Dynamic Menu Interface Designer

Design Specification: Dynamic Menu Interface Designer

This deliverable provides the design specification (DS) for a generic Dynamic Menu Interface Designer (DMID) system that may be suitable for a small or medium sized organization. The DMID enhances software engineering by reducing the amount of time normally taken for user interface planning and construction.

Contents

1.	System Overview .	•	•	•	•	•	•	•	•	•	•	02
1.1	Problem Definition .									•		02
1.2	Proposed Solution .	•	•	•	•	•	•	•	•	•	•	02
2.	Database Specification	•	•	•	•	•	•	•	•	•	•	04
2.1	Introduction		•	•						•		04
2.2	Database Details .	•	•	•						•		06
2.3	Recommended Readings	•	•	•	•	•	•	•	•	•	•	14
3.	Operations Specification	n	•	•	•	•	•	•	•	•	•	15
3.1	Generic Pseudo0codes	•	•	•						•		15
3.2	Administrative Specificat	ion M	Ianage	ement	Subsy	stem			•	•		17
3.3	End-user Access Control	Subsy	ystem									38
3.4	Recommended Reading		•	•	•	•	•	•	•	•	•	43
4.	User Interface Specifica	tion	•	•	•	•	•	•	•	•	•	44
4.1	User Interface Topology	•	•	•	•				•	•		44
4.2	Message Specification	•										46
4.3	Help Specification .											46
4.4	Recommended Readings	•	•	•	•	•	•	•	•	•	•	46
5.	Summary and Concludi	ng Re	emark	S	•	•	•	•	•	•	•	47

Copyright © 2010 - 2011 by Elvis C. Foster

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, or otherwise, without prior written permission of the author.

E. C. Foster

1. System Overview

1.1 Problem Definition

Software engineering has come to the stage where speed of development, level of correctness, interoperability, user friendliness, usefulness, and reusability in different projects are very important factors in determining the success of a software engineering venture. Software consumers have become quite impatient and reluctant to persist with products that do not meet their expectations. Ironically, amidst this state of the market, software engineers and information system developers are expected to deliver projects on or ahead of schedule, or face the wrath and consequence of a disgruntled consuming public.

Menu design and construction take a considerable portion of software development. Additionally, in many cases, the menu systems that characterize software systems are static. These two realities lead to a number of potential problems:

- Software development time is longer, since time has to be allocated for the design and development of the menu for each system.
- If there are changes in the accessibility requirements for end users of the system, these could force programming changes or configuration changes for the software system.

1.2 Proposed Solution

This project proposes the DMID as a software component that has the potential of reducing software development duration. The DMID takes as input a data set that includes the essential information on the operational and security requirements of the system being constructed, and generates a menu of user options based on each user's profile. This component removes the burden of menu design and construction from the software construction phase of the *software development life cycle* (SDLC), thus giving the software engineer more time to concentrate on other pressing and important aspects of software construction.

A10.1.3 System Architecture

The DMID consists of three main components:

- Central Database
- Administrative Specification Management (ASM)
- End-user Access Control (EAC)

Figure 1.1 provides the information topology chart (ITC) and figure 1.2 shows the object flow diagram (OFD).



Figure 1.1: Information Topology Chart for the DMID

Figure 1.2: Object Flow Diagram for the DMID



2. Database Specification

The database specification will proceed under the following captions:

- Introduction
- Database Details

2.1 Introduction

The database specification of the system will be covered in this section. The methodology employed for database specification is the *object/entity specification grid* (O/ESG) developed by the current author (see [Foster, 2010a] and [Foster, 2010b]). Following is a summary of the conventions used:

- Each information entity referenced is identified by a reference code and a descriptive name.
- For each entity the attributes (data elements) to be stored are identified.
- The entities as presented, will easily transition into a set of normalized relations in a normalized relational database.
- Data elements that will be implemented as foreign keys, in the normalized relational database, are identified by comment in curly braces, specifying what entity they reference.
- For each attribute, the physical characteristics will be given (as described in the next section); the attributes implementation name will be indicated in square brackets; it will be indicated whether the attribute is a foreign key.
- Indexes (including primary key or candidate keys) to be defined on the entity are indicated.
- For each entity a comment describing the data to be stored is provided. Additionally, the entities implementation name is indicated in square brackets.
- Each operation defined on an entity will be given an implementation name, indicated in square brackets.

Naming of database objects will be very important for the following reasons:

- The database will host several objects. Without a proper naming convention, it will be extremely difficult to keep track of them.
- The naming convention will enable us to easily categorize database objects on sight.

Figure 2.1 provides the object naming convention for the project, and figure 2.2 shows an entity-relationship diagram (ERD).

2.1 Introduction (continued)

Figure 2.1: Object Naming Conventions

Object Name: SS XXXXXXX MMn where SS represents the system or subsystem abbreviation; MMn represents the object mode or purpose (1-3 bytes); XXXXXXXX represents the descriptive name of the object (6-8 bytes). Valid subsystem abbreviations include: End-user Access Control (EAC) EA: AS: Administrative Specification Management (ASM) DM: Generic resources (including the database tables) for the DMID Valid mode abbreviations include: BR: A base relation (if relational DB model) OT: An object type (if OO DB model) LVn: A logical view (e.g. LV1, LV2, etc.) NXn: An index to a base table or object type (e.g. NX1, NX2, etc.) PK: Primary Key FKn: Foreign Key (e.g. FK1, FK2, etc.) ICn: Integrity Constraint (e.g.IC1, IC2, etc.) AO: An ADD operation MO: A MODIFY operation ZO: A DELETE (Zap) operation 10: An INQUIRE operation FO: A FORECAST operation. RO: A REPORT operation XO: A utility operation DS: A database synonym or alias of a known database table DC: A database constraint DT: A database Trigger DP: A database procedure or function DK: A database package MF: A Message file — a special purpose database table (file) to store the text (and other essential details) for diagnostic error and status messages The descriptor used for a database base relation or object type is consistently used for other objects that directly relate to that object. For instance the objects used for the management of information directly related to system definitions may be: DM_SystemD_BR: The base relation to store data on system definitions DM_SystemD_NX1: An index on the system definitions Operation to ADD system definitions AS_SystemD_AO: Course to MODIFY system definitions AS SystemD MO: AS SystemD ZO: Course to DELETE system definitions AS_SystemD IO: Course to INQUIRE on system definitions AS SystemD RO: Course to REPORT on system definitions AS_SystemD_XO: A utility operation related to system definitions AS_SystemD_LV1: A logical view of the system definitions Attribute implementation names are mere abbreviations of their more descriptive names.

E. C. Foster

2.1 Introduction (continued)

Figure 2.2: Entity-relationship Diagram for the DMID



2.2 Database Details

The underlying database is called **DMID2010_DB**, and it will be implemented in MySQL. Figure 2.3 provides the O/ESG for each information entity comprising the system. Note that the O/ESG also provides useful information that will be used during subsequent user interface development for the system.

Figure 2.3: O/ESG for the DMID

E0 ′	1. System Definitions [DM_SystemD_BR] consists of the following:
Att	ributes:
1.	System Code [SysCode] [A4]
2.	System Name [SysName] [A60]
3.	System Abbreviation [SysAbbr] [A6]
4.	Home Path [SysHome] [A100]
Со	mments:
1.	This entity facilitates the internal identification of all software systems that use the DMID
2.	System Codes will consist of a 2 byte alphabetic code combined with a 2 digit sequential number.
Ind	exes:
1.	Primary Key: [1] (Constraint Name is DM_SystemD_PK)
2.	DM_SystemD_NX2 on [2]
Val	id Operations:
1.	Add System Definitions [AS_SystemD_AO]
2.	Modify System Definitions [AS_SystemD_MO]
3.	Delete System Definitions [AS_SystemD_ZO]
4.	Inquire System Definitions [AS_SystemD_IO]
5.	Report System Definitions [AS_SystemD_RO]

2.2 Database Details (continued)

Figure 2.3: O/ESG for the DMID (continued)

E02. Participating Organizations [DM_OrgD_BR] consists of the following:
Attributes:
1. Organization Code [OrgCode] [A4]
2. Organization Name [OrgName] [A30]
3. Organization Abbreviation [OrgAbbr] [A6]
Comments:
1. This entity facilitates the internal identification of all participating organizations that use the DMID
2. Organization Codes will consist of a 2 byte alphabetic code combined with a 2 digit sequential number.
Indexes:
1. Primary Key: [1] (Constraint Name is DM_OrgD_PK)
2. DM_OrgD_NX2 on [2]
Valid Operations:
1. Add Organizations [AS_OrgD_AO]
2. Modify Organizations [AS_OrgD_MO]
3. Delete Organizations [AS_OrgD_ZO]
4. Inquire Organizations [AS_OrgD_IO]
5. Report Organizations [AS_OrgD_RO]

E03. System Users [DM_User_BR] consists of the following:

Attributes:

- 1. User Identification Code [UsrCode] [A8]
- 2. User Login Name [UsrName] [A15]
- 3. User First Name [UsrFName] [A15]
- 4. User Last Name [UsrLName] [A15]
- 5. User' Organization [UsrOrgCode] [A4] {Refers to E02}
- 6. User Classification [UsrClass] [A3]
- 7. User Password [UsrPssWrd] [A12]
- 8. User Password Change Ceiling in days [UsrPssCeil] [N2]
- 9. Date of Last Password Change [UsrPssChgD] [N8]

Comments:

- 1. This entity facilitates definition of user accounts for the system.
- 2. User Codes will consist of a year concatenated to a sequence number (YYYYnnnn).
- 3. User Classification will be "End" (for end-user) or "Adm" (for administrator).

Indexes:

- 1. Primary Key: [1] (Constraint Name is **DM_User_PK**)
- 2. DM_User_NX2 on [2]

Valid Operations:

- 1. Add User Accounts [AS_User_AO]
- 2. Modify User Accounts [AS_User_MO]
- 3. Delete User Accounts [AS User ZO]
- 4. Inquire User Accounts [AS User IO]
- 5. Report User Accounts [AS_User_RO]

2.2 Database Details (continued)

Figure 2.3: O/ESG for the DMID (continued)

E04. System Operations [DM_Oper_BR] consists of the following:
Attributes:
1. Operation Code [OpCode] [A8]
2. Operation Implementation Name [OpIName] [A15]
3. Operation Descriptive Name [OpDName] [A30]
4. Operation Description [OpDscr] [M]
5. Operation Home Path [OprHome] [A100]
Comments:
1. This entity facilitates the definition of all operations that will be accessed via systems that use the DMID
2. Each Operation Code will consist of an O, the host system code, and a 3 digit sequence number.
Indexes:
1. Primary Key: [1] (Constraint Name is DM_Oper_PK)
2. DM_Oper_NX2 on [2]
3. DM_Oper_NX3 on [3]
Valid Operations:
1. Add Operations [AS_Oper_AO]
2. Modify Operations [AS_Oper_MO]
3. Delete Operations [AS_Oper_ZO]
4. Inquire Operations [AS_Oper_IO]
5. Report Operations [AS_Oper_RO]

E05. System Menu Definitions [DM_MenuD_BR] consists of the following:
Attributes:
1. Menu Code [MnuCode] [A8]
2. Menu Implementation Name [MnulName] [A15]
3. Menu Descriptive Name [MnuDName] [A30]
4. Menu Description [MnuDscr] [M]
5. Menu's System Code [MnuSysCode] [A4] {Refers to E01}
6. Menu's Home Path [MnuHome] [A100]
Comments:
1. This entity facilitates definition of system menus.
2. Each Menu Code will consist of an M, the host system code, and a 3 digit sequence number.
Indexes:
1. Primary Key: [1] (Constraint Name is DM_MenuD_PK)
2. DM_MenuD_NX2 on [2]
3. DM_MenuD_NX3 on [3]
Valid Operations:
1. Add Menu Definitions [AS_MenuD_AO]
2. Modify Menu Definitions [AS_MenuD_MO]
3. Delete Menu Definitions [AS_MenuD_ZO]
4. Inquire Menu Definitions [AS_MenuD_IO]
5. Report Menu Definitions [AS_MenuD_RO]

2.2 Database Details (continued)

Figure 2.3: O/ESG for the DMID (continued)

E06. Menu Constituents [DM_MenuC_BR] consists of the following:
Attributes:
1. Menu Code [MC_MnuCode] [A8] {Refers to E05}
2. Menu Sequence Number [MC_MnuSeqN] [N3]
3. Constituent Operation Code [MC_OpCode] [A8] {Refers to E04}
Comments:
This entity facilitates the specification of menu-operation combinations. The operations for each menu are specified.
Indexes:
1. Primary Key: [1, 2] (Constraint Name is DM_MenuC_PK)
2. DM_MenuC_NX2 on [1, 3]
Valid Operations:
1. Add Menu Constituents [AS_MenuC_AO]
2. Delete Menu Constituents [AS_MenuC_ZO]
3. Inquire Menu Constituents [AS_MenuC_IO]
4. Report Menu Constituents [AS_MenuC_RO]

E07. User-to-Operations Authorization [DM_UsrOprA_BR] consists of the following:

Attributes:

- 1. User Identification Code [UO_UsrCode] [A8] {Refers to E03}
- 2. Authorized Operation Code [UO_OpCode] [A8] {Refers to E04}

Comments:

This entity facilitates specification of the user-operation access matrix for each user.

Indexes:

Primary Key: [1, 2] (Constraint Name is **DM_UsrOprA_PK**)

Valid Operations:

- 1. Add User-Operation Authorizations [AS_UsrOprA_AO]
- 2. Delete User-Operation Authorizations [AS_UsrOprA_ZO]
- 3. Inquire User-Operation Authorizations [AS_UsrOprA_IO]
- 4. **Report** User-Operation Authorizations [AS_UsrOprA_RO]

2.2 Database Details (continued)

Figure 2.3: O/ESG for the DMID (continued)

E08. User-to-Menu Authorization [DM_UsrMnuA_BR] consists of the following:
Attributes:
1. User Identification Code [UM_UsrCode] [A8] {Refers to E03}
2. Authorized Menu Code [UM_MnuCode] [A8] {Refers to E05}
3. User Menu Sequence Number [UM_MnuSeqN] [N3]
Comments:
This entity facilitates specification of the user-menu access matrix for each user.
Indexes:
1. Primary Key: [1, 3] (Constraint Name is DM_UsrMnuA_PK)
2. DM_UsrMnuA_NX2 on [1, 2]
Valid Operations:
1. Add User-Menu Authorizations [AS_UsrMnuA_AO]
2. Delete User-Menu Authorizations [AS_UsrMnuA_ZO]
3. Inquire User-Menu Authorizations [AS_UsrMnuA_IO]
4. Report User-Menu Authorizations [AS_UsrMnuA_RO]

EUS. USEI-IU-SYSTEIII AULIUIIZAUUII IDIVI USISYSA DRI CUIISISIS ULITE IUIIOWIIIQ	E09. l	User-to-System	Authorization	[DM Usr	SysA BR]	consists of the followin
----------------------------------------------------------------------------------	--------	----------------	---------------	---------	----------	--------------------------

Attributes:

- 1. User Identification Code [US_UsrCode] [A8] {Refers to E03}
- 2. Authorized System Code [US_SysCode] [A4] {Refers to E01}
- 3. User System Sequence Number [US_SysSeqN] [N3]

Comments:

This entity facilitates specification of the user-system access matrix for each user.

Indexes:

- 1. Primary Key: [1, 3] (Constraint Name is DM_UsrSysA_PK)
- 2. **DM_UsrSysA_NX2** on [1, 2]

Valid Operations:

- 1. Add User-Menu Authorizations [AS_UsrSysA_AO]
- 2. Delete User-Menu Authorizations [AS_UsrSysA_ZO]
- 3. Inquire User-Menu Authorizations [AS_UsrSysA_IO]
- 4. **Report** User-Menu Authorizations [AS_UsrSysA_RO]

E. C. Foster

2.2 Database Details (continued)

Figure 2.3: O/ESG for the DMID (continued)

E10. Organization-to-System Mapping [DM_OrgSysM_BR] consists of the following:
Attributes:
1. Organization Code [OS_OrgCode] [A4] { Refers to E02 }
2 System Code IOS SysCode] [A4] {Refers to F01}

3. System Sequence Number [OS_SysSeqN] [N3]

Comments:

This entity facilitates specification of the organization-system mappings for the DMID's working environment.

Indexes:

- 1. Primary Key: [1, 3] (Constraint Name is DM_OrgSysM_PK)
- 2. **DM_OrgSysM_NX2** on [1, 2]

Valid Operations:

- 1. Add Organization-System Mappings [AS_OrgSysM_AO]
- 2. Delete Organization-System Mappings [AS_OrgSysM_ZO]
- 3. Inquire Organization-System Mappings [AS_OrgSysM_IO]
- 4. Report Organization-System Mappings [AS_OrgSysM_RO]

E11. System Messages [DM_Message_MF] consists of the following:

Attributes:

- 1. Message Identification Code [MsgCode] [A5]
- 2. Message Descriptive Text [MsgText] [M]
- 3. Message Category [MsgClass] [A3]

Comments:

- 1. This entity facilitates specification of error, diagnostic, and status messages sent to the user.
- 2. Message Code will take the entity reference number, concatenated with a sequence number (format EEEnn) for example E0101.

Indexes:

Primary Key: [1] (Constraint Name is **DM_Message_PK**)

Valid Operations:

- 1. Add System Messages [AS_Message_AO]
- 2. Modify System Messages [AS_Message_MO]
- 3. Delete System Messages [AS_Message _ZO]
- 4. Inquire System Messages [AS_Message M_IO]
- 5. Report System Messages [AS_Message _RO]

E. C. Foster

2.2 Database Details (continued)

Each information entity of the O/ESG will be implemented as a relational table in a relational the relational database. In addition to these tables, a number of logical views will be required. These are summarized in figure 2.4.

Figure 2.4: Important Logical Views for the DMID

User's System Overview [DM_UsrSysA_LV1]: This is the logical join of User-System Authorizations (E09) with System Definitions (E01), and System Users (E03). Attributes will be read-only:

- User Identification Code [US_UsrCode]
- User Login Name [UsrName]
- User First Name [UsrFName]
- User Last Name [UsrLName]
- System Code [US-SysCode]
- User System Sequence Number [US_SysSeqN]
- System Name [SysName]
- System Abbreviation [SysAbbr]

User Menus Summary [DM_UsrMnuA_LV1]: This is the logical join of User-Menu Authorizations (E08) with System Menu Definitions (E05) and System Users (E03), and System Menu Definitions (E05) with System Definitions (E01). Attributes will be read-only:

- User Identification Code (UM_UsrCode)
- User Login Name [UsrName]
- User First Name [UsrFName]
- User Last Name [UsrLName]
- Menu Code (UM_MnuCode)
- Menu Implementation Name [MnulName]
- Menu Descriptive Name [MnuDName]
- Menu's System Code [MnuSysCode]
- System Name [SysName]
- System Abbreviation [SysAbbr]
- User Menu Sequence Number (UM_MnuSeqN)

User Operations Summary [DM_UsrOprA_LV1]: This is the logical join of User-Operation Authorization (E07) with Menu Constituents (E06), System Operations (E04), System Users (E03), and Menu Definitions (E05). Attributes will be read-only:

- User Identification Code [UO_UsrCode]
- User Login Name [UsrName]
- User First Name [UsrFName]
- User Last Name [UsrLName]
- Authorized Operation Code [UO_OpCode]
- Operation Implementation Name [OpIName]
- Operation Descriptive Name [OpDName]
- Menu Code [MC_MnuCode]
- Menu Sequence Number [MC_MnuSeqN]
- Constituent Operation Code [MC_OpCode]
- Menu Implementation Name [MnulName]
- Menu Descriptive Name [MnuDName]

2.2 Database Details (continued)

Figure 2.4: Important Logical Views for the DMID (continued)

Organization-System Mapping [DM_OrgSysM_LV1]: This is the logical join of Organization-System Mapping (E10) with Participating Organizations (E02), and System Definitions (E01). Attributes will be read-only:

- Organization Code [OS_OrgCode]
- Organization Name [OrgName]
- Organization Abbreviation [OrgAbbr]
- System Sequence Number [OS_SysSeqN]
- System Code [OS_SysCode]
- System Name [SysName]
- System Abbreviation [SysAbbr]

System Users [DM_User_LV1]: This is the logical join of System Users (E03) with Participating Organizations (E02). Attributes will be read-only:

- User Identification Code [UsrCode]
- User Login Name [UsrName]
- User First Name [UsrFName]
- User Last Name [UsrLName]
- User' Organization [UsrOrgCode]
- Organization Name [OrgName]
- Organization Abbreviation [OrgAbbr]
- User Classification [UsrClass]
- User Password [UsrPssWrd]
- User Password Change Ceiling in days [UsrPssCeil]
- Date of Last Password Change [UsrPssChgD]

System Menu Definitions [DM_MenuD_LV1]: This is the logical join of System Menu Definitions (E05) and System

Definitions (E01). Attributes will be read-only:

- Menu Code [MnuCode]
- Menu Implementation Name [MnulName]
- Menu Descriptive Name [MnuDName]
- Menu Description [MnuDscr]
- Menu's System Code [MnuSysCode]
- System Name [SysName]
- System Abbreviation [SysAbbr]

E. C. Foster

2.2 Database Details (continued)

Figure 2.4: Important Logical Views for the DMID (continued)

System Menu Constituents [DM_MenuC_LV1]: This is the logical join of System Menu Constituents (E06), Menu Definitions (E05), System Definitions (E01), and System Operations (E04). Attributes will be read-only:

- Menu Code [MC_MnuCode]
- Menu Implementation Name [MnulName]
- Menu Descriptive Name [MnuDName
- Menu's System Code [MnuSysCode]
- System Name [SysName]
- System Abbreviation [SysAbbr]
- Menu Sequence Number [MC_MnuSeqN]
- Constituent Operation Code [MC_OpCode]
- Operation Implementation Name [OpIName]
- Operation Descriptive Name [OpDName]

Key:

- Attributes in black are taken from the reverencing relation
- Attributes in blue are taken from the referenced relation(s)

2.3 Recommended Readings

[Date 2004] Date, Christopher J. 2004. *Introduction to Database Systems* 8^{h} ed. Menlo Park, California: Addison-Wesley. See chapters 3, 6, 12 - 14.

[Elmasri 2007] Elmasri, Ramez and Shamkanf B. Navathe. 2007. *Fundamentals of Database Systems* 5th ed. Reading, Massachusetts: Addison-Wesley, 2007. See chapters 3, 7, 10, 11.

[Foster 2010a] Foster, Elvis C. *Software Engineering: A Methodical Approach*. 2010. Bloomington, IN: Xlibris Publishing. See lectures 9 and 10, and appendix 10.

[Foster, 2010b] Foster, Elvis C. with Shripad Godbole. 2010. *Database Systems: A Pragmatic Approach*. Bloomington, IN: Xlibris Publishing.

[Hoffer 2007] Hoffer, Jeffrey A., Mary B. Prescott and Fred R. McFadden. 2007. *Modern Database Management* 8th ed. Upper Saddle River, New Jersey: Prentice Hall. See chapters 3 – 5.

E. C. Foster

3. Operations Specification

In this section, operation specifications for the DMID system operations are provided. The algorithms needed for some of the operations (for different entities) are similar. Therefore, in the interest of brevity, instead of repeating the same pseudo code for these similar operations, the following generic operation outlines will be referenced. The chapter proceeds as follows:

- Generic Pseudo-codes
- Administrative Specification Management (ASM) Subsystem
- End-user Access Controls (EAC) Subsystem

3.1 Generic Pseudo-codes

Generic pseudo-codes are provided for the ADD operation, the MODIFY operation and the DELETE operation in figures 3.1, 3.2, and 3.3 respectively.

Figure 3.1: Generic ADD Pseudo-code

START							
WHILE (User wishes to continue)							
Accept Key Field(s);							
Check Record Absence or Existence in the primary file;							
IF (Record Absent)							
Accept Non-key Fields;							
Validate Non-key Fields based on Validation Rules;							
WHILE(Any Error Exists),							
Re-display Non-key Fields for possible Update;							
Display appropriate error message(s);							
Validate Non-key Fields based on Validation Rules;							
END-WHILE;							
Re-display full Record for confirmation;							
IF (Confirmation Obtained)							
Write New Record to the primary file:							
ENDIF:							
ELSE Inform the User that nothing was saved: END-ELSE:							
ENDIF:							
ELSE Display Message ('Record already exists'): END-ELSE:							
Check if User wishes to guit and set an exit flag if necessary							
FND-WHILE							
Generate Edit-List							
STOP							

3.1 Generic Pseudo-codes (continued)

Figure 3.2: Generic MODIFY Pseudo-code

START
WHILE (User wishes to continue)
Accept Key Field(s);
Check Record Absence or Existence in the primary file;
IF (Record Present)
Retrieve Record;
Display Non-key Fields for possible Update;
Validate Non-key Fields based on Validation Rules;
WHILE(Any Error Exists),
Re-display Non-key Fields for possible Update;
Display appropriate error message(s);
Validate Non-key Fields based on Validation Rules;
END-WHILE;
Re-display full Record for confirmation;
IF (Confirmation Obtained)
Update Record in the primary file;
ENDIF;
ELSE Inform the User that nothing was saved; END-ELSE;
ENDIF;
ELSE Display Message ('Record does not exists'); END-ELSE;
Check if User wishes to quit and set an exit flag if necessary;
END-WHILE;
Generate Edit-List;
STOP

Figure 3.3: Generic DELETE Pseudo-code

START
WHILE (User wishes to continue)
Accept Key Field(s);
Check Record Absence or Existence in the primary file;
IF (Record Present)
Retrieve Record;
Display full Record for confirmation;
IF (Deletion Confirmation Obtained)
Delete Record from the primary file;
ENDIF;
ELSE Inform the User that nothing was deleted; END-ELSE;
ENDIF;
ELSE Display Message ('Record does not exists'); END-ELSE;
Check if User wishes to quit and set an exit flag if necessary;
END-WHILE;
Generate Edit-List;
STOP

3.2 Administrative Specification Management (ASM) Subsystem

The operation specifications for operations in the Administrative Specifications Management (ASM) subsystem will be provided in this section. These operations relate to management of the entities specified in the previous chapter. The methodology employed in specifying the operational requirements is the *extended operation specification* (EOS) as outlined in [Foster, 2010b].

Figure 3.4: Operations Specification for Managing System Definitions Data

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_SystemD_AO / AS_SystemD_MO / AS_SystemD_ZO Facilitates addition/modification/deletion of system definitions. Mandatory 6 of 10 E. Foster 2010-5-23			
Inputs: Information on new or already participating software systems DM_SystemD_BR — System Definitions (E01)				
Outputs: DM_SystemD_BR — System Definitions (E01)				
Validation Rules for Adding:1. System Code must not previously exist2. Blank System Name not allowed				
Validation Rules for Modifying:1. System Code must previously exist2. Blank System Name not allowed				
Special Notes: None				
Operation Outline: See Generic ADD / MODIFY / DELETE pseudo-code.				

Figure 3.5: Operations Specification for Inquiry/Report on System Definitions

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_SystemD_IO / AS_SystemD_RO Facilitates inquiry/report on System Definitions. Important 8 of 10 E. Foster 2010-5-23		
Inputs: DM_SystemD_BR — Syste	em Definitions (E01)		
Outputs: Monitor / Printer			
Validation Rules: None			
Special Notes: It will be possible to query System Definitions by System Code or System Name.			
Operation Outline: START: While User Wishes to Continue Prompt user for System Code or System Name; Prompt user for preference (System Code or System Name); If (By Code) Starting at that point in DM_SystemD_BR Load a Virtual Data Collection Object with all records until End-of-File, ordering by System Code; Display the Virtual Data Collection Object; End-If; If (By Name) Starting at that point in DM_SystemD_BR, Load a Virtual Data Collection Object with all records until End-of-File, ordering by System Name; Display the Virtual Data Collection Object; End-If;			
End-While; STOP			

Figure 3.6: Operations Specification for Managing Participating Organizations Data

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_OrgD_AO / AS_OrgD_MO / AS_OrgD_ZO Facilitates addition/modification/deletion of participating organizations data. Mandatory 6 of 10 E. Foster 2010-5-23		
Inputs: Information on new or already participating organizations DM_OrgD_BR — Participating Organizations (E02)			
Outputs: DM_OrgD_BR — Participating Organizations (E02)			
Validation Rules for Adding:1. Organization Code must not previously exist2. Blank Organization Name not allowed			
Validation Rules for Modifying:1. Organization Code must previously exist2. Blank Organization Name not allowed			
Special Notes: None			
Operation Outline: See Generic ADD / MODIFY / DELETE pseudo-code.			

Figure 3.7: Operations Specification for Inquiry/Report on Participating Organizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_OrgD_IO / AS_OrgD_RO Facilitates inquiry/report on Participating Organizations. Important 8 of 10 E. Foster 2010-5-23		
Inputs: DM_OrgD_BR — Participa	ting Organizations (E02)		
Outputs: Monitor / Printer			
Validation Rules: None			
Special Notes: It will be p	ossible to query Participating Organizations by Organization Code or Organization Name.		
Operation Outline: START: While User Wishes to Continue Prompt user for Organization Code or Organization Name; Prompt user for preference (Organization Code or Organization Name); If (By Code) Starting at that point in DM_OrgD_BR Load a Virtual Data Collection Object with all records until End-of-File, ordering by Organization Code; Display the Virtual Data Collection Object; End-If; If (By Name) Starting at that point in DM_OrgD_BR, Load a Virtual Data Collection Object with all records until End-of-File, ordering by Organization Name; Display the Virtual Data Collection Object; End-If; End-If; End-If; End-If; End-While:			
STOP			

Figure 3.8: Operations Specification for Managing User Accounts

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_User_AO / AS_User_MO / AS_User_ZO Facilitates addition/modification/deletion of user accounts. Mandatory 6 of 10 E. Foster 2010-5-23		
Inputs: Information on new or already existing user accounts DM_User_BR — System Users (E03)			
Outputs: DM_User_BR — System Users (E03)			
Validation Rules for Adding:1. User Identification Code must not previously exist2. Blank Login Name not allowed			
Validation Rules for Modifying:1. User Identification must previously exist2. Blank Login Name not allowed			
Special Notes: None			
Operation Outline: See Generic ADD / MODIFY / DELETE pseudo-code.			

Figure 3.9: Operations Specification for Inquiry/Report on User Accounts

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_User_IO / AS_User_RO Facilitates inquiry/report on User Accounts. Important 10 of 10 E. Foster 2010-5-23		
Inputs: DM_User_BR — User Acco DM_OrgD_BR — Participat	ounts (E03) ting Organizations (E02)		
Outputs: Monitor / Printer			
Validation Rules: None			
 Special Notes: 1. It will be possible to query Items via any of the following access paths: 1.1 By Identification Code 1.2 By Login Name 1.3 By Last Name and First Name 1.4 By Classification and Login Name 2. Each option will invoke one of four sub-operations (AS_User_I1, AS_User_I2, AS_User_I3, and AS_User_I4). 3. Each sub-operation utilizes the logical view DM_User_LV1, which joins System Users (E03) with Participating Organizations (E02). 			
Operation Outline: START: /* Inquire */ While User Wishes to Continue Present the User with the options mentioned above; Depending on the User's choice, invoke one of the sub-operations; End-While; STOP			
Outline for AS_User_I1: START While User Wishes to Continue Prompt user for Identification Code; Starting at that point in DM_User_LV1, Load a Virtual Data Collection Object with all records until End-of-File, ordering by User Identification Code; Display the Virtual Data Collection Object; End-While; STOP			
{The other sub-operations will be similar}			

Figure 3.10: Operations Specification for Managing System Operations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_Oper_AO / AS_Oper_MO / AS_Oper_ZO Facilitates addition/modification/deletion of system operations. Mandatory 6 of 10 E. Foster 2010-5-23		
Inputs: Information on new or already system operations DM_Oper_BR — System Operations (E04)			
Outputs: DM_Oper_BR — System Operations (E04)			
Validation Rules for Adding:3. Operation Code must not previously exist4. Blank Operation Name not allowed			
 Validation Rules for Modifying: 3. Operation Code must previously exist 4. Blank Operation Name not allowed 			
Special Notes: None			
Operation Outline: See Generic ADD / MODIFY / DELETE pseudo-code.			

Figure 3.11: Operations Specification for Inquiry/Report on System Operations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_Oper_IO / AS_Oper_RO Facilitates inquiry/report on System Operations. Important 8 of 10 E. Foster 2010-5-23		
Inputs: DM_Oper_BR — System C	Operations (E04)		
Outputs: Monitor / Printer			
Validation Rules: None			
Special Notes: It will be po	ossible to query System Operations by Operation Code or Operation Name.		
Operation Outline: START: While User Wishes to Continue Prompt user for Operation Code or Operation Name; Prompt user for preference (Operation Code or Operation Name); If (By Code) Starting at that point in DM_Oper_BR Load a Virtual Data Collection Object with all records until End-of-File, ordering by Operation Code; Display the Virtual Data Collection Object; End-If; If (By Description) Starting at that point in DM_Oper_BR, Load a Virtual Data Collection Object with all records until End-of-File, ordering by Operation Name; Display the Virtual Data Collection Object; End-If; End-If; End-If; End-If; End-If; End-If; End-While; STOP			

Figure 3.12: Operations Specification for Managing System Menu Definitions

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_MenuD_AO / AS_MenuD_MO / AS_MenuD_ZO Facilitates addition/modification/deletion of system menu definitions Mandatory 6 of 10 E. Foster 2010-5-23		
Inputs: Information on new or already menu definitions DM_MenuD_BR — Menu Definitions (E05)			
Outputs: DM_MenuD_BR — Menu Definitions (E05)			
Validation Rules for Adding:1. Menu Code must not previously exist2. Blank Menu Name not allowed			
 Validation Rules for Modifying: 1. Menu Code must previously exist 2. Blank Menu Name not allowed 			
Special Notes: None			
Operation Outline: See Generic ADD / MODIFY / DELETE pseudo-code.			

Figure 3.13: Operations Specification for Inquiry/Report on Menu Definitions

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_MenuD_IO / AS_MenuD_RO Facilitates inquiry/report on Menu Definitions Important 8 of 10 E. Foster 2010-5-23			
Inputs: DM_MenuD_BR — Menu D DM_SystemD_BR — Syste	Inputs: DM_MenuD_BR — Menu Definitions (E05) DM_SystemD_BR — System Definitions (E01)			
Outputs: Monitor / Printer				
Validation Rules: None				
 Special Notes: 1. It will be possible to query Menu Definitions by Menu Code or Menu Name. 2. The operation makes use of the logical view DM_MenuD_LV1, which joins Menu Definitions (E05) with System Definitions (E01). 				
Operation Outline: START: While User Wishes to Conti Prompt user for Men Prompt user for prefi If (By Code) Starting at tha Until E Display the V End-If; If (By Name) Starting at tha Until E Display the V End-If; End-Uf; End-Uf;	nue u Code or Menu Name; erence (Menu Code or Menu Name); at point in DM_MenuD_LV1 Load a Virtual Data Collection Object with all records ind-of-File, ordering by Menu Code; irtual Data Collection Object; at point in DM_MenuD_LV1 , Load a Virtual Data Collection Object with all records ind-of-File, ordering by Menu Name; irtual Data Collection Object;			

Figure 3.14: Operations Specification for Managing System Menu Constituents

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_MenuC_AO / AS_MenuC_ZO Facilitates addition/deletion of Menu Constituents Mandatory 7 of 10 E. Foster 2010-5-23		
Inputs: Information on new or already menu definitions DM_MenuC_BR — Menu Constituents (E06) DM_MenuD_BR — Menu Definitions (E05) DM_Oper_BR — System Operations (E04)			
Outputs: DM_MenuC_BR — Menu Constituents (E06)			
 Validation Rules for Adding: Menu Code must previously exist in DM_MenuD_BR Operation Code must previously exist in DM_Oper_BR 			
Special Notes: None			
Operation Outline: See Generic ADD / DELETE pseudo-code.			

Figure 3.15: Operations Specification for Inquiry/Report on Menu Constituents

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Operation Spec Author:	Dynamic Menu Inter Administrative Speci AS_MenuC_IO / AS Facilitates inquiry/rej Important E. Foster	face Designer (DMID) fication Management 5_MenuC_RO port on Menu Constitu Complexity Rank: Preparation Date:	(ASM) ients 10 of 10 2010-5-23	
Inputs: DM_MenuC_BR — Menu C DM_MenuD_BR — Menu D DM_Oper_BR — System C	Inputs: DM_MenuC_BR — Menu Constituents (E06) DM_MenuD_BR — Menu Definitions (E05) DM_Oper_BR — System Operations (E04)			
Outputs: Monitor / Printer				
Validation Rules: None				
 Special Notes: 1. It will be possible to query Menu Constituents by any of the following options: 1.1 By Menu Code and Menu Sequence 1.2 By Menu Name and Menu Sequence 1.3 By Menu Code and Operation Code 1.4 By Menu Name and Operation Name 2. Each option will invoke one of four sub-operations (AS_MenuC_I1, AS_MenuC_I2, AS_MenuC_I3, AS_MenuC_I4). 3. The operation makes use of the logical view DM_MenuC_LV1, which joins Menu Constituents (E06) with Menu Definitions (E05) and System Operations (E04). 				
Operation Outline: START: /* Inquire */ While User Wishes to Continue Present the User with the options mentioned above; Depending on the User's choice, invoke one of the sub-operations; End-While; STOP				
Outline for AS_MenuC_I1: START While User Wishes to Conti Prompt user for Iden Starting at that point ordering by M Display the Virtual D End-While; STOP	nue Itification Code; in DM_MenuC_LV1 , Ienu Code and Menu Iata Collection Object;	Load a Virtual Data C Sequence;	ollection Object with all records until End-of-File,	
{The other sub-operations w	vill be similar}			

Figure 3.16: Operations Specification for Managing User-Operation Authorizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_UsrOprA_AO / AS_UsrOprA_ZO Facilitates addition/deletion of User-Operation Authorizations Mandatory 7 of 10 E. Foster 2010-5-23	
Inputs: Information on new or already user-operation authorizations DM_UsrOprA_BR — User-Operation Authorizations (E07) DM_User_BR — System Users (E03) DM_Oper_BR — System Operations (E04)		
Outputs: DM_UsrOprA_BR — User-	Operation Authorizations (E07)	
 Validation Rules for Adding: 1. User Code must previously exist in DM_User_BR 2. Operation Code must previously exist in DM_Oper_BR 		
Special Notes: None		
Operation Outline: See Generic ADD / DELETE pseudo-code.		

Figure 3.17: Operations Specification Inquiry/Report on User-Operation Authorizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Operation Spec Author:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_UsrOprA_IO / AS_UsrOprA_RO Facilitates inquiry/report on User-Operation Authorizations Important Complexity Rank: 10 of 10 E. Foster Preparation Date: 2010-5-23	
Inputs: DM_UsrOprA_BR — User-Operation Authorizations (E07) DM_User_BR — System Users (E03) DM_Oper_BR — System Operations (E04)		
Outputs: Monitor / Printer		
Validation Rules: None		
 Special Notes: 1. It will be possible to query User-Operation Authorizations by any of the following options: 1.1 By User Code and Operation Code 1.2 By User Login and Operation Name 1.3 By User Login, Menu Code, and Operation Name, 1.4 By User Login, Menu Name, and Operation Name 1.5 By User Login, Menu Name, and Menu Sequence Number 2. Each option will invoke one of four sub-operations (AS_UsrOprA_I1, AS_UsrOprA_I2, AS_UsrOprA_I5). 3. The operation makes use of the logical view DM_UsrOprA_LV1, which joins User-Operation Authorization (E07) with Menu Constituents (E06), System Operations (E04), System Users (E03), and Menu Definitions (E05). 		
Operation Outline: START: /* Inquire */ While User Wishes to Continue Present the User with the options mentioned above; Depending on the User's choice, invoke one of the sub-operations; End-While; STOP		
Outline for AS_UsrOprA_I1: START While User Wishes to Continue Prompt user for Identification Code; Starting at that point in DM_UsrOprA_LV1, Load a Virtual Data Collection Object with all records until End-of-File, ordering by User Code and Operation Code; Display the Virtual Data Collection Object; End-While; STOP		
{The other sub-operations w	/ill be similar}	

Figure 3.18: Operations Specification for Managing User-Menu Authorizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_UsrMnuA_AO / AS_UsrMnuA_ZO Facilitates addition/deletion of User-Menu Authorizations Mandatory 7 of 10 E. Foster 2010-5-23	
Inputs: Information on new or already user-menu authorizations DM_UsrMnuA_BR — User-Menu Authorizations (E08) DM_User_BR — System Users (E03) DM_MenuD_BR — System Menu Definitions (E05)		
Outputs: DM_UsrMnuA_BR — User-Menu Authorizations (E08)		
 Validation Rules for Adding: 1. User Code must previously exist in DM_User_BR 2. Menu Code must previously exist in DM_MenuD_BR 		
Special Notes: None		
Operation Outline: See Generic ADD / DELETE pseudo-code.		

Figure 3.19: Operations Specification for Inquiry/Report on User-Menu Authorizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Operation Spec Author:	Dynamic Menu Interface Designer (DMID)Administrative Specification Management (ASM)AS_UsrMnuA_IO / AS_UsrMnuA_ROFacilitates inquiry/report on User-Menu AuthorizationsImportantComplexity Rank:10 of 10E. FosterPreparation Date:2010-5-23	
Inputs: DM_UsrMnuA_BR — User-Menu Authorizations (E08) DM_User_BR — System Users (E03) DM_MenuD_BR — System Menu Definitions (E05)		
Outputs: Monitor / Printer		
Validation Rules: None		
 Special Notes: 1. It will be possible to query User-Menu Authorizations by any of the following options: 1.1 By User Code and Menu Code 1.2 By User Code and Menu Sequence Number 1.3 By User Login and Menu Name 1.4 By User Login and Menu Sequence Number 1.5 By User Login, System Name, and Menu Name 1.6 By User Login, System Name, and Menu Sequence Number 2. Each option will invoke one of four sub-operations (AS_UsrMnuA_I1, AS_UsrMnuA_I2, AS_UsrMnuA_I6). 3. The operation makes use of the logical view DM_UsrMnuA_LV1, which joins User-Menu Authorizations (E08) with System Menu Definitions (E05), System Users (E03), and System Definitions (E01). 		
START: /* Inquire */ While User Wishes to Conti Present the User wit Depending on the User End-While; STOP	inue th the options mentioned above; lser's choice, invoke one of the sub-operations;	
Outline for AS_UsrMnuA_I1: START While User Wishes to Continue Prompt user for Identification Code; Starting at that point in DM_UsrMnuA_LV1, Load a Virtual Data Collection Object with all records until End-of-File, ordering by User Code and Menu Code; Display the Virtual Data Collection Object; End-While; STOP		
{The other sub-operations w	<i>w</i> ill be similar}	

Figure 3.20: Operations Specification for Managing User-System Authorizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_UsrSysA_AO / AS_UsrSysA_ZO Facilitates addition/deletion of User-System Authorizations Mandatory 7 of 10 E. Foster 2010-5-23	
Inputs: Information on new or already user-system authorizations DM_UsrSysA_BR — User-System Authorizations (E09) DM_User_BR — System Users (E03) DM_SystemD_BR — System Definitions (E01)		
Outputs: DM_UsrSysA_BR — User-System Authorizations (E09)		
 Validation Rules for Adding: 1. User Code must previously exist in DM_User_BR 2. System Code must previously exist in DM_SystemD_BR 		
Special Notes: None		
Operation Outline: See Generic ADD / DELETE pseudo-code.		

Figure 3.21: Operations Specification for Inquiry/Report on User-System Authorizations

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Operation Spec Author:	Dynamic Menu Interf Administrative Specir AS_UsrSysA_IO / A Facilitates inquiry/rep Important E. Foster	face Designer (DMID) fication Management \S_UsrSysA_RO port on User-System Complexity Rank: Preparation Date:) (ASM) Authorizations 10 of 10 2010-5-23
Inputs: DM_UsrSysA_BR — User- DM_User_BR — System U DM_SystemD_BR — System	System Authorizations sers (E03) em Definitions (E01)	s (E09)	
Outputs: Monitor / Printer			
Validation Rules: None			
 Special Notes: 1. It will be possible to query User-System Authorizations by any of the following options: 1.1 By User Code and System Code 1.2 By User Login and System Name 1.3 By User Login and System Sequence Number, 2. Each option will invoke one of four sub-operations (AS_UsrSysA_I1, AS_UsrSysA_I2, and AS_UsrSysA_I3). 3. The operation makes use of the logical view DM_UsrSysA_LV1, which joins User-Menu Authorizations (E08) with System Menu Definitions (E05), System Users (E03), and System Definitions (E01). Operation Outline: START: /* Inquire */ 			
Present the User with the options mentioned above; Depending on the User's choice, invoke one of the sub-operations; End-While; STOP			
Outline for AS_UsrSysA_I1: START While User Wishes to Continue Prompt user for Identification Code; Starting at that point in DM_UsrSysA_LV1, Load a Virtual Data Collection Object with all records until End-of-File, ordering by User Code and System Code; Display the Virtual Data Collection Object; End-While; STOP			
{The other sub-operations w	<i>i</i> ll be similar}		

Figure 3.22: Operations Specification for Managing Organization-System Mappings

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Administrative Specification Management (ASM) AS_OrgSysM_AO / AS_OrgSysM_ZO Facilitates addition/deletion of Organization-System Mappings Mandatory 7 of 10 E. Foster 2010-5-23	
Inputs: Information on new or already existing organization-system mappings DM_OrgSysM_BR — Organization-System Mappings (E10) DM_OrgD_BR — Participating Organizations (E02) DM_SystemD_BR — System Definitions (E01)		
Outputs: DM_OrgSysM_BR — Organization-System Mappings (E10)		
 Validation Rules for Adding: Organization Code must previously exist in DM_OrgD_BR System Code must previously exist in DM_SystemD_BR 		
Special Notes: None		
Operation Outline: See Generic ADD / DELETE pseudo-code.		

Figure 3.23: Operations Specification for Inquiry/Report on Organization-System Mappings

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Operation Spec Author:	Dynamic Menu Interface Designer (DMID)Administrative Specification Management (ASM)AS_OrgSysM_IO / AS_OrgSysM_ROFacilitates inquiry/report on Organization-System MappingsImportantComplexity Rank: 10 of 10E. FosterPreparation Date: 2010-5-23		
Inputs: DM_OrgSysM_BR — Orga DM_OrgD_BR — Participat DM_SystemD_BR — Syste	Inputs: DM_OrgSysM_BR — Organization-System Mappings (E10) DM_OrgD_BR — Participating Organizations (E02) DM_SystemD_BR — System Definitions (E01)		
Outputs: Monitor / Printer			
Validation Rules: None			
 Special Notes: 1. It will be possible to query Organization-System Mappings by any of the following options: 1.1 By Organization Code and System Code 1.2 By Organization code and System Sequence Number 1.3 By Organization Name and System Sequence Number 1.4 By Organization Name and System Name 2. Each option will invoke one of four sub-operations (AS_OrgSysM_I1, AS_OrgSysM_I2, AS_OrgSysM_I4). 3. The operation makes use of the logical view DM_OrgSysM_LV1, which joins Organization-System Mapping (E10) with Participating Organizations (E02) and System Definitions (E01). 			
Operation Outline: START: /* Inquire */ While User Wishes to Continue Present the User with the options mentioned above; Depending on the User's choice, invoke one of the sub-operations; End-While; STOP			
Outline for AS_OrgSysM_I1: START While User Wishes to Continue Prompt user for Identification Code; Starting at that point in DM_OrgSysM_LV1, Load a Virtual Data Collection Object with all records until End-of-File, ordering by Organization Code and System Code; Display the Virtual Data Collection Object; End-While; STOP			
{The other sub-operations w	vill be similar}		

Figure 3.24: Operation Specification for the ASM Subsystem Menu

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) End-user Access control (EAC) AS_Menu1_XO Presents the ASM subsystem menu and processes the user request Mandatory 8 of 10 E. Foster 2010-5-23	
Inputs: User_Login_Name (accep DM_User_BR — System U	ted as input parameter) Isers (E03)	
Outputs: Monitor display		
Validation Rules: None		
 Special Notes: Present a static window panel with all the ASM user options as represented in the UITC (see chapter 4). Check for each option and call the appropriate method in each related class. Remember, each operation is to be implemented as a class. Each class that represents an operation should have a static method that is invoked from this operation. This operation itself is to be implemented as a controller (driver) class. 		
Operation Outline:		
START Use User_Login_Name to retrieve information on the current user from DM_User_BR; Display the user's name on screen; While (User wishes to continue) do the following: Present the user with a static menu of user options (as specified in chapter 4); Check for, and process the user's request; End-While; STOP		

3.3 End-user Access Control (EAC) Subsystem

The operation specifications for operations in the End-user Access Control (EAC) subsystem will be provided in this section. These operations relate to the generation of the user interface required by each user, based on information stored in the underlying database.

Two categories of users will be facilitated:

- End Users: These users will only have access to system resources to which they have been assigned privileges. Their menus will be loaded based on these options. Options to which a user is not authorized will not be loaded.
- Administrators: These users will have access to the ASM subsystem. They will be able to define and manage various system constraints.

3.3.1 System Login

Operation DM_Login_XO: The system will prompt the user for a login name and password. The System Users table (E03) is checked to validate the user entry:

- If the user does not exist, an error message is returned, and the user is allowed to attempt login again.
- If the user is an administrator (classification code "Adm"), the System Constraints Definition Subsystem menu is presented (see section 3.3.2 below)
- If the user is an end user (classification code "End"), a more detailed process is followed as explained in section 3.3.2 below.

3.3.2 End User Processing versus Administrative Processing

End user processing will be managed by the following utility operations:

- a. **Operation EA_Menu1_XO:** The logical view **DM_UsrSysA_LV1** which provides the **User's System Overview** (see figure 2.4) is to be loaded into a virtual table (Java JTable or Delphi DB Grid) that contains a list of all systems to which the user is authorized. The user should to be able to select a particular system by clicking on it. This will then cause a second operation to be invoked.
- b. Operation EA_Menu2_XO: The logical view DM_UsrMnuA_LV1 which provides the User Menus Summary (of figure 2.4) to be loaded into a virtual table that contains a list of menus corresponding to the system that was selected in the previous operation (EU_Menu1_XO). Each menu represents a subsystem to which the user has access privilege. The user should be able to select a particular subsystem (menu) by clicking on it. This will then cause a second operation to be invoked.
- c. Operation EA_Menu3_XO: The logical view DM_UsrOprA_LV1 which provides the User Operations Summary is to be loaded into a virtual table that contains a list of menus operations to which the user is authorized, and based on the subsystem (menu) that was selected in the previous operation (EU_Menu2_XO). The user should be able to select a particular operation by clicking on it. This will then caused the required operation to be invoked.

Administrative processing will be managed by the utility operation **AS_Menu1_XO**. This operation will present a static window panel with all the ASM user options as represented in the *user interface topology chart* (UITC) of chapter 4. Each major operation is to be implemented as a class with a static method that is invoked from **AS_Menu1_XO**.

3.3.3 Operation Specifications for the End-user Access Control Subsystem

The specifications for the above-mentioned operations follow (figures 3.25 - 3.29).

Figure 3.25: Operation Specification for System Login

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) Generic DMID Login DM_Login_XO Facilitates user login Mandatory 6 of 10 E. Foster 2010-5-23	
Inputs: End-user login information DM_User_BR — System U	Jsers (E03)	
Outputs: Monitor display		
 Validation Rules for Login: 1. User Login Name must previously exist in DM_User_BR 2. Password specified must be in non-display mode, and must match the password stored in DM_User_BR 		
Special Notes: None		
Operation Outline:		
START While (User wishes to conti Prompt for Login and Pas Use Login to check for a v If (Login and Password ar If (the User Classificatio Call operation AS_N End-If; If (the User Classificatio Call operation EA_N End-If; End-If; Else Display Message ("In End-While; STOP	nue) do the following: sword; ralid record in DM_User_BR ; e both valid) n on file is "Adm") Menu1_XO with the User Login as an input argument; n on file is "End") Menu1_XO with the User Login as an input argument; nvalid user or password");	

Figure 3.26: Operation Specification for the ASM Subsystem Menu

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) End-user Access control (EAC) AS_Menu1_XO Presents the ASM subsystem menu and processes the user request Mandatory 8 of 10 E. Foster 2010-5-23	
Inputs: User_Login_Name (accep DM_User_BR — System U	ted as input parameter) Jsers (E03)	
Outputs: Monitor display		
Validation Rules: None		
 Special Notes: Present a static window panel with all the ASM user options as represented in the UITC (see chapter 4). Check for each option and call the appropriate method in each related class. Remember, each operation is to be implemented as a class. Each class that represents an operation should have a static method that is invoked from this operation. This operation itself is to be implemented as a controller (driver) class. 		
Operation Outline:		
START Use User_Login_Name to retrieve information on the current user from DM_User_BR; Display the user's name on screen; While (User wishes to continue) do the following: Present the user with a static menu of user options (as specified in chapter 4); Check for, and process the user's request; End-While; STOP		
Note: This operation really belongs to the ASM subsystem (see figure 3.24). However, in the interest of clarity, it is also presented here		

Figure 3.27: Operation Specification for the EAC Subsystem Menu1

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author:	Dynamic Menu Interface Designer (DMID) End-user Access control (EAC) EA_Menu1_XO Presents the user with a menu of systems to which he/she is authorized, and processes the user request. Mandatory 10 of 10 E. Foster	
Preparation Date:	2010-5-23	
User_Login_Name (accept DM_UsrSysA_LV1 — User	ted as input parameter) 's System Overview (see figure 2.4)	
Outputs: Monitor display		
Validation Rules: None		
 Special Notes: The content of this menu depends on the data obtained from the logical view DM_UsrSysA_LV1. This operation itself is to be implemented as a service class with a static method that can be invoked from another class. 		
Operation Outline:		
START Let UsrSystemsList be a v Use the User_Login_Name and store this into Use Display the User_Login_Na While (User wishes to contin Present the user with UsrS If (the user selects a syste Call operation EA_Menu End-If; End-While; STOP	irtual table (for example, a Java JTable or Delphi DB Grid); to retrieve from DM_UsrSysA_LV1 , a list of all systems to which the user is authorized, srSystemsList; ame on screen; nue) do the following: SystemsList; // the user will select a system by clicking on the row, or quit m) a2_XO, passing the User_Login_Name and the selected System Name as arguments;	

Figure 3.28: Operation Specification for the EAC Subsystem Menu2

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) End-user Access control (EAC) EA_Menu2_XO Presents the user with a menu of subsystems (for a given system) to which he/she is authorized, and processes the user request. Mandatory 10 of 10 E. Foster 2010-5-23	
Inputs: User_Login_Name and User_System_Name (accepted as input parameter) DM_UsrMnuA_LV1 — User's Menus Summary (see figure 2.4)		
Outputs: Monitor display		
Validation Rules: None		
 Special Notes: The content of this menu depends on the data obtained from the logical view DM_UsrMnuA_LV1. This operation itself is to be implemented as a service class with a static method that can be invoked from another class. 		
Operation Outline:		
START Let UsrMenusList be a virtual table (for example, a Java JTable or Delphi DB Grid); Use the User_Login_Name and User_System_Name to retrieve from DM_UsrMnuA_LV1, a list of all subsystems (i.e. menus) to which the user is authorized, and store this into UsrMenusList; Display the User_Login_Name and User_System_Name on screen; While (User wishes to continue) do the following: Present the user with UsrMenusList; // the user will select a menu by clicking on the row, or quit If (the user selects a menu) Call operation EA_Menu3_XO, passing the User_Login_Name, the User_System_Name, and the selected Menu Name as arguments; End-If; End-While; STOP		

Figure 3.29: Operation Specification for the EAC Subsystem Menu3

Operation Biography: System: Subsystem: Operation Name: Operation Description: Operation Category: Complexity Rank: Operation Spec Author: Preparation Date:	Dynamic Menu Interface Designer (DMID) End-user Access control (EAC) EA_Menu3_XO Presents the user with a menu of operations (for a given system and subsystem) to which he/she is authorized, and processes the user request. Mandatory 10 of 10 E. Foster 2010-5-23	
Inputs: User_Login_Name, User_System_Name, and User_Menu_Name (accepted as input parameters) DM_UsrOprA_LV1 — User's Operations Summary (see figure 2.4)		
Outputs: Monitor display		
Validation Rules: None		
 Special Notes: The content of this menu depends on the data obtained from the logical view DM_UsrOprA_LV1. This operation itself is to be implemented as a service class with a static method that can be invoked from another class. 		
Operation Outline:		
START Let UsrOperationsList be a virtual table (for example, a Java JTable or Delphi DB Grid); Use the User_Login_Name and User_Menu_Name to retrieve from DM_UsrOprA_LV1, a list of all operations to which the user is authorized, and store this into UsrOperationsList; Display the User_Login_Name, the User_System_Name, and the User_Menu_Name on screen; While (User wishes to continue) do the following: Present the user with UsrOperationsList; // the user will select an operation by clicking on the row, or quit If (the user selects an operation) Call the selected operation via its Operation Implementation Name, i.e. OpIName; End-If; End-While; STOP		

3.4 Recommended Reading

[Foster, 2010] Foster, Elvis C. *Software Engineering: A Methodical Approach*. 2010. Bloomington, IN: Xlibris Publishing. See lecture 12.

4. User Interface Specification

4.1 User Interface Topology

The user interface design is based on Schneiderman's *object-action interface* (OAI) model for user interfaces (see [Schneiderman, 2005]). The real benefit of this approach is that it is consistent with the way people tend to think: People do not think about the functional intricacies of their daily activities; rather, they think about objects and what they desire to do with them. Because of the natural fit to the typical thought process on the job, user learning will be enhanced.

As mentioned in the previous chapters, users will have access to the DMID via the EAC subsystem and the ASM subsystem. The menu system will be hierarchical, as represented in the user interface topology chart (UITC) of figure 4.1.





4.1 User Interface Topology (continued)

Figure 4.1: User Interface Topology Chart for the DMID (continued)



4.2 Message Specification

All user messages will be stored in a system-wide message file (as a database table) called **DM_Message_MF** (see entity E11 of figure 2.3). Unique codes will be given to each message so they can be accessed easier. Messages will be displayed in the form of pop-up messages.

4.3 Help Specification

The system will host a hypermedia-based help system. Users will access the hypermedia help by clicking appropriate links until they get to the desired help they seek. Additionally, the help system will be organized according to the UITC, so that help will be provided operation-by-operation. As the system matures, the help system can be improved to include context-sensitivity.

4.4 Recommended Readings

[Foster 2010] Foster, Elvis C. 2010. *Software Engineering: A Methodical Approach*. Bloomington, IN: Xlibris Publishing. See lectures 9 – 13.

[Schneiderman 2005] Schneideman, Ben. 2005. *Designing the User Interface* 4th ed. Reading, MA: Addison-Wesley.

5. Summary and Concluding Remarks

This design specification has outlined the blueprint for the Dynamic Menu Interface Designer (DMID) — a software component that can be used to generate menu systems for information systems and menudriven software systems. It includes the following:

- An overview of the system, including problem definition, proposed solution, and system architecture. The overview includes an information topology chart (ITC) that identifies the key information entities managed by the software system.
- The database specification includes a set of design conventions, naming conventions, and an O/ESG for each of the information entities comprising the system.
- Operations specification includes a discussion of the basic system design, followed by an extended operation specification (EOS) for each operation comprising the system.
- The user interface specification outlines the menu hierarchy via a user interface topology chart (UITC). It also includes a blueprint for handling system messages, and a help system.

Version 1 of the software does not include a functional help system. This will be added as an enhancement.