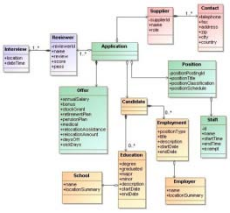
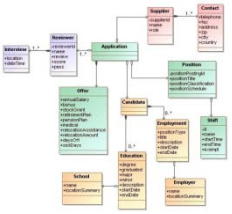


Embarcadero ER/Studio 9.0.1

vs.

Computer Associates ERwin r8



Seán Francis
Database Administrator Consultant

Table of Contents

Document Control

Change Record

Date	Author	Version	Change Reference
March 23, 2011	Seán Francis	1	Draft

Reviewers

Name	Position

Terms Used

Term	Meaning
CASE	Computer Aided Systems Engineering
CTAS	Create Table As Select
DBA	Database Administrator
DDL	Data Definition Language
DM	Data Modeling
ERD	Entity Relationship Diagram
I.T.	Information Technology
PL/SQL	Procedural Language/SQL
RDBMS	Relational Database Management System
RI	Declarative Referential Integrity
SQL	Structured Query Language
TNS	Transparent Network Substrate
UI	User Interface

1. Introduction.

This document provides a comparative review of two of the leading data modeling software tools, Embarcadero's ER/Studio and Computer Associates' ERwin. An Oracle database backend was used to complete the review process.

Product trial or evaluation periods are usually quite short. In that limited time a purchasing decision usually has to be made. It is therefore important that most of that time be spent evaluating the software by actually using it, rather than researching how to use the software. This document is not an exhaustive study of every comparable feature available. Neither is it based on weeks or even months of hands on use of either tool. This document provides an overview of how quickly and easily these tools can be installed, configured and used productively. Each tool was evaluated in the following five categories:

- Installation& Configuration;
  - How simple, quick and easy is it to get up and running?
- Ease of Use;
  - Do these tools have shallow learning curves and are they intuitive to use?
- Features;
  - Can these tools do what you need?
- Performance;
  - Are these tools quick and efficient?
- Support;
  - When you need help is it readily available?

Ease of use, features and performance were assessed using a set of ten tasks. They represent the most common tasks a production or development DBA or designer would likely undertake when first using a data modeling tool. Each tool was evaluated under identical conditions, using comparable resources from their respective manufacturers.

1.1Hardware & Software Environment.

The tools were evaluated in the following hardware and software environment:

TIER	SOFTWARE	HARDWARE
Workstation	ER/Studio Data Architect XE Version 9.0.1 Build 6767	Intel Core i7-930 Quad Core 2.8 GHz
	ERwin Data Modeler Standard Edition 8.0.0.2589	12 Gb DDR3-1066 MHz SDRAM
	Windows 7 Professional 64 bit SP1	2 x 1 Tb 7200 rpm SATA HDs
Database Server	Oracle Database 11g Release 2 (11.2.0.1.0) 32 bit	Intel Pentium D Dual Core 2.8 GHz
	Oracle Enterprise Linux Release 5 Update 4 32 bit	4 Gb DDR2 533 MHz SDRAM
		1 x 250 Gb 7200 rpm SATA HD (OS)
		1 x 1 Tb 7200 rpm SATA HD (database)

1.2About The Author.

Seán Francis has worked in the I.T. industry since graduating with an honors degree in Computer Science from De Montfort University(UK) in 1989. He has been a database administrator since 1990 specializing in Oracle. He has also worked with SQL Server, Sybase and Watcom SQL/SQL Anywhere Studio. His experience with CASE/data modeling tools dates back to Oracle's original CASE tools, including CASE*Dictionary and CASE*Designer. Over the years he has worked extensively with Oracle Designer as well as LBMS Systems Engineer, CA ERwin, Quest Toad Data Modeler, Embarcadero ER/Studio and Oracle SQL Developer Data Modeler.

He has consulted for many companies in many different industries across the US and UK. These have ranged from large multi-national corporations to small companies with few I.T. staff. He holds Oracle Certified Professional DBA certifications for Oracle database versions 7.3 through 11g. He is currently studying for the MCTS and MCITP SQL Server certifications.

1.3 Executive Summary.

Historically, data modeling tools have been notoriously difficult to use. Their steep learning curves and intimidating user interfaces have often required years of hand on use to fully master the breadth of functionality they provide. In addition, product cost, infrastructure requirements and a dearth of third party resources have all been significant obstacles to the widespread adoption of these tools.

Since their inception, database application systems have continued to grow in both number and size. Without using data modeling tools, it is far more difficult to build such systems with performance and scalability in mind from the outset. Consequently, when the challenges of growth reach a tipping point, organizations have often responded by adding hardware to duplicate data and cover up the inadequacies of underperforming systems. This does not however solve the underlying issues. Adding more hardware often exacerbates the problem both financially and technically.

Many companies are realizing the financial benefits of analyzing the data they already own. Two key pieces of information are needed before the benefits become apparent. First, knowing what data exists and second, how it is structured. Using a data modeling tool is an ideal way to gather this type of information and to start documenting and maintaining database systems in an efficient and cost effective manner. This is just one example of how a data modeling tool can add significant value to an organization's I.T. portfolio.

What is needed is a data modeling tool which is easy to learn, produces results rapidly and has the expansion capabilities to grow as the organization's demands upon it become more sophisticated. Two of the I.T. industry's leading data modeling tools are Embarcadero's ER/Studio and Computer Associates' ERwin. The table below provides a summary of how each fared in a number of important categories:

Category	ER/Studio	ERwin
Installation and Configuration	7/10	8/10
Ease of Use, Features, Performance:		
• Reverse Engineering	8/10	5/10
• Physical Server Model Modification	8/10	7/10
• Data Definition Language Generation	8/10	6/10
• Schema Comparison	9/10	6/10
• Impact Analysis	4/10	4/10
• Native Object Type Support	9/10	6/10
• Logical Data Model Creation	9/10	7/10
• Logical To Physical Transformation	8/10	7/10
• Diagram Layout Editing	5/10	8/10
• Printing Support	8/10	9/10
Support	9/10	8/10
Totals:	**92/120**	**81/120**
Percentages:	**77%**	**68%**

As can be seen, there is a clear difference between these two data modeling tools. This review focuses on ease of use and time taken to produce real results. On average, it took 2 to 3 times longer to achieve the same thing in ERwin as it did in ER/Studio. The fundamental reason was the user interface. A data modeling tool which is easier to learn and easier to navigate will be used more productively over one that is not.

Based upon the tests conducted here, ER/Studio is clearly the data modeling tool of choice.

2. Product Installation& Configuration.

When installing a software product for the first time, it should be simple, quick and hassle free. The new software should happily coexist with existing software and not require any significant system configuration changes to accommodate it. Avoiding copious amounts of installation documentation is also generally considered a good thing if you want to be up and running as fast as possible.

2.1 ER/Studio Installation.

Clicking on the erda_901.exe executable produced the first problem. The PC's security software detected "suspicious behavior". The ER/Studio DA Installation was trying to set itself up to run each time the computer is started. I felt this was unnecessary, so opted to deny this activity. I was then greeted with a welcome screen and clicked Next. This caused a box to appear titled, "ER/Studio Data Architect License Required". The message text informed me a valid license is required to complete the installation and would I like to re-try registering? If I selected No, then setup would exit. I was unaware of a previous attempt to register so was rather confused by the offer to re-try it. I clicked the Yes button. The same box from before reappeared. I had managed to find an infinite loop while trying to install the product!

My only option was to start over and tell the security software to allow the alleged "suspicious behavior". The welcome screen appeared as before and clicking Next displayed an Embarcadero Product Registration box. Completing the relevant details and clicking the Register button displayed the obligatory Software License and Support Agreement, which I accepted. The installation was underway, but was interrupted again by the security software. This time ProbeRepoServers.exe was detected as trying to act as a server. I decided to temporarily allow this activity and the installation completed with no further incident.

2.2 ER/Studio Configuration.

The next objective was to point ER/Studio at an existing database running on another machine. This would test its configuration and connectivity setup. Running ER/Studio Data Architect for the first time displayed a dialogue box offering three options of which I picked Reverse-engineer an existing database. This caused the Reverse Engineer Wizard to appear which guided me through the process of creating a Native/Direct Connection to the remote Oracle 11g database. The Database Type, User Name and Password were self-explanatory, less so Datasource. In my case, Datasource meant the Oracle TNS connect string for the remote database. There was one already defined which had been successfully used to connect to the remote database via Oracle SQL*Plus. The Reverse Engineer Wizard was not able to use itand instead immediately reported an "OCI" error. The reason was ER/Studio is not able to use the 64 bit Oracle client networking software. The workaround was to install the 32 bit Oracle 11g Release 2 Runtime Client.

2.3 ERwin Installation.

Clicking on the CA ERwin Data Modeler r8.exe executable caused two alerts to be reported by the PC security software. ERInstallShellSetup_DMCR.exe and ERwin_DM.exe both caused "suspicious behavior" alerts. These related to changes to the Windows registry and is normal for new software installations, so I ignored them. The installation went very smoothly until it appeared to split into two threads. One installation window had a title of "CA ERwin Data Modeler r8 InstallShield Wizard" and the other had a title of "CA Erwin Installation Wizard". Both seemed active. Both had Next buttons. Seeing both was confusing. The CA ERwin Installation Wizard window seemed to take over and the installation completed successfully with no further incidents. After the installation was complete, it was a simple case of dropping in a license file into the appropriate folder on the hard drive and ERwin was ready to go.

2.4 ERwin Configuration.

Reverse engineering functionality was used to test ERwin's remote database connectivity configuration. The Reverse Engineer Select Template screen allows the selection of the target database platform and version. The next screen allows various options to be set, including the name(s) of the schemas to be reverse engineered. The next screen is called Oracle Connection and is where the authentication method, username and password are specified. Selecting Database Authentication enables the User Name and Password fields. The Connection String value field is where the Oracle TNS connect string is entered and clicking the Connect button initiates the connection to the remote database. With the 64 bit Oracle client software installed, ERwin was unable to connect to the remote Oracle 11g database. The connection attempt hung for several moments before reporting an ORA-01041 error. After installing the 32 bit Oracle 11g Release 2 Runtime Client, a database connection was successfully established.

2.5 Summary.

Both tools were installed under the exact same conditions with the exact same PC security software keeping an eye on what was going on. ERwin was less problematic to install, though ER/Studio was quicker to report the initial connectivity issue. Neither tool would connect to the remote Oracle database using the 64 bit Oracle client. Windows 64 bit operating systems have been around since the introduction of Windows XP. Due to their ability to address more memory than their 32 bit counterparts, the 64 bit option has become increasingly popular. Oracle have provided 64 bit versions of their software for the Windows platform for several years now. It is therefore curious why neither tool fully supports this architecture natively. Once this detour was successfully negotiated, both tools were able to connect to Oracle.

3. Task Evaluations.

Ten tasks were devised which represent the most common a production or development DBA or designer would likely undertake when first using a DM tool. Each tool was assessed in terms of how simple, quick and easy it was to complete each task.

3.1 Reverse Engineering.

A production DBA will often inherit a database infrastructure with little or no documentation. What documentation does exist will likely be out of date. Production DBAs are more effective if they know how the production data is structured and how it is accessed. Reverse engineering a production schema is the best way to accurately document how the data is structured. A server model diagram will immediately highlight several important aspects of the physical database design. For example, the presence of RI, including the use of primary and foreign keys. In Oracle, the presence of RI helps the database cost based optimizer calculate the most efficient way to access data.

3.1.1 ER/Studio.

File → New brings up a dialogue box which includes the Reverse-engineer an existing database option. Choosing this option invokes the Reverse Engineer Wizard. This is a 5 screen dialogue which guides you through the reverse engineering process. It includes logging into the target database, choosing the schema owner(s) to reverse engineer, choosing which object types to include, choosing some dependency information and finally a summary of your choices. Clicking the Finish button starts the reverse engineering process.

The first task was to reverse engineer the Oracle SYS schema. This schema contains the Oracle database data dictionary and consists of nearly 1,000 tables. A good stress test for any DM tool. ER/Studio successfully reverse engineered this schema in an average time of 13:00 (mm:ss) with no errors.

The second task was to reverse engineer a custom schema consisting of 10 tables, 10 indexes, 10 primary keys, 11 foreign keys, 5 sequences and a PL/SQL package. This was done in two passes to test how easily multiple reverse engineering sessions could combine their results into a single model. The first pass, which reverse engineered all objects except the sequences and PL/SQL package, only took a matter of seconds. It created a logical model and a physical model. Both diagrams were neatly laid out using the orthogonal layout option.

The second pass, which captured the sequences and PL/SQL package, actually created a whole new model for just those objects. ER/Studio does not use the Reverse Engineer Wizard to add additional objects to an existing model. Instead, it uses the Compare and Merge Utility. This did seem counter-intuitive, but this utility works well and is very flexible. The current model can be compared to another pre-existing model, a SQL file or a live database. You can select the object types you wish to compare and the final screen allows the merging of the differences into the current model.

3.1.2 ERwin.

Actions → Reverse Engineer displays the first of 3 dialogue boxes which guide you through the reverse engineering process. The first dialogue allows the selection of Physical or Logical/Physical along with the Target Database and Version, which were Oracle and 10.x/11.x respectively. Clicking Next displays the Set Options dialogue. Here you can select the source for the reverse engineer process. This is a choice between a Database and a Script File. You also select the different object types you want to include and specify the owner(s) of those objects. You cannot specify which objects to reverse engineer, only which object types. Clicking the In button does allow you to list the tablespaces to consider when reverse engineering tables. Hence, to reverse engineer a very precise subset of a database schema's tables, you would have to re-organize the database by moving just those tables into a specific tablespace, then include that tablespace when reverse engineering the table object type. That may not be possible or even practical for a production database.

Reverse engineering the Oracle SYS schema proved problematic. After less than a minute, ERwin threw an "ORA-600" error, closely followed by an "ORA-932: Inconsistent data types: expected NUMBER got –". The process did complete in an average time of around 7:30 (mm:ss).

Reverse engineering the test schema took just a few seconds. The logical and physical models were formatted neatly in a nice graduated blue hue. Very easy on the eye. Running the reverse engineer process for a second time to add the sequences and PL/SQL package had a twofold effect. First, it caused an "ORA-942: table or view does not exist" error

message. Second, it created a new model with just those object types, which is exactly what ER/Studio did. The Complete Compare function, available via the Actions pull down menu, must be used to merge schema differences into a single model. It can only merge differences between two models inside ERwin, rather than between a model and a running database. The Complete Compare function uses “Left Model” and “Right Model” terminology to identify what will be compared to what. Once the left and right models have been determined, object types are selected and clicking on the Compare button performs the comparison. Once complete, a Resolve Differences dialogue box is displayed which summarizes what the differences are. For each difference there are arrows which can be clicked to merge the difference left or right. As each arrow is clicked in turn, the difference disappears from the screen leaving a simple click of the Finish button to complete the process. Reviewing the Left Model confirmed the sequences and PL/SQL package had been merged correctly.

3.1.3 Summary.

ER/Studio took longer to reverse engineer the Oracle SYS schema and completed it without any errors. ERwin completed the same operation several minutes faster, but generated 2 errors one of which was potentially quite serious. Its results should be viewed with suspicion. When a reverse engineering session completes, ER/Studio provides a scrollable summary box so you can see what happened at any given stage of the process. ERwin’s Status box just closes and returns you to model view. Reverse engineering can take a while and you may want to leave your desk and return later to see what happened. Having a summary waiting for you is preferable. How schema objects are merged marks a distinct difference between the tools. ER/Studio’s Compare and Merge function is vastly superior to ERwin’s Complete Compare. It offers greater flexibility and is far more intuitive to use. In addition, ERwin’s functionality throws Oracle errors which highlights an underlying issue somewhere.

3.2 Physical Server Model Modification.

Over time modifications to the physical server model are likely to be required. To keep the documentation accurate and up to date, the server model diagram should be changed first, rather than manually altering the database schema directly. DBAs will be more inclined to do this if they can make those changes simply and quickly.

3.2.1 ER/Studio.

The first task was to add a new table to the physical server model diagram and link it to an existing table via a foreign key. A new sequence was also added.

Highlighting the Tables folder in the data model explorer pane, right clicking and selecting New Table opens up the Table Editor. From here it was a simple case of naming the new table and adding the necessary columns. A feature of the reverse engineering process is to allow ER/Studio to create data domains for the captured table columns. These data domains are available for selection via the add column dialogue within the Table Editor. Another neat feature of the Table Editor is its ability to switch to another table without leaving your Table Editor session. Oracle supports a primary key viaa unique index. Adding a primary key to the new table automatically created a unique index whose name was in the form *PKnn*. This can easily be changed to something more meaningful, but it would have been nice if the index name had defaulted to *table_name_PK* or something similar. This is possible by using macros, but that is outside the scope of this document. Adding a parent child relationship was simply a matter of clicking the appropriate relationship icon from the toolbar, clicking the parent table then clicking the child table. ER/Studio takes care of adding the foreign key column into the child table, adds “(FK)” after the column name and highlights it in blue. Double clicking the relationship line invokes the Relationship Editor where amongst other things, optionality and cardinality can be changed. For performance and scalability reasons, it is advisable to index foreign key columns in Oracle, but ER/Studio did not create one automatically.

The second task was to alter the structure of an existing table and add a PL/SQL stored procedure to the physical design. Altering a table structure requires the use of the Table Editor again. This is an easy to use and powerful tool. Changing the order of columns, adding indexes, constraints and documentation are just some of the things which can be achieved here. Adding a PL/SQL stored procedure requires a right click on the Procedures folder in the data model explorer pane followed by a left click on the New Procedure option. This invokes the Procedure SQL Editor. The PL/SQL code can be entered directly into this editor or imported from an existing SQL script. Once the code has been created, clicking on the Validate button checks everything is syntactically correct. Once the procedure has been saved it can be

reviewed via the data model explorer pane. Expanding the procedure name folder lists all the dependent objects. A neat feature.

The physical server model diagram remained consistent and stable while several edits were made to it. Even dragging tables with relationships around the screen did not cause any unpleasant side effects. That said, it was possible to leave relationship lines running behind other tables. However, moving relationship lines is very easy and the software works intelligently with you rather than against you. The Table Editor is the heart of the physical server model environment. It is very intuitive and easy to learn.

3.2.2 ERwin.

A right click on Tables in the model explorer pane, followed by a left click on New places a new table object directly on the physical server model diagram. It also adds a table to the Tables folder in the model explorer where you can give it the required name. Dragging the new table object to the desired location on the diagram is simple enough. Right clicking the new table and left clicking Column Properties opens up the Table Column Editor. Clicking the New icon creates a new column giving it the name "_default_" which you can change, along with the data domain, physical data type and any associated key information. Constraints, including nulls, are handled in the tabbed section below this. A neat feature of the Table Column Editor is its ability to edit columns from any table within a single session. Even index definitions can be completed using this editor. As expected, adding a primary key to the table automatically created a unique index to support it. Adding a parent child relationship to the table had an unexpected benefit. Using the Non-identifying relationship icon in the Toolbox toolbar, clicking on the parent then child tables added the required 1 to many relationship. Just as with ER/Studio, ERwin added the primary key of the parent table as a foreign key column to the child table, highlighted it and added "(FK)" after the column name. ERwin also created a non-unique index for the new column automatically. This is good Oracle design. The indexes created by ERwin also follow a readily understandable naming convention which includes the name of the table they are indexing.

Adding a new PL/SQL stored procedure was completed using the Oracle Stored Procedure Editor. The Code box displayed a template CREATE OR REPLACE PROCEDURE statement which could be edited to taste. I chose to cut and paste the required text from a separate Windows Notepad session. I deliberately introduced a syntax error to see if ERwin would detect it. It did not and there was no obvious way to run a validation against the code. Adding a new sequence stayed true to ERwin's style of adding an object first by giving it a name and then adding its properties. Changing the order of two columns in a table was very easy using the Table Column Editor. Highlighting the relevant column and clicking the up arrow icon produced the desired result. A neat feature of changing the column order is the inclusion of an additional arrow icon which, when clicked, sends the highlighted column to the bottom of the column list.

3.2.3 Summary.

Modifying the physical server model using ER/Studio is more immediate and requires fewer detours. It is also more intuitive. The functions which you need are invariably where you expect to find them. ERwin seems to require two steps to achieve something rather than just one. However, ERwin's default naming conventions are better and the automatic creation of an index to support a foreign key column is helpful.

3.3 Data Definition Language Generation.

Once the server model diagram has been changed, the next step is to generate the DDL scripts to affect those changes in a physical database. How these tools handle new database objects and alterations to existing objects will determine how useful they are.

3.3.1 ER/Studio.

The first task was to generate a complete set of DDL for the reverse engineered and modified schema. This would be useful for deployment into non-production database environments. Database → Generate Database opens up the DDL Generation Wizard. The first page of 3 offers the choice of generating a single ordered SQL script file or multiple SQL script files. I initially chose the former to verify ER/Studio understands object ordering and dependency. It does. I ran the generated script unchanged in an Oracle 11g database and it completed with no errors. On page 2 of the Wizard there are a series of tabs for each object type. Not all object types which exist in the model were selected for generation by default. Only tables and indexes were selected, so I manually selected stored procedures, sequences and packages. Page 2 also has a General Options tab which allows further customization of the DDL generation process. Clicking the

Finish button on page 3 generates the DDL. This is followed by a message box which provides the option to load the generated code into ER/Studio's interactive SQL utility, ISQL. From here you can connect to the target database and execute the DDL code directly. However, ISQL is not Oracle's SQL*Plus, so it does require the configuration of an ODBC datasource for Oracle.

The option to generate multiple SQL script files provides a neat solution. When selected, a Physical folder is created on the hard drive, within which several more folders are created, one for each different object type. Under the Tables sub-folder for example, ER/Studio generates a CREATE TABLE script for each individual table.

3.3.2 ERwin.

DDL generation in ERwin is referred to as Forward Engineering. The Forward Engineering function is selected via the Actions menu and selecting the Schema option starts the Forward Engineering Schema Generation dialogue. Everything about the DDL generation is controlled using a single screen. Although this screen does many things, it is very easy to use. How DDL generation behaves can be controlled by the use of database templates. However, that is out of scope for this document. With Schema selected on the left and the required object types selected on the right, clicking the Preview button opens the Oracle Schema Generation Preview window. This displays the generated DDL. You also have the option to generate corresponding drop object statements which is a neat feature. Having a preview feature is useful because clicking the Generate button actually initiates the execution of the DDL against a running database! If you want to save the DDL to a script file, this can be done via the Oracle Schema Generation Preview window or by clicking the Report button.

A few things were noted about the DDL. First, by default, ERwin generates a name for all NOT NULL table column constraints. This is superfluous but can be switched off. Second, also by default, ERwin generates a series of table level triggers to enforce RI for INSERT, UPDATE and DELETE operations. This may or may not be useful for backend development and can be switched off. Third, the CREATE TABLE statement for the table which had its column order changed, retained the original column order. Despite the fact the physical model diagram shows the new order as does the Table Column Editor. After a considerable amount of research, it was found that unchecking the Physical Order Column property on the Options tab within the Forward Engineering Schema Generation dialogue produced the desired result. In ERwin, Physical Order equates to the order in which columns were originally created and if enabled, any subsequent changes to column (display) order are ignored at DDL generation time. The entire batch of DDL commands generated by ERwin all ran successfully in an Oracle 11g database with no errors.

3.3.3 Summary.

DDL generation using ER/Studio is fast, straight forward and provides good documentation. The inclusion of Oracle SQL*Plus PROMPT commands to display what the scripts are doing when executed would make this feature even better. DDL generation using ERwin is less than intuitive, but arguable simpler once you understand the purpose of the Preview, Report and Generate buttons. That said, the time taken to uncover exactly how to generate the desired table column order takes away from ERwin's immediate usability.

3.4 Schema Comparison.

It can sometimes be the case that a process or application works well in a non-production environment and less well in production. Much time can be spent and wasted trying to track down the reasons, especially if the schema in question is quite large. Data sets aside, differences between the production and non-production schemas can sometimes be the cause of these anomalies. Identifying these differences quickly and even generating corrective code does blur the line between a DM tool and a DBA tool. However, do either of these tools help in this regard?

At this point, the physical server model inside the tools differed from the reverse engineered schema used to initially create them. These differences included a new table, primary key, index, sequence and a stored procedure. The column order of another table had also been changed.

3.4.1 ER/Studio.

The first task was to identify the differences between the physical server model held in ER/Studio and the schema held in the database. The Compare and Merge Utility was used to accomplish this. The source and target can also be two

data models and a data model and a SQL file. After providing database connection information, page 4 of the Compare and Merge Utility allows the selection of the object types to compare. Page 5 allows for the fine tuning of specific object comparisons and clicking the Next button initiates the comparison process. Page 6 provides a summary of the results.ER/Studio correctly identified all the differences.

The next task was to generate the necessary SQL code to merge the differences into the target database. The default resolution for the differences is Ignore. Changing this to Merge into Target and clicking the Next button brings up page 7 of the Compare and Merge Utility. Here you can select various object rebuild options and clicking Finish generates the merging SQL script. Just as you could with the DDL Generation Wizard, you can also choose to load the script directly into ISQL. The code to create the new objects was simple enough. The code to alter the table column order was interesting. The table's constraints are dropped then the table itself is renamed. A CTAS operation re-creates the table with its original name and new column order. The constraints are then re-created. The renamed table remains left behind. Although this method works, it should not be considered for a live 24x7 production database unless during a scheduled maintenance window and with sufficient storage to accommodate a copy of the original table. Oracle's online redefinition functionality would be a better option here.

3.4.2 ERwin.

The Complete Compare function was again used to identify the differences between the modified physical server model inside ERwin and the original schema inside the Oracle database. Since Complete Compare needs to compare a 'left' model to a 'right' model, the original database schema had to be reverse engineered into ERwin for a second time. This was done by selecting Database/Script and clicking the Load button which takes you through the same reverse engineering dialogue steps described above. Once again, this did cause ERwin to generate an "ORA-942: table or view does not exist" error message, but the schema was reverse engineered correctly. Once both models were loaded, the Compare Level was set to Database Level and the object types for the 'left' and 'right' models were selected. Clicking on the Compare button opens a Resolve Differences window which displays the summary and details of what is different between the two models. ERwin detected all the differences correctly.

The next task was to generate the SQL code to merge the differences into the target database. This was done in three steps. First, each difference's right pointing arrow was clicked to merge that difference into the 'right' model. This had the effect of making the Right Model equivalent to the modified Left Model. It also highlighted the Right Alter Script/Schema Generation icon in the Impact Analysis toolbar. The second step was to click this icon and select the required options in the Alter Script Schema Generation dialogue box. The Preview button displayed the generated DDL. It contained SQL create commands for the missing sequence, table, index and stored procedure. It also contained SQL alter commands to add the relevant keys to the missing table. Create trigger commands were again included to enforce RI for the missing table, even though these triggers were not reported as a difference between the two models.

There was no code generated to correct the column order for the relevant table, even though this was reported as a difference between the two models. After much research, the method to generate the necessary code was determined. Just as with the previous Forward Engineering session, the Physical Order Column property had to be unchecked and the Default Value Column property had to be changed from Create to Alter. The code which these changes generated differed from the method employed by ER/Studio. The table is renamed, then re-created with the original name and correct column order. This is followed by an INSERT INTO SELECT FROM SQL statement. The CTAS method used by ER/Studio is actually more efficient. Finally, the last step was to click the Report button which saved the generated code to disk.

3.4.3 Summary.

Schema comparison and resolution in ER/Studio was quick and simple to execute using the Compare and Merge Utility. Although the method employed to alter the table structure was less than optimal, it was not unreasonable given ER/Studio is not a dedicated database administration tool. Schema comparison and resolution in ERwin is not as quick mainly due to the need to create an additional model. The Alter Script Schema Generation dialogue box is not intuitive, but did generate the necessary code.

3.5Impact Analysis.

The ability to quickly determine an object's dependencies will help establish the scope of any proposed schema change. Without knowing the scope, changing anything in the database will likely lead to more problems and could ultimately affect application or system availability. In a production environment, impact analysis helps identify what will need to be re-tested as a result of a change. In a development environment, impact analysis can help identify the least expensive or most appropriate solution for a proposed change. The bottom line is, if a DM tool handles database resident code, it should track object dependency and report impact analysis.

3.5.1 ER/Studio.

Within ER/Studio there are no canned reports which summarize an object's dependency tree. However, some of this information can be determined interactively. For example, the Table Editor contains a Dependencies tab which lists the table's dependencies on triggers, functions, procedures, packages and views. In the data model explorer pane, expanding the name of a package or procedure lists the dependent tables. A neat feature is highlighting a dependent table in the data model explorer pane which also highlights the corresponding table in the physical sever model diagram. It is worth noting a word of caution when reverse engineering schemas into ER/Studio. On page 4 of the Reverse Engineer Wizard, there are two check boxes labeled Reverse Engineer View Dependencies and Reverse Engineer Other Dependencies (e.g. Procedures, Triggers, etc.). If those boxes are left unchecked, ER/Studio will just process the objects selected in the wizard. If those boxes are checked, ER/Studio will also find any dependencies (e.g. View, Procedure, Triggers, etc.) missing from objects selected in the wizard. View, procedures, triggers are parsed to populate the dependency tab. Any objects created inside ER/Studio have their dependencies tracked automatically as well.

3.5.2 ERwin.

Object dependency within ERwin is restricted to two areas, both of which are only available interactively. In the Table Editor, the Where Used tab summarizes where a table is referenced. This is limited to which model(s) features the table and which relationships the table is involved in. The Stored Procedure Editor also has a Where Used tab, but this is even more restricted to which model(s) contains that stored procedure.

3.5.3 Summary.

Object dependency and thus impact analysis can only be inferred interactively with either tool. Even then, the functionality is very limited. For a DM tool which handles database resident code, this is poor. Without access to other software tools, the Oracle data dictionary is the best place to determine an object's dependency tree.

3.6 Native Object Type Support.

Some DM tools are tightly coupled with a specific database platform. Others are more generic and cater for multiple platforms. At the logical data modeling stage it does not matter what the deployment database platform will be. At the physical server modeling stage the unique features of the deployment platform should be available to the database designer. A generic physical database design which needs to be manually edited outside of the DM tool makes the use of that tool far less appealing. It is much better to explicitly select the physical features you want to implement from within the tool itself.

3.6.1 ER/Studio.

The first task was to add date range partitioning to one of the tables in the physical server model and generate the DDL to implement it. The Partitions tab in the Table Editor provided this functionality. A partition type pull down menu offered hash, list, composite as well as range partitioning options. Clicking the Add button invokes the Table Partition Editor from where you can define and add the required partitions to the table. Once this is complete, clicking on the DDL tab in the Table Editor shows the full CREATE TABLE command complete with all its associated partitions. Clicking on the Copy button copies the full SQL command to the Windows clipboard from where it can be pasted into your favorite text editor.

At this point, the table inside ER/Studio had a partitioned design and the corresponding table in the Oracle database had a non-partitioned design. Therefore, having a new CREATE TABLE SQL command with partitioning was not overly

useful. What was needed was a series of SQL commands to take care of all the table's constraints and indexes and rebuildit using partitions. Once again, the Compare and Merge Utility provided the solution.

The next taskwas to add a function based index to an existing table inside the physical server model. Double clicking the table in the server model diagram invokes the Table Editor. Clicking the Indexes tab shows all currently defined indexes. Clicking the Add button invokes the Index Editor. Most of the index types can be selected via the Properties tab. The function based index option is selected via the Columns tab. Highlighting the relevant column, then clicking in the check box labeled Function Based changes the Add button into the Add Expr button. Clicking the Add Expr button places the name of the highlighted column in the Selected Keys section of the Index Editor. This is also where you add your own expression syntax. This was not immediately obvious since you can save the function based index definition without providing an expression. The CREATE INDEX DDL was not immediately available. That had to be generated by using the Compare and Merge Utility.

3.6.2 ERwin.

Double clicking on a table in the physical model opens the Table Editor. The Partitions tab provides the functionality to define the necessary partitions. Clicking the Partition Type checkbox allows the selection of the relevant partition type radio button. The middle section of the screen lists all the available table columns on the left and the selected columns on the right. The bottom section of the screen allows the naming and range value for each partition. The value field also contains a single value pull down menu for the MAXVALUE option. A neat feature of this screen is the Edit Properties button for each defined partition. Clicking this button opens the Partition Description Dialog window where all manner of storage options can be specified. The DDL to create the partitioned table was generated by following the Forward Engineering process described above.

Right clicking over a table in the model diagram, then left clicking Index Properties opens the Index Editor. Clicking the New icon offers the option of adding a unique or non-unique index. After giving the new index an appropriate name, selecting an index type of function based proves rather more difficult. There are 8 tabs within the Index Editor and none of them offer this type of index for selection. The Physical tab offers Unique, Bitmap and even Reverse, but not function based. ERwin'sbuilt in help system does not even mention Oracle's function based indexes. After much research, the method by which a function based index can be defined was discovered. The Members tab shows a list of available columns on the left and a list of Index Members on the right. Index Members is essentially a list of table columns included in the index. Even if you need to define an expression using one of the table columns, it must not be selected for inclusion in Index Members. Instead, a left click in the first blank record within Index Members activates the New button towards the bottom of the dialogue box. Clicking on the New button opens a New Expression box into which the function based index expression can be entered as free text. Clicking OK adds the expression to the Column/Expression box and to the Index Members list. Clicking the Close button in the Index Editor saves the new index definition. As with ER/Studio, the DDL was not immediately available from within the Index Editor. It was generated successfully using Forward Engineering.

3.6.3 Summary.

ER/Studio's support of Oracle object types seems comprehensive and uses terminology familiar to those who work with Oracle. ERwin's support of Oracle object types seems equally as comprehensive, but again its UI lets it down. When creating a function based index, nowhere is the term "function based index" ever mentioned on screen and the method used to create one is obtuse. This is one of the more obvious examples where ERwin can do what you need, but doing it is completely unintuitive and somewhat confusing. This makes the tool far less immediately useable.

3.7 Logical Data Model Creation.

All diagrams are essentially about communication and one of the most important is the logical data model. A logical data model or ERD is an essential tool with which to verify the information gathered about the process or business is complete and accurate. Being able to create an ERD quickly and easily for review will greatly enhance a project's chances of success. Do these tools allow for this?

3.7.1 ER/Studio.

File → New brings up a dialogue box where the Draw a new data model option can be selected. Clicking OK changes the display to a model explorer pane on the left and a larger empty model view pane on the right. Expanding the Main Model in the explorer pane lists the types of objects which can be created. These include Entities, Views, Relationships,

Users, Roles and Shapes. A right click on Entities followed by a left click on New Entity opens the Entity Editor. This is basically the logical equivalent of the physical Table Editor. From here a new entity can be named and attributes added.

When the entity is given a name, it is duplicated in the Table Name field. This does give you precise control over object naming, but you have to provide the plural table name yourself. When adding attributes, any previously defined data domains can be selected via the Domain Name pull down menu. Also, the distinction between logical and physical is clearly emphasized when selecting attribute data types. The list of available logical data types goes way beyond the finite list of physical data types you would expect to see for a given RDBMS platform. The list includes data typesTEXT, MONEY and BIT amongst others.

Relationships can be added from either the modeling toolbar or by right clicking Relationships then left clicking New Relationship in the model explorer pane. The Relationship Editor has a neat feature whereby you can add phrases which describe the relationships in plain English. For example, A DEPARTMENT “may employ one or more” EMPLOYEE(s). The Relationship Editor provides the entity names and you provide the connecting phrase. Once defined, the phrases appear on the diagram making it much easier to review with end users. By default, only the attribute names are displayed for each entity. Additional attribute information, for example data type and optionality, can be added by using Diagram And Object Display Options. This is accessed via the View menu.

3.7.2 ERwin.

File → New opens the Create Model Select Template dialogue box. From here you can select the model type from Logical, Physical or both. Clicking OK returns you to the main screen with the Model Explorer pane on the left and a blank page in the model view pane on the right. A new entity is added by right clicking Entities, then left clicking New in the model explorer. Initially, only an entity name can be entered. Attributes are added as a separate action by right clicking on Attributes, then left clicking New. Again, only the attribute name can be entered. Once an attribute exists, right clicking on it and selecting Properties opens the Entity Editor. From here, more attributes can be added, along with their data domains, data types and key information. Just as with ER/Studio, the list of available logical data types is extensive. It includes TEXT, MONEY and BYTE. There was no equivalent for the BIT data type.

After defining an entity, it did not appear in the model view automatically. Right clicking on the entity and left clicking Add To Diagram added it to the diagram. Relationships were added to the diagram by simply clicking on the relationship icons in the Toolbox toolbar. Double clicking on a relationship line opens the Relationship Editor. Here phrases to describe the relationship in both directions can be defined and the relationship itself can be given a name. This name can be added to the diagram. Controlling what is displayed on the diagram, for example attribute data types and optionality, is achieved via Diagram → Diagrams, which invokes the Diagram Editor.

3.7.3 Summary.

Creating an ERD with ER/Studio is a quick and simple exercise. By adding phrases and expanding attributes’ displayable properties, a very readable diagram can be built. Creating the same diagram using ERwin was more work, with a single activity taking several steps compared to ER/Studio’s single step. ERwin certainly has the options to customize how things can be done and how they appear on screen. Unfortunately, you have to go looking for them because its default behavior is less intuitive and more awkward to use. For example, enabling the display of an attribute’s data type. Neither tool does this by default. In ER/Studio, View → Diagram and Object Display Options, followed by clicking the Datatype checkbox displays the attribute data types in justified column format. In ERwin, View → Display Level → Attribute, turns on the display of data types, but the attribute name and data type are merely separated by a colon character which makes it difficult to read. To achieve a justified column format, go to Diagram → Diagrams, click on the Entity tab then click the checkbox labeled Show Attributes/Columns as Grid. It achieves the same result as ER/Studio, but it involves more work to get there.

3.8 Logical To Physical Transformation.

A logical data model can be transformed into a physical server model in a variety of ways depending upon the needs of the organization and the deployment database platform. How do these tools handle things like many to many relationship transformation and logical to physical data type mappings?

3.8.1 ER/Studio.

A modest logical data model built using ER/Studio contained several many to many relationships and some logical data types for which there are no direct physical equivalents in Oracle. These included TEXT, MONEY and BIT. A right click on Main Model followed by a left click on Generate Physical Model in the model explorer pane starts a 5 page dialogue

to generate the physical model. The first page allows you to name the physical model and select the target database platform. The list of supported platforms is impressive and includes support for Oracle from 11g back to Oracle7.x. Subsequent screens allow for object type selection, naming standards and physical storage parameters. The last screen includes an option to handle many to many relationships. The options are create an associative entity or create tables without a relationship. Clicking the Finish button processes the transformation into a physical model. There is also an option to validate the model as well.

ER/Studio correctly decomposed the logical many to many relationships, but refers to the intersecting object as an entity which it is not. The intersecting objects are listed as tables in the model explorer pane and double clicking them on the physical model diagram invokes the Table Editor. Before the Table Editor appears, a message is displayed which warns about editing the *entity* and that doing so would remove the associated one to many relationships from the physical diagram. An additional attribute was added to the intersecting *entity* and the relationships remained intact. Even the generated DDL contained all the relevant code to create the necessary primary and foreign key constraints. The logical data types TEXT, MONEY and BIT were converted to Oracle physical data types CLOB, NUMBER (10,2) and NUMBER (1,0) respectively.

3.8.2 ERwin.

Starting with the same logical data model as with ER/Studio, the process of creating a physical model was started using Actions → Design Layers → Derive New Model. This opened the Derive Model dialogue where the type of model can be selected. Clicking the Physical radio button, allowed further selections to be made for Target Database and Version. These were Oracle and 10.x/11.x respectively. Subsequent screens allowed for the selection of the objects types to be derived and even naming standards could be tuned to taste. Finally, clicking the Derive button generated the physical model.

ERwin also correctly decomposed the logical many to many relationships into two 1 to many relationships, giving the intersecting table the appropriate keys from the related tables. ERwin did not have an issue with changing the structure of the intersecting tables. The logical data types TEXT, MONEY and BYTE were transformed into the physical data types LONG VARCHAR, DECIMAL (19,4) and SMALLINT respectively. The DDL generated to create the tables in an Oracle 11g database executed without errors. The database itself transposed these data types into LONG, NUMBER (19,4) and NUMBER (38) respectively. The main point to note here is the LONG data type in Oracle is really only there for backwards compatibility. It was superseded by the CLOB data type from Oracle8 onwards.

3.8.3 Summary.

ER/Studio's Generate Physical Model dialogue is easy to follow and produces its results quickly. Its handling of logical many to many relationships is a little confusing, so it is perhaps better to decompose those in the logical realm and generate to the physical realm with fewer complications. ERwin's method is a little unintuitive and does produce physical data types which are unfamiliar to most Oracle users. That said, it was equally fast and handled the job with relative ease.

3.9 Diagram Layout Editing.

Large diagrams containing tens or even hundreds of objects can be cumbersome to deal with. It is often easier to understand and manage a small section of the main model at a time. Creating a sub model from the main model is an ideal way to achieve this. Do these tools support sub models and do they maintain links between them? Large diagrams also tend to have many crossed relationship lines. This can be confusing and makes the model less understandable. Can these tools assist with reducing or eliminating crossed lines or is that laborious task left to the human designer?

3.9.1 ER/Studio.

The first task was to create a sub model from the main physical server model. Model → Create Submodel opens the Create Submodel dialogue box. From here it is a simple case of giving the sub model a name and selecting the required

objects. Clicking OK opens the sub model in model view and adds it the model explorer pane. Adding a new column to a table in the sub model added the same column to the same table in the main model. Initially it did not look like it had because the table box in the main model had not been automatically resized to display the additional column. Deleting the column in the main model also removed the column from the relevant table in the sub model. So, the main model and sub model were definitely linked.

The next task was to test how well ER/Studio handles re-drawing a model to reduce or eliminate crossed relationship lines. The physical server model used for this test had 11 tables which could be arranged in such a way as to completely eliminate crossed lines. This version of the model was saved before 5 crossed lines were manually introduced to the layout. ER/Studio has 5 standard layouts called Circular, Hierarchical, Orthogonal, Symmetric and Tree. These can be accessed via the Layout pull down menu or from the Layout and Alignment toolbar.

The Circular layout took a 2 page model and made it into a 7 page model reducing the crossed lines to just one. The Undo function available via the Edit pull down menu did not return the model to its pre-Circular layout, so the saved copy had to be re-opened. The Hierarchical layout created a 5 page model with several tables spanning page boundaries. It eliminated all but one crossed line. The Orthogonal layout produced a 4 page model and eliminated all crossed lines. It also created the most tables which crossed page boundaries. The Symmetric layout created a 9 page model with no crossed lines. Some pages contained a single table or part of a relationship line, so the wastage of space was significant. The Tree layout produced a 3 page model with only one crossed line. It also produced the least number of tables which spanned page boundaries.

The best result in terms of both aesthetic layout and crossed line elimination was Orthogonal. However, the gains made were more than offset by having to re-position multiple tables so they did not span several pages.

3.9.2 ERwin.

Creating a sub model with ERwin simply involves creating an additional diagram which only contains a subset of the objects contained in the main diagram. This is achieved through Diagram → Diagrams. This displays the Diagram Editor. Clicking the New icon adds an additional diagram record to the upper section of the editor's screen. The lower section is where a list of available objects can be selected for inclusion in the new model. The 3 tables selected were added to a new diagram on top of each other. Some drag and drop mouse work separated them into a readable layout. This would have been more difficult and time consuming had many more tables been involved. A column was added to one of the tables in the sub model. The same table in the main model inherited the same change. A neat feature of ERwin's model view is to automatically re-size the table so new columns are immediately visible. Deleting the table column from the main model also deleted the table column from the sub model. Main models and sub models are definitely synchronized correctly.

Just as with ER/Studio, the physical server model was initially laid out with no crossed relationship lines and was spread over 2 pages in an identical fashion. The same 5 crossed lines were introduced to test how well ERwin's different diagram layout options coped with reducing or eliminating crossed relationship lines.

The Circular Layout produced a 4 page model with 2 crossed lines and 3 tables which spanned page boundaries. Unlike ER/Studio, ERwin's Edit → Undo feature returned the model to its exact layout before the Circular Layout was attempted. This avoided having to throw away the changed model and re-open the saved copy of the model. The Hierarchical Layout produced a 4 page model with 2 crossed lines and 4 tables which spanned page boundaries. In addition, two of the relationship lines were drawn behind tables so were only partially visible. The Orthogonal Layout produced a 4 page model with 2 crossed lines and 2 tables which crossed page boundaries. The Symmetric Layout produced a 4 page model and eliminated all crossed lines. It also created 2 tables which spanned page boundaries, but there was adequate space to drag and drop those tables to eliminate that flaw. Doing so would not have compromised the layout of the remaining diagram. Finally, the Tree Layout also produced a 4 page model, but with more crossed lines than it started with, a total of 6. It also had the most tables which spanned page boundaries with a total of 5.

The best result for crossed line elimination was the Symmetric Layout. The best result for model appearance was Orthogonal, but not by much. The Symmetric Layout was very usable and would not have required much more work to eliminate tables spanning pages.

3.9.3 Summary.

Both tools support sub models well with the main difference being ER/Studio calls it a sub model and ERwin does not. At least not in the context of using the tool in a standalone way. Both tools had the exact same pre-built diagram

layout options, but their performance was very different. ER/Studio was consistently very wasteful of page space and produced some bizarre layouts. ERwin consistently kept the page count down and the Symmetric Layout actually eliminated all crossed relationship lines. With ERwin's default subtle use of different blues to color the model, its intelligent Undo feature and its superior diagram layout options, it is the better tool for diagram layout editing. That said, the human designer is still the best option.

3.10 Printing Support.

Diagrams will ultimately end up on paper at some stage. Chances are your diagrams will be printed onto several standard size printer pages. Do these tools provide a detailed print preview function which shows page boundaries? Can you easily avoid printing diagram objects across several pages? Are the pages numbered and can you print specific pages rather than the whole diagram?

3.10.1 ER/Studio.

File → Print opens the print preview window. This shows the entire diagram superimposed onto page boundaries. It will therefore be immediately apparent what the final printed output will look like. Chances are several objects will straddle page boundaries which is usually undesirable. To correct this, click Cancel and return to the model view. By default, this view does not show page boundaries, but they can be switched on. View → Diagram And Object Display Options opens up a dialogue box where you can click the checkbox labeled Show Page Boundaries. Clicking OK returns to the model view where the page boundaries are now visible. From here it is a simple case of dragging and dropping the model's objects to fit wholly within a page. The model view also displays page coordinates in the top left corner of each page.

There is a difference between the page boundary and the printable area. Placing any object too close to the edge of a page will cause cropping. A neat feature within print preview is Page Selection, accessed via the button with a check mark. This allows the selection of specific pages to be printed which is useful if only a small part of the model has changed and you do not want to re-print the entire diagram. By default the entire diagram will be printed, so selecting specific pages with Page Selection effective de-selects them for printing. Greyed out pages will not be printed. Also within print preview, the Printer button allows for the selection of a specific printer and the desired page orientation. The model name and a timestamp can also be selected for printing. In fact, all your printing customizations can be saved by clicking Save Settings, so you only need to do this customization one time.

3.10.2 ERwin.

ERwin has three printing related functions on its File pull down menu. They are Page Setup, Print Preview and Print. The Page Setup function includes the selection of paper size from an extensive list. It allows the selection of page orientation and from the Margins tab, 6 different margin types can be precisely defined. Also on the Margins tab are areas to specify a page header and footer. The Print Preview screen only shows a single page at a time. Moving around the model is done using the Next Page and Prev Page buttons. The Print function, which can also be accessed via Print Preview, is standard fare. Printer selection, Page range and Number of copies are all here. The Print order can be selected via one of two radio buttons, labeled "Over, then down" and "Down, the over". Which pages to print can be specified as a comma separated list, so knowing which way the diagram will be printed is useful.

Visible page boundaries are enabled via the View pull down menu and selecting the Page Grid option. A neat feature of ERwin's printing function is its default understanding of the printable area. Just as with ER/Studio, several tables were deliberately placed very close to page boundaries. Unlike ER/Studio, ERwin accommodates for this with its Page Grid and no cropping was seen when the diagram was printed.

3.10.3 Summary.

The printing function built into ER/Studio takes almost all the guess work out of getting your model onto paper. Its standout feature is its full diagram print preview. ERwin makes you work just a little harder to get what you want onto paper. Its standout feature is handling a page's printable area more intelligently. Both tools default fonts and overall printing style produce good looking diagrams. ERwin's relationship lines are easier to see than ER/Studio's light grey and its better use of page headers and footers just gives it the edge.

4. Product Support.

The greatest software in the world will eventually become unusable and unused if it is not adequately supported. Product support comes in a variety of guises, the primary one being the technical support offered by the manufacturer. This you generally pay for one way or another. Online knowledge bases and support forums are also useful ways to access the information you need to make the most of the software.

4.1 Embarcadero Customer Support.

I used the email option to contact Support via Embarcadero's main website at http://embarcadero.com.The form is very succinct, requiring only basic information including your name, contact details, product details, operating system and a description of the problem. It really only takes a few minutes to complete the form and when you are finished the website generates a case number for you. After which you wait to be contacted. I waited on average between 60 and 90 minutes for an initial reply which, based upon my experience with other software manufacturers, is fairly rapid.

The support analysts provide their full name which does help establish a rapport with the person who is trying to help you. They were all consistently polite without it seeming contrived and their written English was first class. A refreshing change. The solutions and explanations I received were, as far as I could tell, complete and technically accurate.

4.2 Other Embarcadero Support Channels.

The Embarcadero Developer Network, available at http://edn.embarcadero.com, is a very useful resource. It includes product communities, discussion forums and an abundance of technical documentation. Using Google to find information on ER/Studio invariably led back to a page on the Embarcadero Developer Network website. I found it easy to navigate, though a little slow with a tendency to hang for a while at initial login.

4.3 Computer Associates Customer Support.

Running ERwin under evaluation/trial conditions does not qualify for CA's online support offering. You therefore cannot open a support case via CA's support website. Technical support is only provided via their 800 telephone number. A support case was opened on my behalf and I received a call from a support analyst within an hour. That was impressive. The support analyst was polite, professional and knowledgeable to the extent they understood the issue and believed they had a solution. This was emailed to me within a couple of hours, but did not solve the problem.

All other issues, problems and questions were funneled through an ERwin Account Manager. He was extremely helpful, achieved very rapid turnaround on the issues and made several call backs to make sure I had everything I needed. I could not have asked for more, though I suspect this had more to do with me writing an evaluation than it did with CA accommodating product trials per se.

4.4 Other Computer Associates Support Channels.

Access to CA's online support portal, at https://support.ca.com, is free and provides access to CA's Knowledge Base and technical documentation. I found it awkward to navigate and of little use when researching issues. Google was better and enabled me to find http://supportconnect.ca.com which did provide the solution to some of the issues I encountered with ERwin's UI.

4.5 Summary.

Both companies were very generous with their time and access to their technical support personnel. To be fair, I did not make extensive use of either of their official product support channels, so it is difficult to make a definitive comparison. As for online resources, Embarcadero has the edge with regards to site navigation and intelligent document searching.

5. Conclusion.

Any investment in a DM tool is an important and strategic decision for any organization. It is not unreasonable to want to see a return on that investment as soon as possible. The tool chosen should therefore lend itself to ease of use to enable its practitioners to quickly demonstrate its worth.

Any software product's ease of use can be quantified in a variety of ways. These include the style, intuitiveness and navigation of its UI. Its feature set, performance and configuration options are also important criteria to consider. As is the support offered by the manufacturer.

Approaching ER/Studio and ERwin with an equivalent level of knowledge and experience of both, there are clear distinctions between them when it comes to ease of use. Put simply, with ER/Studio features are called what you expect them to be called and related functions are consistently accessed from a single place. With ERwin, the presence of key features is far less obvious and it often takes several steps, using several different dialogues, to achieve something.

Starting to use any new DM tool will invariably involve learning the process and learning the tool itself. By way of example, when using a word processor the most important activity is creating the document itself, not searching for the underline feature.

The bottom line is, with ERwin you will initially spend more time learning how the tools works and less time actually getting your modeling job done. With ER/Studio, you will initially spend more time getting your modeling job done because it is far more intuitive and easier to use.

For that reason, ER/Studio is the superior product.