lastly, simple obsolescence may mean that the old system is out of date or no longer capable of interacting with more modern equivalents required for other business operations. Whatever the reason may be, the result is a migration process that impacts both the organization and individuals who work directly with the system.

So what are the users' experiences as a product of this migration? Users must learn and adapt to the new information system while still completing their job tasks. Due to the complex nature of these systems, users will need an understanding of the system that allows them to solve the problems they might encounter in the course of their work. Users combine training and practical experience, often unknowingly, to develop their understanding into a mental model that helps them to navigate the underlying data and gives them expectations of the software. As the users interact with the system, they develop expectations and learn how it reacts to certain inputs and how to complete their tasks within the system. Training, documentation, and experience all contribute to the mental models of the users, and over time, these models become more robust, leading to an understanding that is essential to successfully navigating the system, especially when faced with problems beyond simple completion of repetitive tasks.

This study observes the impact of such a migration process on the users' experiences at a non-profit organization undergoing a database migration from one database and service provider to another. This is a participatory study where the researchers worked with the organization assisting with navigating the migration process and learning the new system. The researchers recorded their observations and documented meetings, training, and support sessions with the users before, during, and after the five-month migration. The migration itself consisted of back-end data migration, user training, and post-migration support.

Nearly one year after the migration was deemed complete by the service provider, the researchers conducted interviews with the users to further understand their experiences by looking at their perception of the system, the migration process, and the ongoing benefits and challenges observed following the migration. Using the researchers' notes to corroborate and support these interviews, this thesis addresses the following research questions:

1. What are the users' experiences in a database migration?
 - What tasks do users utilize the database for?
 - What aspects of the migration do they view favorably?
 - What challenges do they face?
 - How do users learn during migration?
2. How do the users adapt to the new database?
3. What are the users' mental models of the new database?
4. How do users' mental models impact the use of the new database?

CHAPTER TWO. LITERATURE

This chapter reviews two lines of research: the system migration process and user experience with a focus on mental models that the users construct through interactions. During and after the migration, the users and the system impact each other, and understanding the migration process as well as the experience of the users during the migration is essential to understanding both.

Database Migration

The Process

Database migration is a process of moving an existing database application to another database management system (DBMS) or service provider. During this process, the existing data are exported from the current DBMS and imported to the new DBMS. Although the same data are likely available in the migrated database, interactions with the database are likely to change, especially in cases when migrations occur from very old systems, those built on different technologies, or when the new database is not constructed specifically for the intended users. These systems exist across businesses and organizations of all different sizes and compositions, from Fortune 500 companies to academic libraries. While different in function to the end-user, these systems share many of the same underlying technologies. Additionally, much of the actions taken to prepare for the migration from a management standpoint are relevant, regardless of the environment in which the migration is taking place.

The migration process involves some basic tasks as defined in Bilal (2014): extract the data, install the test site, load the data, test the data, and identify errors or problems in data mapping, retest the data after the problems are fixed; import the data to the new system upon implementation of new system. Further, Bilal highly recommends data cleanup before migrating to the new system. Data cleanup involves: removing and weeding data that no longer in use, handling missing or lost data, correcting erroneous data, and purging patron records if they are no longer affiliated with the organization. Singh (2013) encourages similar steps when migrating a library system to an open-source alternative, providing recommendations for planning, testing, data preparation, customization, installation, maintenance, and training. Evidence of the successful employment of these strategies can be found throughout the literature, and while the organizations differ, the principles behind the migration are shared between them. Much as information systems are integral to businesses and organizations of different size and scope, so too are they present in a wide variety of industries and types of organizations. These example studies can serve as further guidance to the migration process, and the lessons learned can improve the quality of the migrations going forward.

Libraries, especially academic libraries, have extensive experience of information system migrations because managing, storing, and eventually migrating data is central to the core services provided by libraries and information professionals. Multiple case studies have been conducted on migrating integrated library systems. For example, Nfila, Dintwe,

and Rao (2005); Clarke and Morris (2009); Julich, Hirst, and Thompson (2003); Hallmark and Garcia (1992); and others, as evidenced by Balas (2011).

Thomer, Cheng, Schneider, Twidale, & Ludäscher's (2017) look into natural history museum databases provides another look at migration in a different environment. They provide insight regarding migrating systems, both between two systems and upgrading from a legacy system to a newer version. While this study focuses on migration from one system to another, many cases may exist where the migration is from one version of a system to a newer version, and many of these same steps may take place in both types of migrations.

Gholami, Daneshgar, Beydoun, and Rabhi (2017) studied migrations to cloud platforms in businesses. Non-profits differ from businesses in many ways, but the information systems upon which they are founded are not that dissimilar, and in many cases, they share the same providers. Migration to cloud platforms is a cutting-edge technique, but it is important to analyze going forward into an increasingly cloud-based future.

Land (2013) also provides a set of guidelines for users planning a content migration, and notably includes both an information architect (IA) and a content specialist (CS) as key members of the migration team. These key members of the migration team exist to oversee the handling and migration of data and assure its integrity and usability on the other side of the migration. Prior to the migration, Land employs a "discovery phase" where data is analyzed and a strategy for migration is put in place. Many of the steps recommended

by Bilal and Singh fit well inside this discovery phase, such as planning and data cleanup. With the IA and CS as members of the migration team, informed choices can be made about how to proceed with the migration while ensuring the integrity of the data. The information architect is able to take on the responsibility of looking at how the data was and will be structured, and how best to make the old database interact with the new one effectively. This relieves some of the technical burden from other users, and also frees up time and effort for the CS to properly inform the process with regard to the organization's data specifically. In a small organization, the IA may be hard to identify. A member of the information technology department might be an obvious choice, as technical proficiency and familiarity with both the data and underlying systems will be integral to the IA's responsibilities. However, it is entirely likely that a small organization might not have a full-time information technology staff member, not to mention department. Fortunately, identifying a CS in such a small organization is likely easier, as a smaller staff means that employees are likely more immersed in the content and can better serve as the specialist required to fill this role.

Bisbal, Lawless, Wu, and Grimson (1999) write about dealing with legacy information systems. They view all options surrounding these mission-critical information systems from a complete redesign to migration. As part of the discussion of the option, they deal with some strategies and issues surrounding a legacy system migration. How a migration will be performed differs from one database to the next. The database, hardware, and underlying technologies all affect the migration process. Some "migrations" bypass

the actual data migration entirely, opting instead for a system called a “gateway.” Gateways are systems that stand between legacy systems and modern systems to translate data between the two, thereby avoiding the migration process by leaving the legacy system intact but making its data available to the newer system. Wu, Lawless, Bisbal, Richardson, Grimson, Wade, and O'Sullivan (1997) propose a strategy known as the “Butterfly Methodology” for handling migrations of legacy systems. The Butterfly Methodology provides an alternate to gateways, which allows for a gateway-less solution to the system migration problem. This methodology provides a straightforward plan for taking the data from an old system, a metaphor for the larva, and providing a process whereby the migration can take place resulting in the creation of a new system, the butterfly.

Not all migrations are successful, and the cause is not always grand over-arching problems. Estelle (2017) describes a case of software migration and how a relatively minor problem resulted in a completely failed migration. There are similarities between the organization in Estelle’s study and the organization in this study, in that they have complaints and issues regarding their migration. Understanding failure is key to preventing it, and Estelle provides a window into one such failed migration.

Impact on Users During and After Migration

While these case studies provide valuable insights, lessons, and best practices for database designers and system administrators, few studies reveal users’ perspectives—their experiences and challenges in the migration process, especially users in small non-profit organizations that do not have strong technical support. This study aims to fill that gap.

User Experiences in System Migration

User Tasks

The database within an organization supports organizational operations and maintains data across functions. These functions supporting user tasks are delivered through a user interface. Lauesen and colleagues compared two classic approaches to interface design, the data-oriented approach, and the task-based approach. The former was easier for the database designers but lacked efficient task support of users. The latter requires the designer to start with user tasks and take steps to complete these tasks so that the system can support user tasks well. Although the task-oriented design makes the system easier to use, the users never get to know the entire database. They proposed a new approach to link user tasks, data models, and interface design: the Tasks and Support method. (Lauesen, 2003; Lauesen & Harning, 2001).

Mental Models

Databases support tasks that may have been previously managed by papers and files, or another system with similar functions. A certain level of understanding of the objects in the physical world or a database is needed to interact with these objects. This understanding is conceptualized as mental models.

Mental Models were first defined as people's mental representations of domain knowledge that provides the basis for people to make inferences about the domain (Gentner & Stevens, 1983; Johnson-Laird, 1983; Roger, 1992). It is possible that people develop coherent but naïve theories of physical phenomena; mental models that represent only

fragmented knowledge, which by nature is a set of loosely connected ideas (Roger, 1992). The concept of mental models has been discussed at two levels in psychology: the knowledge representation level and the reasoning level. At the knowledge representation level, a mental model “mirrors” the perceived structure of the external system being modeled (Johnson-Laird, 1983), so that meaning can be derived from actual states of affairs as they exist in the world. At the inference and reasoning level, because mental models are simplified and incomplete versions of the reality, running mental models is less cognitively demanding. As a result, people can make inferences about the world efficiently.

Many researchers have tried to classify mental models. Rasmussen’s (1986) five types of mental models include physical form, physical function, functional structure, abstract function, and functional meaning or purpose. Based on Rasmussen’s work, Rouse and Morris (1986) proposed that mental models could represent a system purpose (why a system exists), function (how a system operates), state (what a system is doing), or form (what a system looks like). Another approach to mental models divides them into structural models, also called a surrogate model, and functional models, also called a task/action model (diSessa, 1986; Young, 1981). One of the three models by Carroll and Olson (1987) is the metaphor model, which is a direct comparison between the system and some other systems already known to the user. This is similar to Young’s strong analogy or surrogate models, by which the users are trying to understand the system at the assimilatory level. The more advanced level at which the user understands the system is considered the accommodatory level (Young, 1981). Mental models impact learning and interaction with

a new database. Users develop mental models through training and experience, and previous experiences can influence the models they develop about new systems. Understanding the users' mental models and how they are interacting with the software because of this model can be significant in understanding how to solve problems better and improve the users' experiences. The more complete a user's mental model, the easier for the user to adapt to a new system in performing work tasks.

How People Adapt to New Technologies

DeSanctis and Poole (1994) developed Adaptive Structuration Theory (AST), an idea built on capturing the mutual relationship between advanced information technologies and their users. Building on sociotechnical systems theory, structural symbolic interaction theory, and other research into Group Decision Support Systems (GDSS), AST acknowledges the deep relationship between information technologies and their users, especially in a professional capacity. AST looks at both sides of the equation to find solutions that are optimal for both technology and user. Further, DeSanctis and Poole posit that advanced information technologies inherently have a "spirit," which roughly defines the general intent of the system, and that spirit must be communicated to the user. The method and effectiveness of how this communication occurs can affect the usability of the system by limiting options with restrictive structures, or expanding options with a more informal spirit, but at the risk of increasing confusion. The spirit of the system can also be employed faithfully or unfaithfully, and the system may be used for an intent other than its spirit, and this can be seen in decisions by both management and users. DeSanctis and

Poole note that one of the attitudes a group can display toward a technology is, “the extent to which groups are confident and relaxed in their use of the technology” (p.130). The spirit of the system and how the organization goes with or against this spirit can inform how best to solve problems and make management decisions regarding efficiently employing the system for organizational tasks.

Beaudry and Pinsonneault (2005) define the Coping Model of User Adaptation. This model looks at how users handle changed in information technology in their professional life. Beaudry and Pinsonneault adapt the psychological Coping Theory to these human-computer interactions. Notably, they observe how users interact with a system before ever seeing it in their appraisals, looking at how they believe a particular change in the technology will affect them personally and professionally. Following the appraisals, users take actions to adapt to the change, known as their coping efforts. These efforts can be emotional adaptations, centered around changing the user’s perception and reducing stress, or problem-focused adaptations, which directly address issues with the system. These adaptations can be through changes to the system itself, changes to the user’s understanding through training or further experience, or adjusting work responsibilities.

Beaudry and Pinsonneault (2005) identify four principal adaptation theories: benefits maximizing, benefits satisficing, disturbance handling, and self-preservation. Benefits maximizing is making the most of a change seen as an opportunity, as users seek to maximize the benefit for themselves and their regular duties from a function. While this can be challenging for those who have trouble seeing the opportunity in the change, it can

be greatly beneficial to the organization, as the individual becomes user, expert, and advocate for the new technology. Benefits satisficing tends to arise from situations where users feel they have less control and simply seek to get satisfactory events from the change without seeking maximum benefit. Disturbance handling comes from a perception of the change as a threat that the user feels can be handled. In many ways, this is a “damage control” mode where users shift into the mindset of trying to handle problems associated with the transition head-on. Depending on the success of the problem-solving efforts and the end-result impact on the users’ job responsibilities, this strategy can result in the users feeling more confident and in control of the new system, or at least restoring some emotional stability as they believe they can survive and perform their duties after the transition. Lastly, self-preservation stems from a perception of the change as a threat over which they have minimal control. In this case, the user will take steps to assure the preservation of their emotional stability by reducing negative consequences however possible. This often results in an improvement for the user, but has little impact on the organization or the work completed by the individual. These attitudes and response mechanisms play a significant role in how users involve themselves in a transition. Understanding the user’s appraisals of the change in software allows for employment of these adaptations that assist both user and organization in successfully navigating the migration.

CHAPTER THREE. METHODOLOGY

This study is a participatory field study where the researcher worked as one of the volunteers with an organization to help users in the database migration process while observing the process of the migration. After the completion of the database migration, the users were invited to take part in an interview to better understand their experiences and the mental model they have developed learning and using the system. Interviews are an important research design to collect data from users with the advantage of the interaction between the interviewer and interviewee.

Field observation following the established methods of workplace observation (Clancey, 2006) and participatory research (Wildemuth, 2017) collected data systematically during the migration process when the researcher was learning the new system with the users and helped them in troubleshooting problems. The structured field notes focus on what problems users encounter, how the users attempted to solve the problem, and the nature of the problems. These field notes serve two purposes: to design handouts for user tasks and to corroborate with user interviews. When the volunteer work started as an outreach service to the non-profit organization, we kept the structured field notes for documenting the database migration. After the research project was approved, these notes became valuable data.

Organization and Participants

The study was conducted in a small, non-profit organization in the Eastern United States. The organization provides support and services to members of the community. The organization was faced with a price increase to its existing database that also lacked many desirable functions. Upon examining alternatives, they determined that a newer, more modern product could better meet their organizational needs while saving them money compared to the existing solution. The organization began working with the service provider on the migration process in late 2017. Upon the suggestion of a board member, the organization reached out to the University of Tennessee, Knoxville for database expert assistance. Three volunteers began working with the organization in December of 2017. The service provider began data migration in January of 2018 and completed the process in April of 2018. User training sessions provided by the service provider were given from January to March of 2018. During this period, the volunteers attended most of the user training sessions and helped the users solve problems they identified in database outputs. The volunteers also communicated with the service provider using database terminology. Field notes were kept as documentation of the database and data migration.

Recognizing the values of the experiences from this database migration project, the director of the organization suggested that what was learned in this real-world project would also help other organizations in their database migrations. Therefore, the researchers applied for approval of the Institutional Review Board at the University of Tennessee to

conduct this field study. In the research proposal, interviews with migrated database users would focus on collecting users' experiences in their own words.

The organization has five full-time employees and numerous volunteers that contribute their time to the organization. The organization provides free classes, workshops, seminars, and charity events. The database keeps records of members, donors, events, and participation. Of the organization's five employees, four of them use the database. All four users were interviewed for this study. The level of technical expertise varies among these users. Three of them had experience with the organization's old database, while the fourth, the newest employee, was tasked with learning the new database. The volunteers did not use the database at the time of the interviews; thus, none of the volunteers participated in this study. A full timeline of the migration is as follows:

Timeline

December 2017

- 5th – Email contacts began regarding database help.
- 12th – Researchers visited the organization for preliminary discussion regarding the database migration project, field note 20171212-1.
- 18th – Further emails regarding the assistance of the researchers.

January 2018

- 8th – Last day to enter data to the old database.
- 14th – Final export of old database to be given to service provider for migration.
- 16th – First of the user training sessions that would continue until March

- 21st – Preliminary data import into the new database completed and access given to the organization for review.
- 30th – A total of 10 training sessions covering different modules and functions completed, field note 20180130-1.

February 2018

- 20th – Meeting between researchers, director of organization, and staff member, field note 20180220-1.
- 28th – Meeting between researchers, organization, and CEO of service provider, field note 20180228-1.

March 2018

- 7th – Training provided by service provider.

April 2018

- 1st – End of unlimited support from service provider.
- 11th – Meeting between researchers and organization, field note 20180411-1.

May 2018

- 14th – Meeting between researchers and organization regarding regular mailings, field note 20180514-1.
- 16th – Meeting between researchers and organization regarding regular mailings, field note 20180516-1.
- 22nd – Meeting between researchers and organization, field note 20180522-1.
- 29th – Launch of service provider's knowledge base/help center.

- 30th – Meeting between researchers and organization regarding data migration issues, field note 20180530-1.

June 2018

- 3rd – Meeting between researchers and organization, field note 20180603-1.
- 26th – Meeting between researchers and organization, field note 20180626-1.

August 2018

- 2nd – Meeting between researchers and organization, field note 20180802-1.
- 8th – Meeting between researchers and organization, field note 20180808-1.

September 2018

- 5th – Institutional Review Board approval.
- 6th – Meeting between researchers and organization, field note 20180906-1.

October 2018

- 3rd – Meeting between researchers and member of organization's staff for training, field note 20181003-1.
- 9th – Meeting between researchers and member of organization's staff for training, field note 20181009-1.
- 18th – Meeting between researchers and member of organization's staff for training, field note 20181009-1.

November 2018

- 7th – First Service-provider Monthly Support call on the topic of Email Communications.

December 2018

- 5th – Service-provider Monthly Support call on the topic of Member Data and Process Management.

February 2019

- 28th – First interview.

March 2019

- 6th – Second and third interviews.
- 12th – Fourth interview.

Data Collection Instrument

An interview guide was developed based on the four research questions. The interview starts with an open-ended question to engage the conversation: “When you think of [name of the database], what comes to your mind?”

The interview guide includes a total of 18 questions (Appendix A). As Table 3.1 shows, the interview questions were designed to solicit aspects of user experiences, adapting to the migrated database, and user’s mental models. For example, question 3 asks, “What are the important parts of [name of the database] that are relevant to your job tasks?” This question solicits information on the user’s tasks and the system’s features that support the completion of the tasks.

Because users’ mental models of the database represent how users understand the system and how they perceive its functions, mental models impact their interaction with and feelings about the system. Seven questions throughout the interview elicit users’ understanding of what the database contains and how it works.

Understanding not only the nature of the incidents identified by the users, but also the way in which the users perceived these incidents was a central focus of the interview. Several questions were designed to simultaneously ask the user to identify incidents, whether they viewed them positively or negatively, and why. Question 8 elicits negative incidents and question 10 positive incidents:

8. Could you tell me a specific instance when you encountered difficulties working with [name of the database]? I am interested in what you were trying to accomplish, what problems occurred, and how the problems were solved.

10. Could you tell me a pleasant instance you have experienced using [name of the database]?

Question 17, not mapped in Table 3.1, was only relevant to the decision-maker of the migration: “Do you have any advice to other directors of non-profit organizations in need of database migration?”

Table 3.1. Question mapping between research questions and interview questions

Research Question	Facet	Interview Question
Users’ experiences		2, 3, 8, 9, 10, 11, 12, 13, 18
	Usage	2
	Tasks and Features	3, 8, 9, 10, 11
	Favorable experiences	10, 11, 18
	Challenges	8, 9, 12, 13, 18
	Learning	12, 13
Adapting to the migrated database		8, 9, 10, 11
Users’ mental models		1, 4, 5, 6, 7, 14, 15
	Data storage	4, 6, 7
	Data organization	5

Interviews

The researcher conducted interviews between February 28th and March 12th, 2019. Both the participant and researcher were in their respective offices, but due to distance limitations, interviews were conducted online using the Zoom web conferencing software. Participants received a consent form and were asked verbally if they granted permission to record the interview as audio-video data of the research. These recordings were stored on the university's secure server only accessible by the researchers.

The researcher relied on the interview guide (Appendix A) to structure the interview, but due to the difficulty in expressing elements of the interview's desired outcomes, such as mental models, the researcher often had to provide clarifying statements that helped to guide the participant to an answer without influencing the content of their response. For example, when participants were asked how they believed the data was stored, three of the four participants answered that they did not understand the question or did not know. Therefore, the researcher asked the participant to consider a higher-level view of how the data might be organized. This was successful in eliciting a response that gave insight into the user's mental model of the database.

“When you're looking at the data and deciding what you need to extract, what is the whiteboard you're looking at deciding how to extract that information?”

The interviews ranged from thirty-six minutes to one hour and seventeen minutes. In total, participants' responses consisted of 21,472 words. Question 17 was asked only of the administrative member of the organization.

Field Observation

Field notes were created for each incident related to the system, including every interaction with members of the organization. These notes began as documentation of the issues encountered and solutions created for each issue addressed by the researchers. Over time, these notes evolved into an element of the research that was able to supplement the data received from the interviews with the users. Each note was assigned a unique identifier, given an incident name, a full description of the incident(s) involved or discussed, and any thoughts or strategies from the researcher. When possible, communication, such as emails or other correspondence, was attached to the field note. Each incident was assigned a tag from the tag schema to isolate the incident's source(s).

Each note was created following an interaction with a member of the organization or observation of an incident by the researcher within the system itself. Because notes were organized by incident, a single meeting or interaction could result in multiple notes being created as several incidents might be discussed. In addition to documenting what incidents were discussed in the meetings, the notes also allowed for the researchers' insights to be documented for help during the problem-solving process. Notes were updated as issues were often ongoing and required more than a single meeting or interaction to resolve. Incidents generally required time and experimentation to reach a solution. Interactions with members of the organization, the service provider, or between the researchers were stored as attachments to the field note or added to the description of the incident. Presented here is a condensed example of a field note from the project:

UID: 20180220-1

Date: 2018-02-20

Issue: Duplicate Records in Mailing Lists

Description: As the organization prepared to send out regular mailings, they noticed duplicate records that did not exist in previous mailings generated by the old system. User has been unable to confirm the existence of duplicate records for individuals who are on the list multiple times. A ticket has been submitted with the service provider but has been unable to provide resolution in a timely manner.

Attempted to generate a list using the mailing that in some way did not include these duplicates. Attempts to use the de-dupe tool were unsuccessful, as the duplicates found by hand are not detected by it. Generated a list using the Reports area and Excel that contains all of these duplicates so that they can be fixed manually, but not likely to be done by the time of mailing.

Working with [staff member], decided to generate a list of members that have been added and opted in to the mailing from the new system and combine it in Excel with the known-good list from last mailing.

Researcher Notes: De-duping tool is not registering the observed duplicates as dupes and therefore cannot auto-merge them. Manual inspection shows inconsistent results as to whether there are actually two records in the system for duplicates observed in the mailing list. Inspection of SQL query generated by the system report is not revealing as it is overly complex.

Areas of Concern: Database error, data integrity, support

Attached: 20180220-Email.pdf, 20180220-Recording.mp4, Old Mailing List (Excel), New Mailing List (Excel)

Field notes were consolidated into a central database for long-term storage and ease of use (Figure 3.1). MySQL was used for the database, and a simple PHP program was used to interact with and display the data through a web browser. The database allowed for incidents to be related to each other for research and historical purposes, as solutions may be shared across multiple incidents. This system allowed for quick access to view and change information related to an incident, as well as searching for related incidents found in other field notes. Many tasks share common database tools, such as the reporting area, so that solutions might be found in other field notes. Attached files were stored on the local disk, but can be opened directly from the browser for quick access when reviewing or troubleshooting. Despite the web-based nature of the program, it was only run locally due to security and privacy concerns regarding the data.

Creating field notes throughout the researchers' participation allowed for additional insight when combined with the interviews. Because the researchers were familiar with the issues discussed in the interviews, coding data was made easier and more accurate. The researcher was also able to separate similar problems with different causes, such as duplicate records because of data migration issues, and duplicate records that were just the same record being displayed twice due to multiple addresses attached to it. Because the users lacked the familiarity and expertise with databases, they would have grouped these issues together, when, in fact, they were two problems presenting in a similar fashion.

←

→

↺

🏠

http://localhost/display.php

Thesis DB

UID:

20120220-1

Date:

2012-02-20

Incident:

Duplicate Records in Mailing Lists

Attachment:

[Email-20180220-1.pdf](#)

Full description:

As the organization prepared to send out regular mailings, they noticed duplicate records that did not exist in previous mailings generated by the old system. User has been unable to confirm the existence of duplicate records for individuals who are on the list multiple times. Support has been contacted but has been unable to provide resolution in a timely manner.

Researcher Notes:

De-duping tool is not registering the observed duplicates as dupes and therefore cannot auto-merge them. Manual inspection shows inconsistent results as to whether there are actually two records in the system for duplicates observed in the mailing list. Inspection of SQL query generated by the system report is not revealing as it is overly complex.

Tags:

1.0 Back-end Incidents

1.1 Database Error

1.2 Data Integrity

1.2.1 Legacy Data Issue

1.2.2 Migration Issue

1.2.3 Data entry/process

2.0 Front-end Incidents

2.1 User Interface

2.1.1 Affordances

2.1.2 Labeling/Identification

2.2 User Experience

3.0 Service Provider Related Incidents

3.1 System Functionality

3.2 Support

3.3 Training and Documentation

Edit

Add New...

Figure 3.1 Example of Field Note Database Display

Coding Schemas and Data Analysis

A coding schema was developed for interview data analysis. The interview coding schema was developed using a top-down approach based on the research questions that was then supplemented after the interviews based on the participants' responses. Based on the literature and previous experiences, a basic set of codes were generated prior to beginning the coding process. While coding, the researchers added new tags to the schema to accommodate any responses that did not fit within an existing tag. Presented here is the final schema that is a result of this iterative process:

1. Users' experiences

- | | |
|----------------------------------|--|
| 1.1 usage | 1.4.2 confusing output |
| 1.1.1 frequent-light use | 1.4.2.1 duplicated labels |
| 1.1.2 frequent-heavy use | 1.4.2.2 missing affiliation |
| 1.1.3 infrequent use | 1.4.2.3 spurious results |
| 1.2 tasks members use database | 1.4.3 interface |
| 1.2.1 membership management | 1.4.3.1 inconsistent actions |
| 1.2.1.1 member records | 1.4.3.2 too complex |
| 1.2.1.2 newsletter (semi-annual) | 1.4.3.2 lack of customization |
| 1.2.1.3 calendar (quarterly) | 1.4.4 lack of confidence |
| 1.2.2 development | 1.4.5 user training |
| 1.2.2.1 event planning | 1.4.5.1 irrelevant examples |
| 1.2.2.2 fundraising | 1.4.5.2 training prior to data migration |
| 1.2.2.3 attendance tracking | 1.6 learning strategies |
| 1.2.3 statistics or reporting | 1.6.1 attending training sessions |
| 1.3 favorable experiences | 1.6.2 trial-error |
| 1.3.1 speed improved | 1.6.3 replaying training recording |
| 1.3.2 more/easier access | 1.6.4 submitting help tickets |
| 1.3.3 intuitive | 1.6.5 user guides/"cheat sheets" |
| 1.3.4 added functionality | |
| 1.4 challenges encountered | |
| 1.4.1 dirty data | |

2. Adapting to the migrated database

- 2.1 manual checking of data
- 2.2 paper archive
- 2.3 using other tools to complete or support some tasks
- 2.4 external help (database experts)
- 2.5 postponing vital functions

3. Users' mental models

- 3.1 data storage
 - 3.1.1 on a server
- 3.2 data organization
 - 3.2.1 metaphor/analogy
 - 3.2.2 physical objects

Examples of coding analysis of transcripts:

"I probably use it every day, but only for a few minutes." [1.1.1]

"At least once a day, sometimes all day I'm on it" [1.1.2]

"... from just logging in, which in our old database literally took ten minutes to log in, and have like tons of passwords, it was awful, it [the new database] was just super easy and each user can have their own kind of profile, so that's cool and great." [1.3.1]

"..., try to see maybe who has stopped supporting us, can we figure out why, ..." [1.2.3]

"look, [x], he's in there twice because he's got two addresses, but it's the same PID number, so, I don't know." [1.4.2.1]

"... put [x] on the top donors of all time, and [x] is clearly not, [y] is. ... I was kind of skimming through the top, and I'm thinking, 'She couldn't have given us that much money', and sure enough, others money was in her account, so we'll have to get to the bottom of it. And I think there are probably a lot of situations like that, that we may not know about until we come across them, ... We have to dig deeper and figure out how

*to A) get it correct and B) how to keep it straight going forward.” [1.4.1]
[1.4.2.3]*

“Another problem, they’re coming to me now! ... when people go to make online donations, they usually do it sort of as a guest rather than trying to log in to see if they have an account, which creates a second record.” [1.4.1]

“multiple people—some people have multiple records anyways because when you accidentally type an extra ‘c’ or an ‘h’ or not the other, you don’t see it, you create a new record.” [1.4.1]

“I mean, I would guess—I don’t know if this is a thing—but just, kind of a spreadsheet, cause I’ve seen, you know, a billion rows, or columns, and then it just says ‘status’, ‘address’, ‘household ID’, just has kind of everything.” [3.2.1]

All interviews were transcribed into text files. Combined with hand-written notes taken by the researcher during the interviews, the text units (a sentence or a paragraph) were coded. This coding schema separates each answer or portion of an answer and attaches it to a research question, as well as defining it within a set of pre-defined values related to the expected response types. Interviews were coded using Provalis QDA Miner.

A tag schema was developed for the field notes using a bottom-up approach based on the type and frequency of incidents discussed during the researcher’s participation with the organization. This schema began as high-level categories such as “front-end” and “back-end” incidents, differentiating between incidents that were related to the software

the users interact with directly and the underlying database or data. In much the same way as the interview coding schema, the tag schema was expanded by adding specificity as more incidents were documented. When new field notes were generated, they were either assigned to an existing category or a new category was created. Because the researchers' database was updated with each new field note, both the database and PHP program required updating when the schema was changed. The final tag schema can be presented here:

1.0 Back-end Incidents

1.1 Database error – Issues regarding the database and extracting data from it.

1.2 Data integrity – Issues with the integrity of the data, either from entry error or as part of the migration.

1.2.1 Legacy data issue – An issue in the data provably or possibly caused by an error in the data prior to the migration.

1.2.2 Migration issue – An issue that occurred as a result of migrating the data from old system to new.

1.2.3 Data entry/process – An issue that occurred with the data due to a process issue or error in data entry.

2.0 Front-end Incidents

2.1 User interface – Issues with the user interface of the system and confusion created from trying to use the system's interface.

2.1.1 Affordances – User interface not intuitive to users.

2.1.2 Labeling/identification – Unclear written description of an entity within the system, either in tooltips, labels, or documentation.

2.2 User experience – Issues with interacting with the database due to confusion about its function or underlying structure.

3.0 Service Provider Related Incidents

3.1 System functionality – Issues with the way the system has been designed that result in errors for the organization.

3.2 Support – Issues in receiving and dealing with the vendor's provided support.

3.3 Training and documentation – Issues with the training and documentation provided by the vendor during and after the migration.

CHAPTER FOUR.

FINDINGS

Overall, the users are positive about the new database despite the challenges, both throughout the migration process and while working with the migrated database. These challenges are mostly contributable to either the design of the migration process or the new database's back-end schema and front-end design. Because this study focuses on the users' experiences, the interviews are presented first, and the field observation notes are used selectively to supplement those findings. Because notes were kept with a very wide scope, not all notes were deemed relevant, and the notes for the purpose of documentation at the technical level were therefore not included in data analysis although they contributed to documentation.

Interview Findings

Use of Database

Four members of the organization were interviewed, and of those four members, three identified as frequent users and one identified as an infrequent user. Frequent users said they use the database at least once a day, while the infrequent user only accessed the database a few times a week. Among the frequent users, two identified as heavy users, making use of the database many times throughout the day for multiple purposes, and one identified as a light user, still accessing the system daily, but not for prolonged or in-depth tasks.

The most common and important task mentioned by the interviewees was the communications, which consist of mailing and interactions with both donors and members as seen in the “Communications Center” (Figure 4.1). This module was used by three of the four interviewees, and by one extensively. This module handles both physical and electronic mailings. Quarterly program calendars and semi-annual newsletters were specific communications tasks mentioned by the participants as important tasks to their jobs.

Being a non-profit organization, services provided to the community are made available at no cost to attendees. The database handles both financial and statistical information about donors, donations, events, and other contributions to the organization in the fundraising area. Statistics and reporting was also mentioned frequently at eight times, and is the only task mentioned by all four participants. The reporting function has many built-in templates for reports that are used for a wide variety of tasks throughout the organization. It also allows for the creation of new reports to extract almost any data out of the database if queried correctly. The database provides a query builder that allows users to select data from the database and display selected data fields, limiting the results based on certain criteria provided by the user. This allows great flexibility in selecting which records are provided in a report, and what information is provided from those records. Because of this, the reports area can expand its possible uses beyond the provided, pre-generated reports. The Report Management Center can be seen in Figure 4.2.

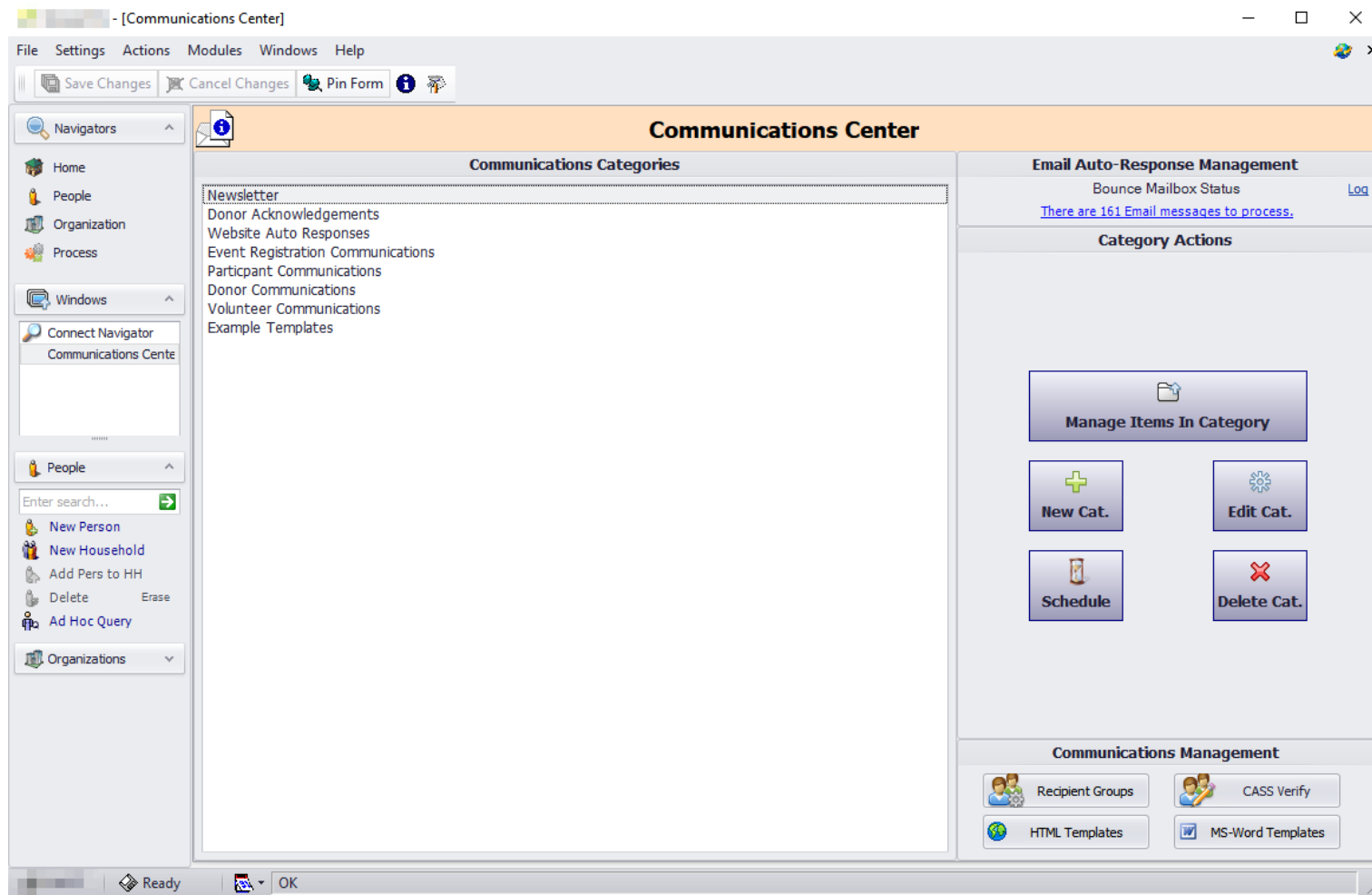


Figure 4.1 User Interface: Communications Center

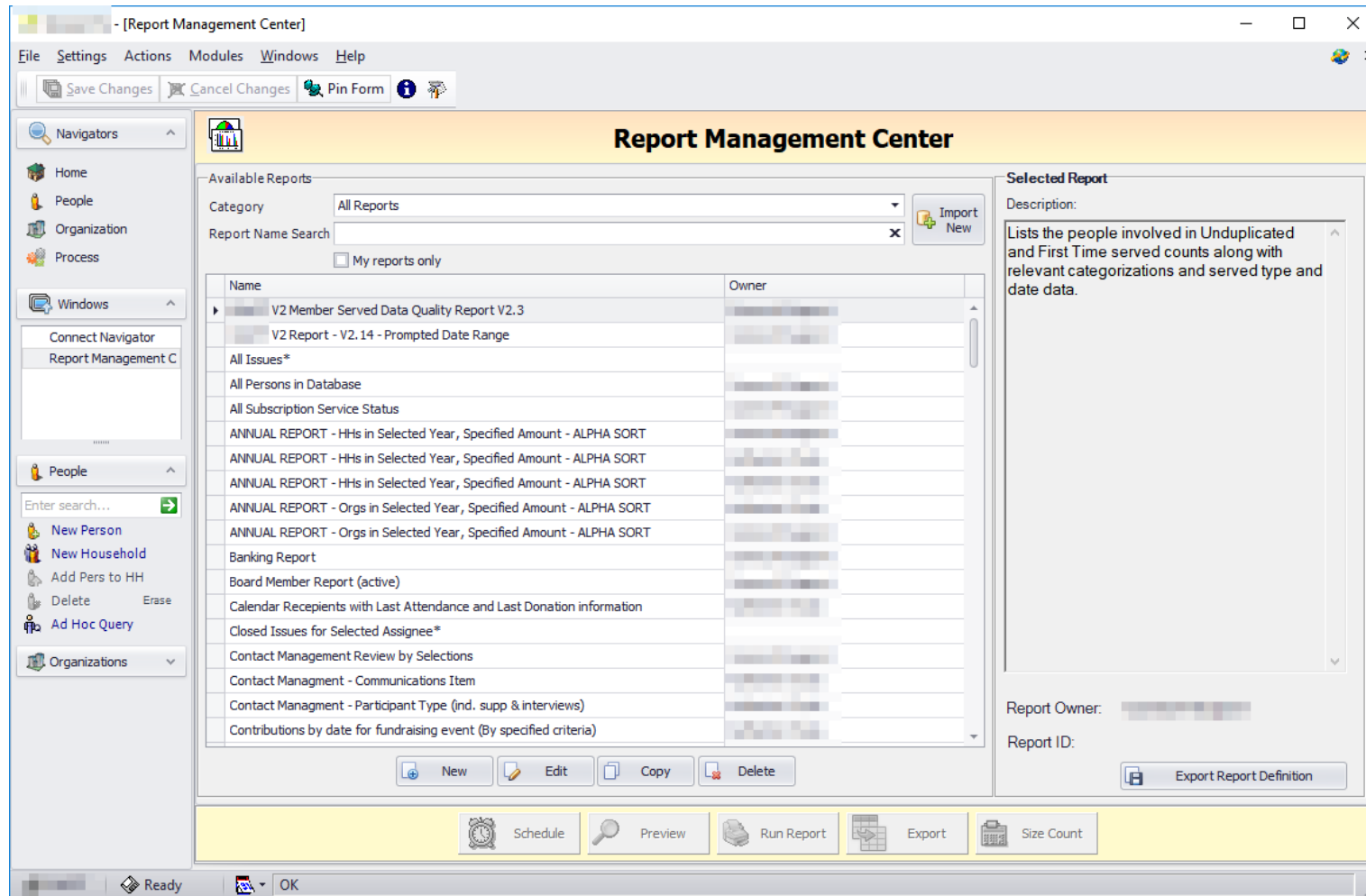


Figure 4.2 User Interface: Reports Management Center

In the field notes, attendance tracking is noted as a high impact activity for the organization, but it is only mentioned specifically in one participant's interview. This participant is largely responsible for both organization and assessment of attendance for programs within the organization. Attendance tracking also overlaps with reporting because the reports area is used to generate the attendance reports created by the database. The event management and attendance area can be found in Figure 4.3.

Member management and member records are mentioned seven times across all four interviews. Member management is used by the organization to view and change member data without using the reporting area. This is one of the functional areas of the database that is available and used by all members of the organization, meaning it sees a wide variety of uses. To accommodate this, it is also very information-dense. Each member's record pulls together information from other areas in the database, such as contributions, attendance, and mailing, into a single location where it can be viewed or changed as needed. An example member record can be found in Figure 4.4.

Favorable Experiences

The participants identified forty instances of positive experiences with the database, most of which had to do with improvements over the existing database. The most common favorable experiences were instances where the users found the new database intuitive and easier to use, especially when compared to the old database.

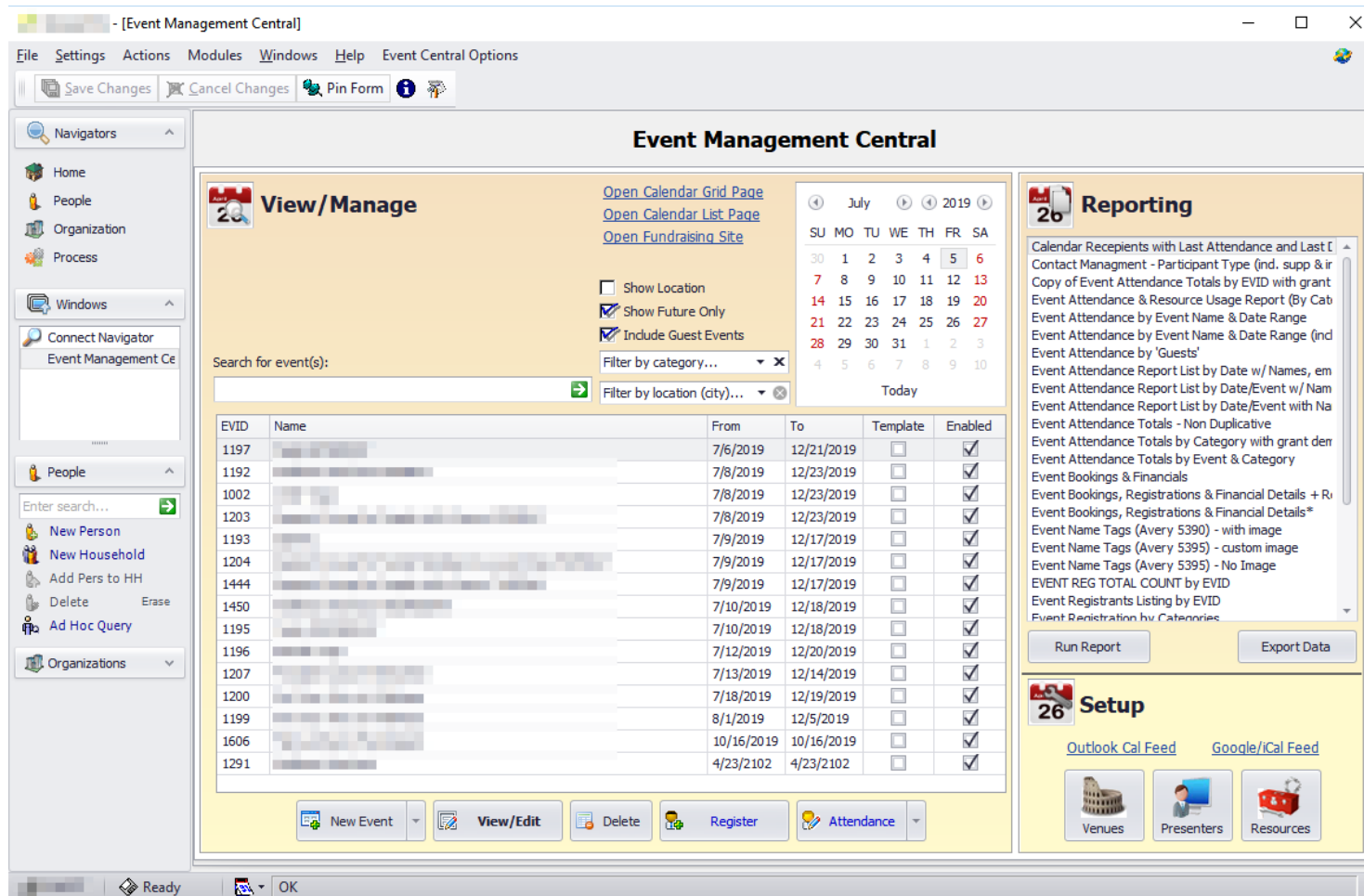


Figure 4.3 User Interface: Event Management and Attendance Reporting

“So when I think of [database], it really is a good database for me, because I can actually use it for more than looking up addresses, and that’s easier than the last database as well. So, I have a generally positive, favorable kind of experience with it, or, thought about it.”

The participants describe how various screens, ranging from the initial login screen filled with shortcuts and recently accessed records to the event management screen filled with upcoming events and relevant reports, are easy to figure out, even when not formally trained on them. The new database allowed the organization to discontinue a service used for emailing members and maintaining mailing lists in favor of an internal emailing. This email module is described by the interviewee as “very intuitive” relative to other modules of the database and the old email system. One participant described the database mailing function as a dramatic improvement because it allows for smart generation of mail items based on attributes in the database.

“Attributes, it’s super easy, it’s just, you know, point and click, which again, makes it really easy for us to volunteer to do some work in this area.”

The second set of experiences had to do with either providing more access to the organization’s data or making that data easier to access. This improved level of access was mentioned eight times throughout the interviews and was mentioned by every participant. The participants noted that the new database was quicker to locate a record, and it was easier to access meaningful data related to that record. Further, strict access control allowed

everyone in the organization, including volunteers, access to relevant records, improving both speed and accuracy of their job-related tasks. One member noted a new feature for grouping member records and easy access to information related to that specific group.

The third set of experiences had to do with speed improvements over the old database. Two of the four participants cited this as a favorable experience a total of eight times. Two interviewees noted the old database could take ten minutes or longer to log in. One interviewee even said that he/she previously had no access to the database because it was too unwieldy or slow to be worth their time and effort for small tasks. The remaining remarks had to do with intelligent navigation or automation within the database. The participants noted that the database would often allow for smart navigation to a related household or organization. It would also intelligently auto-fill based on previous data entry, or allow for easy copies of existing records to be made, which proved useful in generating often-repeated events.

“Everything ... from just logging in, which in our old database literally took ten minutes to log in, and have like tons of passwords, it was awful. This was just super easy and each user can have their own kind of profile...”

Another set of favorable experiences related to new functions available in the new database that were viewed as an improvement over the old database. The participants praised the interconnectedness of the records, noting that relationships between members, affiliations with organizations, and the ability to create internal groups are all new and useful features in the new database. The centralized and connected nature of the data also

improved situations on the financial front, prompting one participant to respond in the communications area to an invoice or donation generated by another participant elsewhere in the database. Financial reports can also be generated directly from the database, which proves highly beneficial when functioning as expected. Lastly, the complexity and lack of crucial functions in the old database resulted in one participant saying that they believe fewer tasks “fall through the cracks” because the new database does such a good job tracking and informing the user regarding key pieces of data.

“[W]e ask them to fill out this yellow form when they first come in, and this system really does a good job of catching people who maybe didn’t fill out their yellow form, but have come to a program. That’s actually huge ... Last year, we thought we had, like, an increase in participation by like 20%, and I think—we did have more people—but I think this database is much better at actually catching people who come to programs than the other one, like, much better.”

User Challenges

The challenges encountered by the participants when using the new database were mentioned a total of fifty-six times throughout all four interviews. These concerns covered a wide variety of tasks in many areas of the database.

In the user interface, the users found increased complexity and lack of customization was mentioned by all four participants and was tied for the most mentioned challenge with eleven. Organization and confusing labels within the reporting area complicates and confuses both the creation of reports and the interpretation of the data received after a report is created. This issue goes directly to the complexity of the database

and the users' mental models of the database. If the users construct a more complete mental model of the database, they are more prepared to deal with the organization scheme provided by the database. Simplification of the "schemas" presented by the database for use within the reporting area was also noted by the users as something that would lessen the difficulty of building and making use of reports.

Because data cleaning was not performed before the data migration, duplicates and "dirty" data in the database were bound to occur. "Dirty" data and duplicated records were mentioned five and four times respectively across three of the four interviews. In order to combat duplicate records, the database provides a duplicate detection and removal function which uses fuzzy search to locate duplicate records in the database and provide tools that easily allow them to be merged or removed. However, even after extensive use of the duplicate removal tool, the users were observing duplicates in reports and mailing lists that did not show up in the tool. This resulted in extra confusion and work for the users until the issue came up with the researcher, who made an effort to track down the source of this problem. After some searching and looking at the evidence, the researcher discovered that the issue was individual records that had multiple addresses assigned to them. In the database, individuals-to-addresses is a one-to-many relationship, which is common in relational databases, but not familiar to users without experience with such databases. This meant that despite there only being a single record for the person, they showed up twice in the reports, once for each address. Armed with this new information, the users were able to remove or choose "do not mail" options for the duplicate addresses, significantly

reducing the occurrence of this problem. Further, by default, the database did not seem to adhere to the “do not mail” option, and upon inspecting the database queries, the researcher discovered that the mailing function did not even look at that field. A custom query had to be created that extracted the information required for mailings and abided by the contents “do not mail” field. Had the users been explained the nature of the relationship between individuals and addresses and included that in their mental model of the database, they would have been better equipped to observe this potential issue and resolve it without external intervention.

The database’s interface was also a source of confusion for its users. One notable issue was with the consistency of function across the database. The participants noted an issue where the database would only save member records when a traditional “save” button is pressed, as seen in Figure 4.5. Further, this button can only be pressed once data in a field is changed, and the user exits the field for the database to recognize the change. If the user closes the record immediately after changing the contents of a field, the record will be closed without saving the data. Elsewhere in the database, the user noted that simply by changing data in a field, it would be saved, even if closed immediately thereafter. No “save” button is even made available on that screen. This inconsistency confuses the users and prevents them from building a coherent mental model of how to save records across the different modules of the database.

Even when tasks were performed correctly, not everything behaved as expected. Three of the four participants identified incidents where they discovered a “bug” or

The screenshot shows a software application window titled "Person & Household Actions". The interface is divided into several sections:

- Menu Bar:** File, Settings, Actions, Modules, Person & Household Actions, Windows, Help.
- Toolbar:** Save Changes, Cancel Changes, Pin Form, and other icons.
- Left Sidebar (Navigators):**
 - Home
 - People
 - Organization
 - Process
 - Windows
 - Connect Navigator
 - People (expanded):
 - New Person
 - New Household
 - Add Pers to HH
 - Delete
 - Ad Hoc Query
 - Organizations
- Main Content Area:**
 - Person (ID=):**
 - Title/First/Mid
 - Last Name
 - Prof. Name
 - Gender: Male
 - Home Phone
 - Mobile Phone
 - Bus. Phone
 - Other Phone
 - Email Address
 - Date of Birth
 - Age Group: Select...
 - Record Type: Active
 - Rel'shp Mgr.
 - Member #
 - Status: Unverified - Gen. Public
 - Period from: 10/31/2017 to
 - Organization Affiliations (Dbi-click to display Org.)
 - Relationships (Dbi-click to display person)
 - Household/Address:**
 - HH Name
 - HH Salutation
 - Acknowledge Contributions: to Household (selected), to Each Individual
 - Select Address
 - Address Name: Primary Address
 - Address Line 1
 - Address Line 2
 - City
 - State: TN
 - Zip Code
 - Address Status Information: Do Not Mail Here, Invalid Address, Address Distance, Not Available
 - Primary HH (selected), Head of Household

A tooltip for "Record Type" is displayed over the "Record Type" field, stating: "The record type field can be used to designate the status of an individual record - such as Active, Inactive, Deceased, etc..."

Figure 4.5 Example of Save Interface

otherwise broken function within the database. One such issue was with data being entered and not saved in the database. One participant identified this issue as a source of frustration given that they would complete the work, but when they came back to the record, the data was gone. It is possible that this could have something to do with the above issue of save inconsistencies, but the current cause of the issues is unknown to both users and researcher. Two participants brought up a related issue where the database is supposed to store a record of completed mailing tasks, including the contents of the mailing, and it does not do so. The user has been in contact with the service provider regarding this issue, and it seems to be a known issue at this time. In the meantime, the user is working around this issue by storing the data manually outside of the database in an Excel spreadsheet. This is another potential cause of losing confidence in the data and requires extra effort for the user.

Customization of the database design and its user interface came up eleven times in different contexts in all four interviews. Among those areas mentioned are the customer-accessible website portal, which one user notes is not supportive to the needs of the organization's potential users, and those users' knowledge cannot be changed by the organization. Even when the database does allow for organization-level customization, as it is with the availability of certain fields and functions within the database, the organization was discouraged from changing those options by the service provider, giving the user the idea that the "intended use" of the database invalidated the organization's preferred methods and existing processes. Lack of customization is also a problem with regard to the reports available in the database. The reports provided by the database are both complex

and unable to be changed even if a user was knowledgeable and so inclined. While users do have the ability to make their own reports, the complexity of the provided reports would make this an onerous task for the user. The users expressed concerns that the reports they have to generate must meet certain specifications, and if the report generated by the database does not match, they cannot change either the type or format of the data in the report. Data exporting integrations were also a feature in the database that did not perform as expected. Integration with online accounting software was a feature promoted to the organization by the service provider, but there were issues with this accounting integration assigning records to the correct accounts, so the organization has changed back to manual reporting of financial data rather than relying on the database's automatic integration.

Mental Models

Of the four participants interviewed, two types of mental models were observed: metaphor/analogy and physical objects. The first model was metaphor/analogy, whereby the users identified a spreadsheet as an analogy for the existing database to describe how the data were organized in "rows and columns." Both users viewed the database as one giant spreadsheet containing all of the information about members, organizations, contributions, etc., and assumed some types of relationships between the rows and columns of the sheets as a record and its attributes in much the same way the database does. One of the participants even went so far as to observe a primary key within the database's table as a record's identifier and mention how that number would connect it to other data in another sheet within the workbook.

The second model is the physical objects model. One participant likened the data in the database to files in a filing cabinet, similar to those used by businesses for decades to store and organize their data. This model groups data appropriately, with all data about any entity within the database being contained within a folder and enumerated on pages within, and folder tabs serving as the identifier for the folder. Some more advanced relationships, such as relationships between members of a household or employment at an organization, can be difficult to understand or result in duplicate data in this model, resulting in potential incongruity between the data and its representation in the database in the eyes of the user.

The other mental model in a physical form was presented by the interviewee who likened the data in the database to the print membership sheet or event attendance sheet, with a vague idea about the mapping of each field within the database to a field on a sheet of paper. The difficulties this user encountered were how to find related records and why the affiliation was missing from the member's record. To support users with the physical objects mental model, the effective perceived affordances for data input should be closer to a paper form and records browsing should be similar to flipping through pages. Given the inclusive all-in-one interface of the migrated database, it is a real challenge for users to transit from the physical objects mental model to a database mental model.

After a year of using the new database, the users have attained a basic level of understanding of the database with two types of user mental models. Each of these models has its own unique benefits and weaknesses. It will be easier for the analogy model (the

spreadsheet metaphor) to transit to a coherent database mental model if appropriate context-sensitive help can be embedded into the database's interface. It might be more difficult for the physical objects model (paper forms or file cabinets) to build a database mental model. For these users, additional training sessions should be considered alongside embedded elements to help build an effective model if they need to use the database for complex tasks.

User Adaptation Strategies

Because of the challenges they faced, the users developed some strategies to overcome those challenges by altering their way of handling certain tasks. To overcome their lack of confidence in the data output by the database, the users have resorted to manually checking data. For example, the organization is responsible for generating a report to their parent organization. Although notified by the service provider that in the database the users can simplify the reporting process, the first version of the report output by the database had a few errors due to dirty data or missing links between records. Although the errors were identified and fixed, the incidents hurt users' confidence in the database. To cope with potential errors, they developed strategy to manually check outputs and reports.

Their lack of confidence also resulted in the continued use of paper records for verifying data and maintaining archives of records. The users still use these paper records as backups for the data already input in the database, such as in the areas of attendance and member records. When members attend one of the organization's events for the first time,

they are asked to fill out a sign-up form for the organization. Additionally, attendance of events is kept on paper as well. These paper records are used both as a backup for the database, and so that the paper records can be checked against the data reported by the database. The database does offer a kiosk module whereby members could check-in to the database directly when attending an event, but this would require the users to have confidence in the data being stored and reported by the database, as there would be no paper records to go back and check. For this reason, the organization has decided to forego the use of this kiosk database entirely.

Users have also used other tools outside of the database to support their job tasks and supplement the functionality of the database. Mailing reports were found to have duplicates of the same person several times throughout the process. When the de-duplication tool failed to detect duplicates that the users were able to find manually, they utilized other tools, like Excel, to sort and remove duplicates before finalizing the mailing lists that occur quarterly and semiannually. Excel also was used to support other tasks, such as maintaining historical records of completed tasks. When using the database to generate a mailing list, it does not provide information of the previous mailing recipients. To keep track of past mailing recipients, Excel was used as an extension of the database.

The users also reached out to the database expert volunteers for troubleshooting and assistance with complex tasks. The researchers' initial involvement was assisting the users with the data migration and learning of the new database. During the ongoing assistance, the researchers realized that the user manuals from the service provider were

too generic to be useful. Therefore, the researchers created user guides to assist with the specific tasks that the users needed regularly, but not on a daily basis.

In summary, users' adaptation strategies include 1) keeping paper archives to check database outputs; 2) using other applications to complement the migrated database in certain tasks; 3) postponing implementation of modules; 4) getting help from database expert volunteers.

Field Observation Findings

Field observation notes were created by the researchers throughout the migration process on a wide variety of topics and incidents. The field notes contained in this section were selected to supplement the topics covered by the interviews with the participants, as those were deemed most relevant to the users' experiences during and after the migration.

Database Back-end (Data Schema)

Following the initial data load by the service provider, numerous issues were discovered. Some had been identified by the users, but more were discovered by the researchers while investigating the new database and addressing problems discovered by the organization. Several problems identified by the users in the interviews as "dirty" data can be attributed to these migration issues. The researchers were provided with access to the raw database file containing all of the data normally only made accessible to the users through the interface. The researcher observed that the database was extremely large, containing 182 individual tables. Of those 182 tables, only 85 had data in them, and only 54 had ten or more rows. This confirmed the fact that the database had been built as a much

broad, general database for use across many organizations, and little had been done to tailor the database to the needs of this particular organization.

Member records, being the most important and central entity in the organization's structure, proved a common source of problems. The database stores members, households, and addresses in individual tables with one-to-many relationships. This means that one member can be a part of two households, each of which has three associated addresses. Notably, there is no duplicate detection in place for the address field, which was ultimately the cause of many "duplicates" in mailing lists that did not actually have duplicate member records.

As with membership data, organizational records within the database also contained duplicate and incorrect data. Upon reviewing the old database, the researcher noted that the database was not fully normalized and organizations were not their own entity within the database. Each member in the database might have an organization attached to them, and when two members were attached to the same organization, no connection was made between those members. This resulted in many duplicates within the new database after the migration. Any organization that had been listed in the old database was created as a new, single entity connected to its member in the new database. This meant that, for example, a bank with numerous financial supporters of the organization now had seven entities in the database, all with the same name and address. However, a lack of consistency with regard to name and address, as well as the fact that other branches of the same bank with different addresses made merging these organizations in the database

difficult. No merging tool was provided by the service provider in the way that one was provided for merging members. Ultimately, this had to be reviewed and fixed manually by the users.

One of the most important tasks the organization completes is mailing, primarily of bi-monthly calendars and a bi-annual newsletter. The organization allows the members to decide whether they will receive these communications, and in many cases, multiple members of the same household may opt-in to receive the letter. In the new database, this would result in multiple entries in the report generated pointing to the same address. As noted above, some households contained multiple addresses, and some even contained the same address multiple times. Due to these problems in data output, the mailing list from the database needed to be cleaned before it could be used. Identifying the problem and finding a solution to the problem took several months following the migration. One of the solutions was to use the database's query module to output unique records and code the field as "do not mail" for the addresses that should not be used for mailing. The solution involved analyzing the data structure to find the existence of the field that flags which address to mail or not to mail, and to remove duplicates from the resulting query. The custom report can be found in Figure 4.6.

Database Front-end (Interface)

In the member management area, the interface defaulted to displaying a form containing many tabs, most of which were not used by the organization. Many functions are presented to the user, such as updating records, adding addresses, connecting

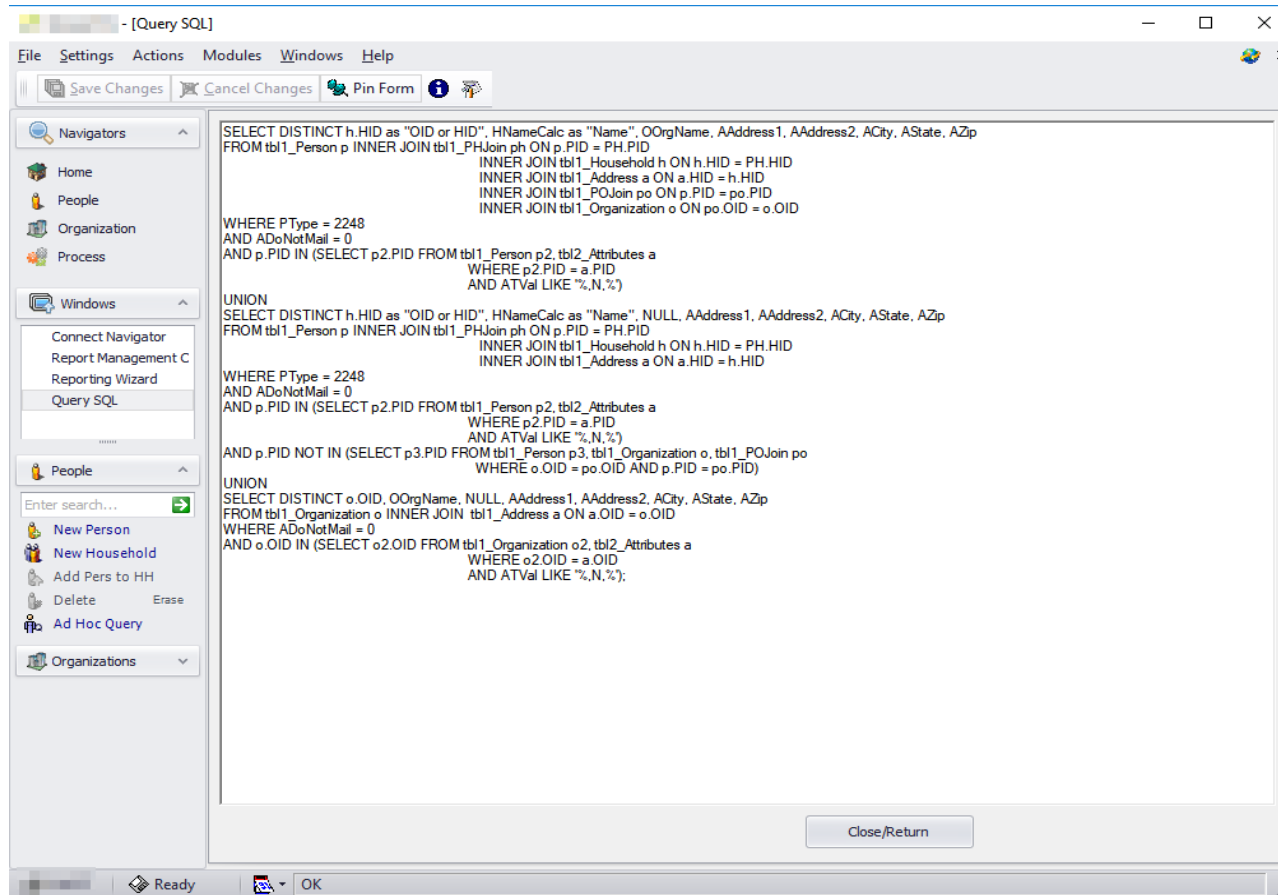


Figure 4.6 User Interface: Advanced Feature for SQL Query Builder

households and relatives, etc. This suggests that the interface is inclusive with all member-related functions accessible from this area. As a design trend, making all functions visible could improve efficiency for certain users but can be overwhelming to new users. In the early stages, the users did not consider the database easy to use because it is not possible to figure out what all these tabs will do and how to use them in relation to their tasks.

One of the challenges facing the users was how to extract data to generate outputs. The Reporting Wizard (Figure 4.2), also called Reports area, was a central location for querying data from the database, whether it be financial statistics, attendance reports, or newsletter mailings. The reports area is a “wizard” style design for creating new reports, but this wizard is based around a “query builder” and several “schemas.” While a “schema” is familiar to those well-versed in the use of databases, its meaning differs in the context of this database, and further, it was a foreign concept to the end users. Additionally, the purpose of each of the schemas was unclear to the users. These are the schemas as presented to the user by the database:

- Persons Grouped by Households & Related Info
- Volunteer Applications & Meetings
- Volunteer Assignments
- Persons with Household & Organization & Related Info
- Fundraising Information
- Event Attendance
- Attributes Management (System Administration)

- Organizations with Primary Contacts
- Organizations with Contacts & Financial History
- Event Sponsorship Administration
- Database User Information
- Opportunities/People/Organizations
- Volunteer Positions with Assignments
- Reference Data (System Administration)
- Event Bookings Communication & Financials
- Volunteer Positions
- Communications Analysis
- Opportunities
- Inventory Management
- Event Administration
- Event Registration Administration
- Resource Books & Related Events
- Event Presenter Administration
- Member Number Management

Once selected, each schema can be used to select fields from the database through an in-depth tree menu where a user is tasked with selecting criteria for a query to determine what data will be contained within the report, and then uses the same schemas and menus

to select which fields should be displayed from the rows selected. In order to access the necessary data to be available in the dropdown, however, the users must select the correct schema from the dropdown, and the only clarifying information about the schema and its contents are the names themselves. An example of a query might be a report that lists only members whose donations exceed five thousand dollars; the output from the database includes donor's name, mailing address, and last active date. An example of the query builder and its drop-down menu can be found in Figure 4.7. From the figure, it is clear that the database wizard process is quite complex because at each step, the user must select from a list of alternatives.

Migration Process and User Training

Several issues can be contributed to the migration process in terms of sequence of implementation, or the lack of customization in database interaction design. There are rigid steps for completing tasks that may not be the preferred process for users who regularly perform these tasks. Either the users could not change the task's process or they were discouraged from changing the design. For example, the users knew that they could hide certain tabs in the interface and they would like to do so for the unused tabs, but they were instructed not to hide them. Because the database was not designed exclusively for this organization, adopting this database requires the organization to change its task processes. This was confusing for the users, and further, the training provided by the service provider neglected to fill in these gaps.

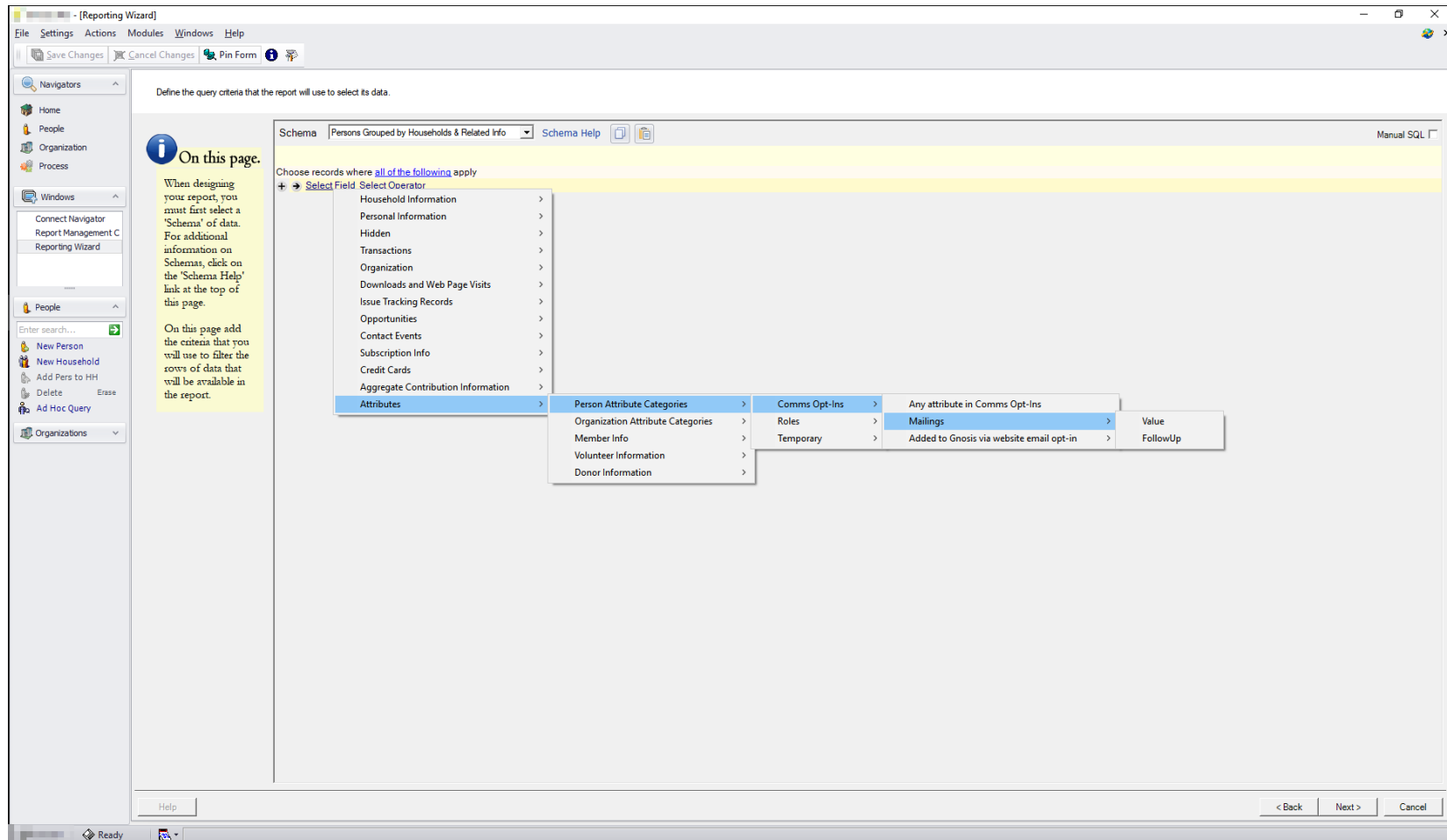


Figure 4.7 Query Builder

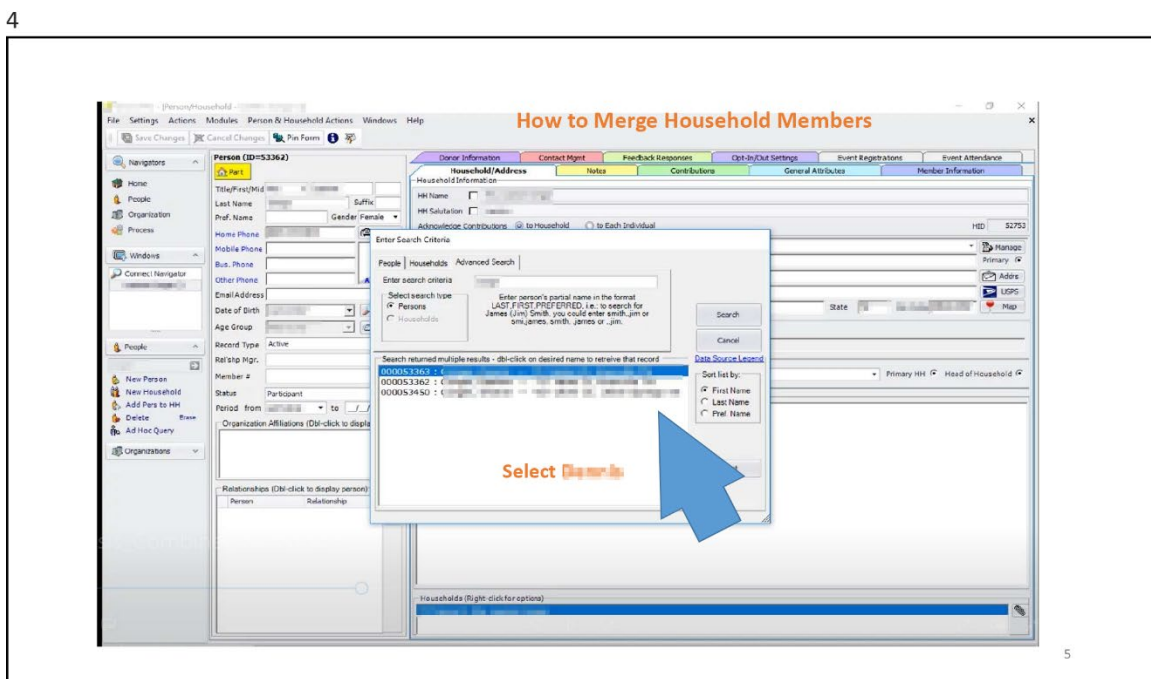
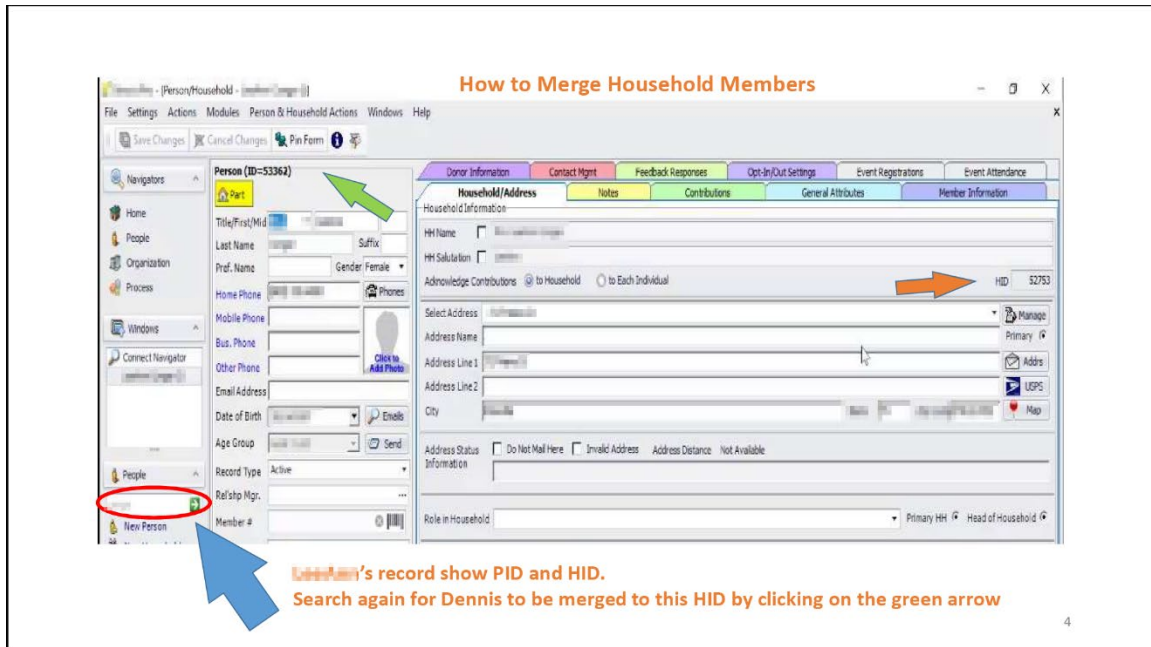
The online training setup was complex, requiring users to call in using a telephone (with limited lines) and connect using a computer with the database client installed, allowing them to connect to a web conferencing server. The session was not recorded for the attendees. The researchers were able to attend several training sessions with the organization's users to learn about the new database. Although the company does provide standard training videos for users, these videos were not customized to the specific database application. Once the researchers realized the need, they began to record the training for the users using Zoom web conferencing software; these recordings were valued by the users as they replayed sections they wanted to review.

The training sessions were held when the data migration was still ongoing. This was a disadvantage to the users because the training utilized an empty database or a different demonstration database to enter data or provide examples. Had the data migration been completed, the training sessions would have used examples from the database that were familiar to them. Further, some of the issues, such as duplicate records or empty fields in the records, would have been easily spotted during the training.

Documentation for this database application was not provided, although there were manuals about the database (that other organizations also use) accessible from the provider's website. These manuals are too complex for the current state of the migrated database. Users occasionally felt confused about the database and wanted to have other documentation to consult. To fill this gap, we created four user guides, affectionately referred to as "cheat sheets," with screenshots and step-by-step explanations on the

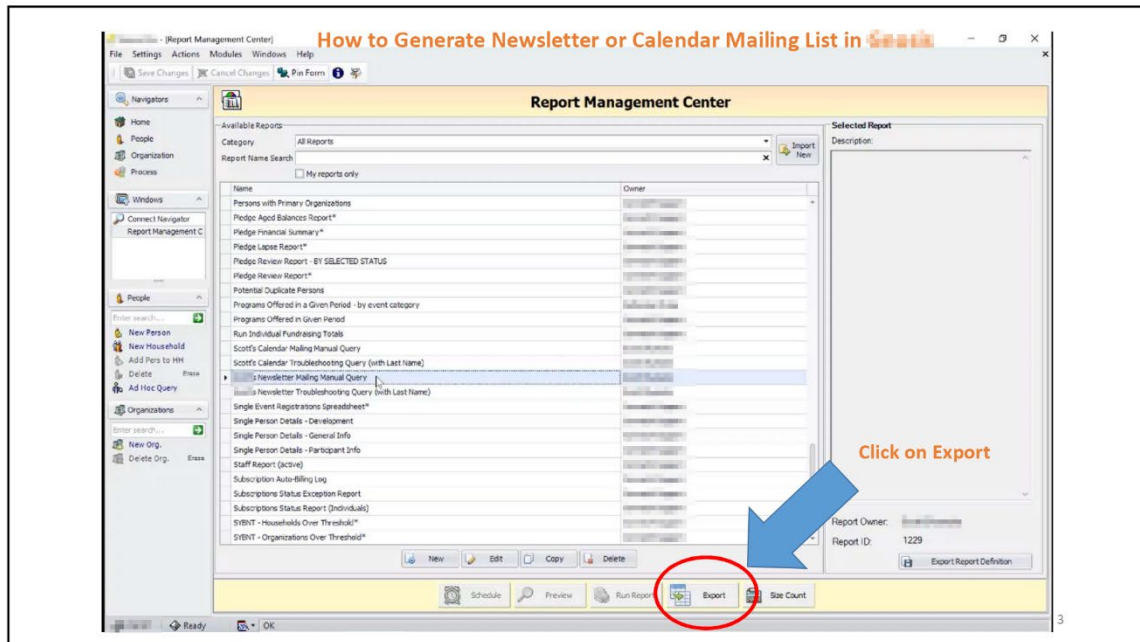
important tasks. Excerpts from two user guides can be found in Figure 4.8 and Figure 4.9. These guides provided visual aids and written instructions with directions regarding what steps should be taken to complete the tasks specified. Feedback from users regarding this style of user guide was very positive.

Technical support was free and unlimited during the migration period. After migration, the organization could be charged hourly for technical support beyond “basic” support. It is not clear what is considered basic support and what is considered billable support. Because of this, the organization saw it as imperative to solve as many problems as possible within the transition period. During the unlimited support period, the support team was not able to solve all of the reported problems, and fortunately, the provider promised to honor the free technical support for issues reported during the migration period. One unsolved problem discovered during the migration centered on member’s organizational affiliation. Because member’s affiliations were entered as a field in the member’s record in the old database, a substantial amount of duplicated organizations were migrated to the new database; name variations of the same organization also occurred in different records. A true relational database must ensure that each organization has only one record with a unique identifier. For a member to be affiliated with an organization, the field for the organization in a member’s record takes the organization’s identifier so that the name of the organization will be consistent across all the records where the organization is linked. At this writing, this issue has not been resolved.

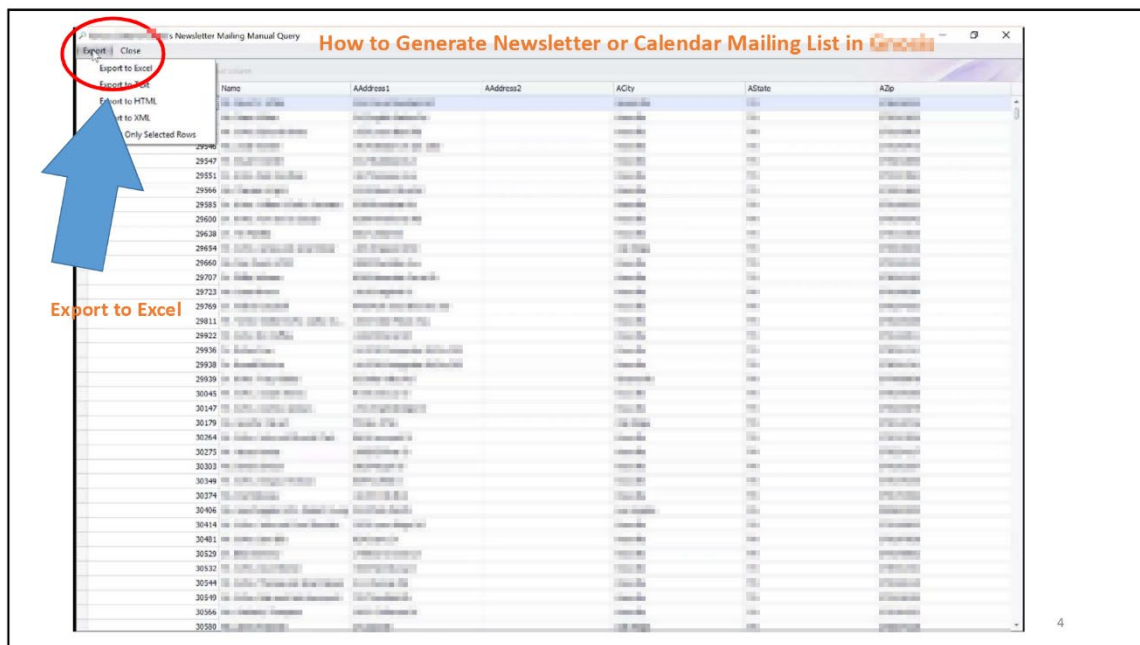


5

Figure 4.8 User Guide/ "Cheat Sheet" Created by Database Volunteers



3



4

Figure 4.9 User Guide/“Cheat Sheets” Created by Database Volunteers

CHAPTER FIVE. DISCUSSION

What are the users' experiences in a database migration?

What tasks do users utilize the database for?

The most frequently used task in the database, given the postponed implementation of several modules, is communications with both members and donors about events, programs, newsletters, fundraising, and acknowledgements. There are different types of members requiring communications to be member-group specific. It is important for the database to produce accurate results because an error could be detrimental. The importance of accuracy and timeliness of these communications is well spoken by an interviewee:

“[I]f someone gives us money, you know, it’s just, you don’t want to send it out wrong. You want them to think, ‘Ok, I know who you are, I appreciate what you did’.”

Because this organization is a non-profit, fundraising was the second most-mentioned function needed for the database. Maintaining open lines of communication and accurate records with respect to fundraising was mentioned several times by the users. The database handles not only direct donations, but in-kind donations of objects for fundraising as well; the latter is similar to a silent auction during the organization’s annual fundraising event. Fundraising emphasizes the importance of the integrated nature of the database, as there are member records, financial information, and communication in the form of thank you letters all resulting from a single interaction with the organization. Donations may be

made to honor someone who may not be a member of the organization. Donors may also prefer to be anonymous. There are many scenarios that make these database function more complex. When the database worked as expected, the users referred to it as a great improvement over the software previously used. However, there were scenarios where the database could be improved. For example, when one donation was meant to be split between two donors, the database could not handle it directly. The users needed to find a way to enter the donation manually as two separate donations.

Reporting in the database serves dual functions: it fulfills the responsibilities of providing information to financial institutions and the parent organization, and it acts as a way for users to create custom reports to meet specific informational needs during the course of their work. With all of the data stored in a central location, the organization has the ability to use these custom reports to extract data from the database to solve problems. However, the system provided by the database to create these reports can result in confusing or inaccurate results if used improperly, resulting in confusion or errors in the eyes of the users. For example, as reported in the interview quote coded as 1.4.2.1 and 1.4.2.3, output from the system is inaccurate, which could contribute to the users' lack of confidence in the database. A number of large donations from an organization at which the member is an employee were misattributed to the individual. It was only through the knowledge of the user that the error was corrected. Without the expertise and keen eye of one of the users, this error may not have been corrected, as the database has no way of indicating this kind of error.

A common thread among many of the incidents documented by both organization and researcher is that a thorough cleaning of the data prior to the migration could have prevented or eased the difficulty of many of the challenges encountered. Ideally, this could have taken place in conjunction with the migration team of the service provider, but many of the steps could have been undertaken by the organization, by at least partially making use of Land's (2013) content specialist and information architect team members on the migration team. These team members would bring both familiarity with the data and a working knowledge of organizational tasks and practices to the migration process, meaning that the resulting data and new processes would be more closely tailored to that of the organization. Mailing issues, lack of confidence in the data, inaccurate reports, and lack of proper handling of both member and organization records within the database could all have been improved through a data cleaning step. Further, Bilal's (2014) recommendation to load and test the data, identify errors, solve problems, and then retest the data before the final migration would have provided an integral middle step to the process whereby problems could be identified before the users are expected to make use of the new system for regular tasks. When the users depend on the system for their job tasks, there is much more pressure than in a testing phase. Extending and improving on this testing phase would likely provide considerable benefit. It's worth noting that this process would likely add time to the migration process, though many factors regarding both data and available staffing would affect exactly how long this might be.

What aspects of the migration do they view favorably?

The users revealed that despite many challenges encountered using the system, there were many favorable aspects as well. Some sections of the database were seen as intuitive and easy-to-use for users. These sections communicate their functionality to the user in an effective way and allow the users to utilize the database's tools without confusion.

Users addressed their improved access, either allowing them more access, more data, or faster access to the data. In such a small organization, the users noted that it was an improvement that each member can access basic member records and make use of that data, rather than relying on one or two members with access to retrieve it for them. Further, they noted that one of the reasons users were previously restricted from using the old database was the complicated login procedures and extended login time involved in accessing the system. These elements of a modern, connected database were perceived as a plus for the organization. Users are afforded the opportunity to perform more tasks themselves without reliance on others, and other members of the organization are spared the responsibility of having to fulfill requests for others.

The new database allowed the organization to perform more tasks and integrate more capabilities than the old database. The new database allowed them to not only migrate their data, but consolidate tasks such as reporting, mailing, and financial management into a single location. Though there were early difficulties with the mailing system, the users expressed that they did like this consolidation with regard to mailing, especially the

electronic mail system. The users did identify some ways in which the system had “bugs” or broken functionality, but still seemed to view the change as a net positive. Previously the organization was paying for and maintaining a separate contact database to perform mailing. This resulted in some tasks, such as updating member addresses or adding new members, requiring the user to change data in two places.

What challenges do they face?

The users identified over fifty challenges they encountered in working with the system or being trained on it. However, many of these challenges share common root causes. The first is with regard to the data itself: no cleaning of the data was performed prior to the migration, and the organization was largely uninvolved with this step in the migration process. Singh (2013) includes data preparation as a recommended step in a migration process, and Bilal (2014) recommends cleaning, testing, and re-testing data during and after migration to the new system. While the service provider may have performed some of these steps behind the scenes, little to none was observed by the organization and researchers. This led to incidents after the migration directly attributed to the data. Some were inherited from the old database, and others were created by incongruities in the migration process itself. These errors in the data itself led to a significant number of the challenges encountered by the users. Notably, the migration was not the only source of data errors, as the researchers observed occurrences where the users found new data that had been entered completely in the new database that was duplicated or incorrect.

The second root cause of multiple challenges was the different mental models users had developed throughout the training and their use of the database for their regular job duties. The database asks users to construct queries without a thorough understanding of the underlying data structures. This was noted as particularly challenging by the participants. Further, the query builder uses the aforementioned schemas to present the users with fields from the system, but no concrete way to determine which schema was correct for the data being sought. Even in more routine tasks, users employ their mental models in problem solving of everyday tasks dealing with data. Even a relatively trivial task like entering a new member record into the database requires an understanding of the underlying data. To create a new record, users must first search to see that the member does not already have a record, and in order to assign them a household, that a spouse or roommate does not already exist within the system as well. Knowing how to properly perform these searches and interpret the results of the searches is important in preventing bad data from being entered into the system. The users identified several times where data entered after the migration was entered incorrectly, likely as a result of problems such as these.

Customization was a recurring issue throughout the migration and was mentioned by the users directly or indirectly throughout the interviews. The organization, labels, and functions within the database software were all defined without input from the users. Just in the interviews, users identified eleven times where confusing labels or organization caused them trouble in making use of the database. While training is also an essential

component in preparing users for making use of the database, customization to the organization's job tasks and information needs was also lacking from the migration strategy observed. The service provider had an existing database that was used as a framework for the database provided to the organization, and the users were expected to learn and adapt to the new database's functions and terminology. This meant that even for tasks that did not require adaptation, users were sometimes confused by how the software labeled certain functions because no consideration or alteration was made based on the existing culture of the organization and understanding of the users.

Although the new database was viewed favorably, the complexity requires time and effort to get comfortable. If there was not enough time for learning, users could feel frustrated because they were expected to complete the same tasks in the new database efficiently.

How do users learn during migration?

One of the most interesting comments found in the data was that the participant still did not feel confident and well trained in the system, even after a year of daily use. In the course of the discussion following training, the user expressed concerns that system was still not well understood. It was a common sentiment that the training did not teach in a way that was conducive to learning for the users, and an understanding of the complexity of the underlying system was not embedded in those training sessions. Because of this, training was a central and consistent point of contention for all of the participants. Several key factors were identified by the users as making the training less than useful for their

needs. First, the training was conducted in long sessions over the phone. One of the training sessions took place on a day where inclement weather prevented the users from being in the office, so the training was conducted on a day-long phone session from their own residences. Users described this as especially difficult, and that they were unable to maintain their attention and retain the instructions presented to them. Further, the instructions were given more as step-by-step instructions than attempts to form conceptual understandings of how the system is constructed, which contributed to the lack of understanding. Second, training was conducted without screen sharing or other visual aids. While the training sessions did focus on how to complete tasks within the system, rarely, if ever, were visual aids or screen sharing used so that the users could observe the tasks and how they were completed. Third, it was conducted before the users had access to the system, and therefore could not learn by practicing what they had learned on their own data. The users described themselves consistently as those who learn by doing. Because they had no access to the system or did not have any data within the system, they had no way to practice the tasks they were presented with in the training and therefore did not retain the information as well. Lastly, the trainings were not selected by the members of the organization and did not always focus on features relevant to their jobs, or even functions utilized by the organization. An entire training session was spent on the kiosk system for attendance at the behest of the service provider, but the organization did not use or intend to use this kiosk system, both for the sake of their internal processes and the ease of use of their patrons.

Recordings and documentation were also concerns that further exacerbated the training problem. Users were told that the training sessions would be recorded so that they could be reviewed later when the users had questions or wanted to practice completing tasks. However, no such recordings were ever provided by the service provider. Further, instructional videos were provided but were few in number, and no official documentation existed at the time of the observed organization's migration. Since that time, the service provider has built a knowledge base (Figure 4.10) and made it accessible to all users of the system that contains both written and video guides to certain features within the system, but the resource is still being developed at the time of publishing. In one of the conversations with the users,

"Here we are, over a year later, and you'd say that you still don't feel comfortable with the system?"

"That's correct."

How do the users adapt to the new database?

Four main strategies were utilized by the users to adapt to the new database. At the organizational level, postponing implementation of certain modules until the database was well structured and data were cleaned. This decision resulted in implementing functions in other specialized software applications, although the organization had to maintain data in both the database and the specialized software application.

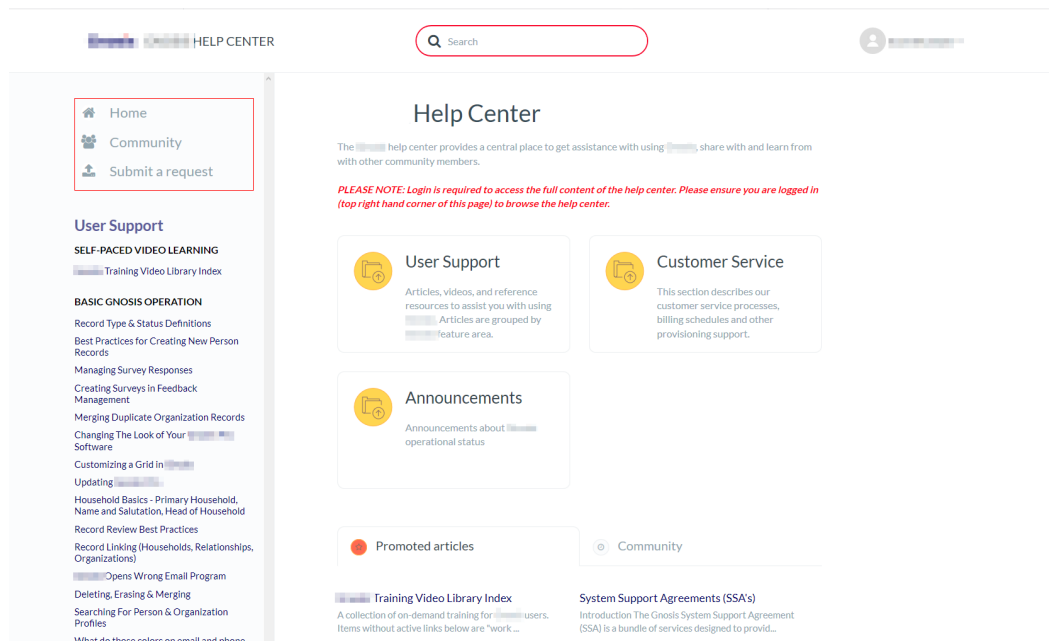


Figure 4.10 Knowledge Base / Help Center

The interviews showed three strategies to adapt to the new database. The first strategy was the use of external tools to complement the functions of the database that the users found difficult to use or lacked confidence in the results. Excel was the tool most commonly used by the organization to document or analyze data outside of the database. Import and export between Excel and the database was not difficult. Excel was used to keep a log of mailings, and to analyze past mailing recipients and to search for duplicates.

The second strategy to overcome the lack of confidence in the database was manually checking the data in the database and in the output. The reports, event attendance, and mailing lists were checked manually by members to catch errors using the paper archives they keep.

The third strategy was to keep paper records in an archive. There were file cabinets full of these archives. These paper records were used for the second strategy to manually check data accuracy.

“[I]f I checked it once and there were no errors then I might start to feel confident, but there’s so many that I feel like I have to go back and check every single one”

What are the users’ mental models of the new database?

There are two kinds of mental models discovered in the interviews with the users: metaphor/analogy and physical objects. The metaphor/analogy users referenced other familiar software systems to explain their mental models of the system. These users have knowledge of Excel and/or other spreadsheet software, and therefore spreadsheets were an

easy means of explaining their mental perception of how the data was stored. Because the database also supports exporting of data to Excel files, especially from the reports area, this likely reinforces this notion that the data is stored much like a spreadsheet might be.

The other two users were more familiar with physical mediums for storing data, which led to their descriptions as mental model metaphors for physical objects. The filing cabinet metaphor is more robust and begins to encompass relationships between pieces of data, organizations, and individuals, while the form model does not document those relationships. Understanding that these users had developed this kind of mental model or attempting to provide the users with a more robust mental model could potentially have benefitted both user and service provider during the migration process.

How do users' mental models impact the use of the new database?

After one year of learning and interacting with the new database, the users still do not have coherent mental models of the database. The gaps between the database's conceptual model and the user's mental model impact how the database is used, especially troubleshooting problems or extracting the data. Examples of this include the users' challenges involving duplicates and understanding the difference between duplicate records and duplicate rows in a report that has other causes, such as multiple addresses. When viewing the data as a paper form or file in a cabinet, the user might assume a one-to-one relationship between members and addresses. When entering a new organization, the user needs to search for an existing organization instead of directly entering the name like one might on a paper form. The user might be confused trying to decide which

organization to choose from the entries. Troubleshooting the duplicates or understanding the difference between a paper form for membership and online form for membership will require a basic understanding of a one-to-many relationship between one member and the member's several addresses; or one organization and several members who are affiliated with it. In addition, correcting or deleting records in a database can result in error messages because of the constraints of associated records. The database stores the query for repeated tasks, but building the initial query is challenging without knowledge of database schema and query language.

The complexity of the database is also due to the fact that the schema must accommodate the needs of many similar organizations, which discourages users from customizing the database application to this organization's needs. Therefore, the interface is not intuitive for some tasks users need to complete. The complexity also makes developing a coherent mental model more difficult.

Many of the challenges encountered in the migration process could have been avoided according to the recommendations in the literature reviewed in this thesis. The single most recurring issue in many migration projects was data cleaning. The inaccurate outputs the users encountered when the database was first used were due to the dirty data, duplicate records, and name variations. Because the migration imported the data first, the deduping tool was used to identify the duplicate records. However, the tool could not identify all the problematic records. A more robust process would be to identify the dirty

data and correct them, then transform the data structure to match the new database's data structure before importing data.

Challenges in learning can be minimized if the real database with real data is used to help users understand the database rather than empty or demonstration databases. It is important to make the connection between the tasks they need to perform and the database interface design to support the task. Young (1981) suggests that a training model based on understanding and building a coherent mental model is more useful when dealing with complicated information systems. Databases require more than memorizing steps to complete tasks, and training would be more effective after the migrated database is functional.

Given the complexity of an inclusive interface, users need to hide unused functions (tabs) to avoid cognitive overloading. Without being distracted by unused functions, the users can focus on the functions they need to master. This can also help improve users' learning experiences.

Limitations and Contributions

Because this study observed a single non-profit organization, only four employees were database users who participated in the interview. Although this is a limitation, it is also a strength of the study because the researchers were volunteers and have interacted with the users since the beginning of the migration. All interviewees were very open in sharing their experiences and thoughts. The interviews were in-depth. Although mental models are important to work with a database, the researchers' observations of mental

models are limited in interviews. Other methods to follow users' mental model development over time are needed to strengthen this line of research.

The findings from this study, although based on a real-world migration project, are limited to its organizational context. Therefore, the experiences and lessons learned from this project may not be directly transferrable to other non-profit organizations. If an organization would like to adopt this thesis' recommendations, they need to evaluate their situations to make decisions.

Further studies of database migrations might find these research methods helpful for systematically observing the process and conducting semi-structured interviews with the users. The researchers encourage organizations to collect data and share their experiences in database migration projects. When more studies become available, systematic reviews will identify best practices and models for robust database migration projects.

CHAPTER SIX.

CONCLUSIONS AND RECOMMENDATIONS

Overall, the users are positive about the new database. They considered the migrated new database a significant improvement over its predecessor: 1) multiple users can access the database, and each user has their own login and password; 2) the response time is much faster than the old database; 3) the new database integrates several functions that handled by other applications, which increases efficiency.

There have been some challenges along the way. The issues related to data integrity and errors in records affected the users' confidence in the database. However, the users developed four strategies to adapt to the new database: 1) postponing implementing certain functions; 2) using familiar applications to complement the database; 3) keeping paper records as archives; 4) manually checking database output.

After one year of use and training, the users have not developed a coherent mental model that will be vital in interacting with a complex information system. Based on the findings and user experiences from this database migration project, we propose the following five recommendations for a robust database migration process:

1. Cleaning data and transforming the data structure before importing data into the new database. This will require a well-planned pre-migration process to identify existing data problems, aiding in the development of strategies for cleaning data and transforming the data structure. Using these strategies will ensure the integrity of the data during migration. Several different processes are involved in this data cleaning and were observed in one form or another in this study.

2. Incorporating elements to help build effective mental models. Users with coherent mental models are proficient and able to troubleshoot problems. This could be accomplished by training and by providing context-sensitive help. To incorporate elements in training, the training sessions should be designed with users' tasks in mind and should make connections between system features, data, and tasks. Providing context-sensitive help in addition to a help center can make knowledge accessible at specific interaction points at which users likely need help. It can be a mouse-over definition of a concept or a function, a list of suggested actions, or a pop-up window with examples. Learning is most effective while doing; having knowledge directly accessible from the database can accelerate mental model building.
3. Training should use the real migrated database. Scheduling training before the completion of the migration does a disservice to the users because the examples may not be relevant to them; they also miss the opportunity to test the migrated database. Moving the training to after the data migration allows the users to see and try the database they will use. Further, if real data rather than demonstration data was used to explain the database, the users will have more familiarity with the data and likely learn more from it. The users would also be able to log in to the database and follow along with the training, which will give them hands-on experience with the database prior to when they are expected to begin working in it.

4. Customizing the interface to hide functions not currently in use. This is also important for users to avoid cognitive overload and focus on relevant functions.
5. Creating task-based user guides for the application. The general user manuals are bulky and too much to comprehend for new users. The migrated database should have its own user guides.

LIST OF REFERENCES

- Balas, J. L. (2011, October). How they did it: ILS migration case studies. *Computers in Libraries*, 31(8), 37.
- Beaudry A., & Pinsonneault A. (2005). Understanding user responses to information technology: a coping model of user adaptation. *MIS Quarterly*, 29(3), 493. <https://doi.org/10.2307/25148693>
- Bilal, D. (2014). *Library automation: Core concepts and practical systems analysis* (Third edition). Santa Barbara, CA: Libraries Unlimited.
- Wu, Bing, Lawless, D., Bisbal, J., Richardson, R., Grimson, J., Wade, V., & O'Sullivan, D. (1997). The Butterfly Methodology: A gateway-free approach for migrating legacy information systems. Proceedings of the Third IEEE International Conference on Engineering of Complex Computer Systems (pp. 200–205). Como, Italy: IEEE. <https://doi.org/10.1109/ICECCS.1997.622311>
- Bisbal, J., Lawless, D., Wu, Bing, & Grimson, J. (1999). Legacy information systems: issues and directions. *IEEE Software*, 16(5), 103–111. <https://doi.org/10.1109/52.795108>
- Borgman, C. L. (1986). The user's mental model of an information retrieval system: an experiment on a prototype online catalog. *International Journal of Man-Machine Studies*, 24(1), 47–64. [https://doi.org/10.1016/S0020-7373\(86\)80039-6](https://doi.org/10.1016/S0020-7373(86)80039-6)
- Carroll, J. M., & Olson, J. R. (1987). *Mental models in human-computer interaction: Research issues about what the user of software knows*. Committee on Human Factors, Commission on Behavioral and Social Sciences and Education, National Research Council. Washington, DC: National Academy Press. <http://www.nap.edu/books/POD266/html/>
- Clancey, W. J. (2006). Observation of work practices in natural settings. In K. A. Ericsson, N. Charness, P. J. Feltovich, & R. R. Hoffman (Eds.), *The Cambridge Handbook of Expertise and Expert Performance* (pp. 127–146). Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511816796.008>

- Clarke, L., & Morris, A. (2009). Library system migration: A case study of change management at Oxford University. *Libri*, 48(3), 153–162.
<https://doi.org/10.1515/libr.1998.48.3.153>
- DeSanctis, G., & Poole, M. S. (1994). Capturing the complexity in advanced technology use: adaptive structuration theory. *Organization Science*, 5(2), 121–147.
<https://doi.org/10.1287/orsc.5.2.121>
- diSessa, A. (1986). Models of computation. In D.A. Norman & S.W. Draper (Eds.): *User-Centered system design: New perspectives in human-computer interaction*. Hillsdale, NJ: L. Eldbaum Associates.
- Estelle, L. (2017). Software migration gone wrong; or, how I learned to stop worrying and love the finance department. *Alki*, 33(1), 6–8.
- Flanagan, J.C. 1954. The critical incident technique. *Psychological bulletin*, 51(4), 327–358.
- Gentner, D., & Stevens, A. L. (1983). *Mental models*. New York; London: Psychology Press.
- Gholami, M. F., Daneshgar, F., Beydoun, G., & Rabhi, F. (2017). Challenges in migrating legacy software systems to the cloud — an empirical study. *Information Systems*, 67, 100–113. <https://doi.org/10.1016/j.is.2017.03.008>
- Hainaut, J.-L., Cleve, A., Henrard, J., & Hick, J.-M. (2008). Migration of legacy information systems (pp. 105–138). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-76440-3_6
- Hallmark, J., & Garcia, C. R. (1992). System migration: Experiences from the field. *Information Technology and Libraries; Chicago*, 11(4), 345.
- Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge: Harvard University Press.
- Julich, S., Hirst, D., & Thompson, B., (2003). A case study of ILS migration: Aleph500 at the University of Iowa. *Library Hi Tech*, 21(1), 44–55.
<https://doi.org/10.1108/07378830310467391>

- Land, P. (2013). Migrations: Not just for developers any more. *Bulletin of the American Society for Information Science and Technology*, 40(1), 42–44.
<https://doi.org/10.1002/bult.2013.1720400112>
- Lauesen, S. (2003). Task descriptions as functional requirements. *IEEE Software*, 20(2), 58–65. <https://doi.org/10.1109/MS.2003.1184169>
- Lauesen, S., & Harning, M. B. (2001). Virtual windows: Linking user tasks, data models, and interface design. *IEEE Software*, 18(4), 67–75.
<https://doi.org/10.1109/MS.2001.936220>
- Nfila, R.B., Dintwe, M.N., Rao, K.N. (2005). Experience of systems migration at the University of Botswana Library: a case study. *Program*, 39(3), 248–256.
<https://doi.org/10.1108/00330330510610582>
- Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. L. Stevens (Eds.), *Mental models*. Hillsdale, N.J: Erlbaum.
- Poole, M., & DeSanctis, G. (1990). Understanding the use of group decision support systems: the theory of adaptive structuration. In J. Fulk & C. Steinfield, *Organizations and Communication Technology* (pp. 173–193).
<https://doi.org/10.4135/9781483325385.n8>
- Punch, K. (2014). *Introduction to social research: quantitative & qualitative approaches* (3rd ed). Los Angeles, California: SAGE.
- Rasmussen, J. (1986). *Information processing and human-machine interaction*. Amsterdam: North-Holland.
- Roger, Y. (1992). Introduction. In Y. Rogers, A. Rutherford, & P.A. Bibby (Eds.), *Models in the mind: Theory, perspective and application*. London: Academic Press.
- Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: prospects and limits in the search for mental models. *Psychological Bulletin*, 100(3), 349-363.
- Singh, V. (2013) The nuts and bolts of migration to open source ILS: Experiences and recommendations from librarians. *Libri*, 01/2013, 63(2), 108 – 122

- Thomer, A., Cheng, Y.-Y., Schneider, J., Twidale, M., & Ludäscher, B. (2017). Logic-based schema alignment for natural history museum databases. *Knowledge Organization*, 44(7), 545–558. <https://doi.org/10.5771/0943-7444-2017-7-545>
- Yin, R. K. (2009). *Case study research: design and methods* (4th ed). Los Angeles, Calif: Sage Publications.
- Young, R. M. (1981). The machine inside the machine: Users' models of pocket calculators. *International Journal of Man-Machine Studies*, 15(1), 51–85. [https://doi.org/10.1016/S0020-7373\(81\)80023-5](https://doi.org/10.1016/S0020-7373(81)80023-5)

APPENDIX

Appendix A – Interview Guide

Thank you for taking the time to meet with me. Your experience of using [name of database] database for your work is valuable for database design and redesign. In the form I sent you, there is information about this approved research. Would you mind if I record the interview? The recording will be saved on the UTK Office365, a secure server and accessible only by me and my thesis advisor.

I will be asking a series of question about your usage of and experience with [name of database]. You may not have the answers to all of these questions, and that's a perfectly acceptable and valuable input. Please just answer to the best of your knowledge.

1. When you think of [name of database], what comes to mind?
2. How often do you use this database?
3. What are the important parts of [name of database] that are relevant to your job tasks?
4. Where do you think the data is stored?
 - a. Follow-up: if locally, how is it shared with others?
5. How do you believe the membership data is organized?
 - a. [if the interviewee is confused, offer methods, such as drawing or other visual/verbal depictions]
6. How are the member records stored?

7. How are the organization records stored?
8. Could you tell me a specific instance when you encountered difficulties working with [name of database]? I am interested in what you were trying to accomplish, what problems occurred, and how the problems were solved.
 - a. [record the issue]
 - b. [record the issue]
 - c.
9. Follow-up on 8. [be sure to have information on task, action, result, thoughts, or insights by the interviewee. Ask questions to fill gaps.]
10. Could you tell me a pleasant instance you have experienced using [name of database]?
11. Follow-up on 10. [be sure to have information on task, action, result, thoughts or insights by the interviewee. Ask questions to fill gaps.]
12. In retrospect, approximately how much time, e.g. a total number of hours, did you spend on learning the basics of the system? What were the challenges to learn [name of database]?
 - a. [record challenge]
 - b. [record challenge]
 - c.
13. Follow-up on 12. [get details on challenges related to ask, conception, interface, ...]

14. Assume I am a new staff member and you will give me an introduction about how [name of the database] works, how would you describe it?

15. How would you explain to a new person how the database [name of database] works?

16. Would you open the database [name of database] and show me what you like and why; what you think should be improved and how?

17. Do you have any advice to other directors of non-profit organizations in need of database migration?

18. Do you have any other important stories to tell?

Thank you! I will be happy to share my thesis with you when it is completed.

VITA

Scott Shumate is currently employed at Austin Peay State University as the IT Analyst for the Digital Services unit of the Felix G. Woodward Library. Scott received his undergraduate degree in computer science and information systems from Austin Peay State University in 2016. He pursued his degree in information sciences from the University of Tennessee – Knoxville at the encouragement of his colleagues after his experience working in the Woodward Library led him to a passion for applying his knowledge about technology to the information space. Scott hopes to continue working to improve the technological systems we use to ease the burden of accessing the wealth of knowledge made available in the information age.