INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION, *14*(3&4), 279–306

DB2 Universal Database: A Case Study of a Successful User-Centered Design Program

Rick Sobiesiak
IBM Toronto Laboratory

Robert J. Jones
IBM Silicon Valley Laboratory

Scott M. Lewis
IBM Toronto Laboratory

This case study describes the application of user-centered design (UCD) principles to the development of a complex middleware software product—IBM's DB2® Universal Database. Motivated by trade press reviews highlighting ease-of-use problems, a multisite development team utilized IBM UCD to transform ease of use into a featured product attribute. This case study describes the initial application of UCD to DB2[1] and the positive business results that followed: critical acclaim, increased marketshare and user satisfaction, and increased support and adoption of UCD across the IBM database product family. Although ease-of-use objectives, target markets, and human factors practitioners' roles have evolved over 6 years and multiple versions of the product, DB2 continues to achieve positive results with UCD. In addition to the flexibility of the UCD methodology, other factors that influenced DB2's success included man-

The authors acknowledge the contributions of Katherine McMurtrey, who edited this article and made it much more readable than it would have been otherwise. In addition, the thoughtful comments from three anonymous reviewers contributed significantly to the value of this article.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataHub, DataPropagator, DB2, DB2 Universal Database, DB2 UDB, IBM, OS/2, OS/390, OS/400, RISC System/6000.

The following terms are trademarks or registered trademarks of other companies: Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation; Java and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both; UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Requests for reprints should be sent to Rick Sobiesiak, IBM Toronto Laboratory, B2/EB1, 8200 Warden Avenue, Markham, Ontario, Canada L6G 1C7. E-mail: ricks@ca.ibm.com

[1] DB2 is a family of database products that consists of two main groupings: (a) DB2 Universal Database for UNIX, Windows, and OS/2 operating systems; (b) DB2 Universal Database for midrange and large systems, including OS/400, OS/390, and VSE/VM. In this article, we use DB2 to refer to DB2 Universal Database for UNIX, Windows, and OS/2.

agement support, resource commitment, cross-site communication and collaboration, detailed product specifications, and a focus on supporting user tasks.

1. INTRODUCTION

September of 1997 marked the release of DB2 Universal Database Version 5, a relational database management system (RDBMS) product for UNIX®, Windows®, and OS/2®. This was the first version of DB2 developed using the IBM user-centered design (UCD) approach (Vredenburg, Isensee, & Righi, 2002). Although this was the first application of UCD to DB2, it was nevertheless a comprehensive investment, both in terms of resources and organizational commitment.

The results of this investment have been encouraging. DB2 Version 5 met its targeted ease-of-use objectives based on overall user satisfaction measures gathered through design validation tests. In addition, trade press reviews of DB2 Version 5 highlighted major enhancements in ease of use, and post-release customer satisfaction surveys showed significant improvements in ease-of-use ratings. Another encouraging factor—albeit not directly attributable to UCD—was the overall business success of the release, as revenue and marketshare grew considerably. Subsequent versions of DB2 have continued to leverage the UCD approach and build on the momentum generated by Version 5.

This article examines the UCD program that started with the early stages of development for DB2 Version 5 in late 1995, and follows this program through the Version 5 release, the major follow-on releases (Versions 6 and 7), and into the current-day activities for upcoming releases. We examine how this program started, how it evolved over 6 years, and the key factors that contributed to the success of DB2 in the marketplace.

Although there have been other published articles related to UCD in practice (e.g., several of the articles introduced by Dykstra-Erickson, 2000, and Dykstra-Erickson, 2001), this case study has a unique combination of attributes:

- Application domain. DB2 is a complex middleware-product that is used worldwide by thousands of users.
- Program duration. This case study describes activities and results across 6 years and multiple versions of the product. In much the same manner that our user interface (UI) designs have been iteratively tested and improved, our UCD approach has also been iteratively improved during this time frame.
- Organizational scale. This UCD program has involved dozens of practitioners from multiple disciplines, IBM development sites, and countries, as well as thousands of hours of user involvement in the development process.
- Measures of success. Human–computer interaction (HCI) studies often focus on user-based metrics such as task time, user satisfaction, and error rates. Although these types of data were collected throughout our UCD program, we describe only the overall results of our design validation tests. Our focus is on measures of product success such as marketshare, revenue growth, customer satisfaction, and trade press reviews. Although these are, to varying degrees, indirect measures of ease of

use and the success of our UCD program, we find that they are among the most important factors for our stakeholders.

Sections 2 and 3 of this case study provide background information for readers not familiar with the relational database technology, marketplace, audience, or user tasks. There is also a brief overview of the DB2 graphical UI. Sections 4, 5, and 6 describe the initial application of UCD to DB2 in terms of motivation, ease-of-use objectives, UCD activities, and UI architecture and design. Section 7 describes how our UCD approach has evolved over subsequent releases of DB2, and Section 8 provides examples of the results of our UCD program in terms of business results. Last, Sections 9 and 10 describe the factors that have influenced the success of our UCD program and the organizational impact they have had beyond DB2.

2. DB2 PRODUCT DOMAIN

DB2 provides a rich technology offering for a wide range of complex, multifaceted environments. This section describes the DB2 product domain in more detail.

2.1. DB2 Product Domain: The Technology

RDBMS products such as DB2 belong to a class of software known as *middleware,* functioning between the operating system and end-user applications. An RDBMS stores persistent data in *relational databases,* which are collections of data that are logically stored and accessed using the relational data model, a tabular approach to conceptualizing data (described in Ullman, 1988). The data in an RDBMS are accessed by end-user applications, which benefit from various RDBMS services such as concurrency of access, security management, backup and recovery, and a common query language known as *structured query language* (SQL).

DB2 is used with a wide range of application types. It is often used in online transaction processing (OLTP) applications, which focus on the daily transactions of a business, as in the case of processing a credit card purchase in a retail store. OLTP applications typically involve a large volume of relatively simple database transactions. In contrast, DB2 is also used in data warehousing applications, which involve large stores of decision-making data that are used by analysts to, for example, evaluate sales trends. Data warehousing applications are characterized by elaborate, complex database queries. Each of these potential uses of DB2 demands a specialized set of features and characteristics.

A key attribute of DB2 is its cross-platform nature. DB2 runs on non-IBM computers, including UNIX servers from Sun and Hewlett-Packard, as well as on IBM computers such as the RISC System/6000®. In addition, DB2 runs on various operating systems, including IBM AIX®, Linux, Sun Solaris®, HP-UX, Windows NT®, and Windows 2000.

2.2. DB2 Product Domain: The Audience

The user audience of an RDBMS consists of information technology professionals who assume various roles. Two key roles are *database administration,* which focuses on designing, implementing, and maintaining databases; and *application development,* which focuses on developing application programs that access RDBMSs for storing data. Various other professionals use RDBMSs, including those who carry out the roles of system administration, system operation, and network administration. Note that end users of RDBMS applications typically do not interact with an RDBMS directly—end users interact with an application that, in turn, uses the services of the RDBMS to perform its functions.

The roles involved in using an RDBMS are often combined, depending on the size and complexity of the organization. For example, a small organization may assign the roles of database administration and system administration to one person, whereas large enterprises typically have many database administrators with different areas of specialization, such as storage management or data movement.

Information about the audience targeted for DB2 Version 5 is provided in Section 5.1. Audience Analysis, Task Analysis, and Scenario Modeling.

2.3. DB2 Product Domain: The Marketplace

DB2 is used around the world, is translated into over a dozen languages, and serves a broad range of industries, including the financial, retail, and telecommunications sectors. It is used in various organizations, from small startup companies to large enterprises. DB2 is often implemented as part of a "mission-critical" system in which high availability and consistently fast performance are essential to meeting the organization's business goals. A recent example of DB2 usage is the 2000 Sydney Olympics, where DB2 provided data on athlete rankings and other statistics for live Web site reports and television broadcasts.

The highly competitive RDBMS marketplace was estimated to be worth approximately $7 billion in 2000, and has experienced a 15% growth rate from 1999 to 2000 (Gartner Dataquest, 2001). This marketplace has evolved considerably during the period of time being discussed in this article. In 1995, the RDBMS marketplace included a wide range of RDBMS vendors, many of whom specialized in a niche area such as data warehousing. A consolidation occurred in the years since, and currently there are three major vendors that dominate most of the marketplace. IBM, with its DB2 product, is one of those vendors.

Ease of use is a compelling issue in the RDBMS marketplace. In a recent industry survey of 303 information technology professionals (Whiting, 2000a), the top four responses to the question "What are the biggest technology issues that your company faces in executing its database management strategy?" were (a) ease of administration (58%), (b) available and qualified administrators (58%), (c) compatibility with other key enterprise software (57%), and (d) available and qualified programmers (51%). (Multiple responses were allowed for this survey.) Therefore, an element of ease of use found its way into the top responses, both through the explicit reference to ease of administration as a top issue and through the role of ease of use in facilitating skills development for administrators and programmers.

3. THE DB2 PRODUCT: A USER'S VIEW

This section provides an overview of the user experience for DB2 and sets the stage for examining the UCD program in the sections that follow.

3.1. Tasks Performed Using DB2

As discussed previously, two of the key roles involved in using an RDBMS are database administration and application development. Some of the key task categories associated with these roles are listed later. This list is not exhaustive, but rather a short collection intended to provide a sampling of the types of activities involved in using DB2.

- *Object management* refers to creating, altering, and deleting database objects such as tables (collections of data logically arranged in rows and columns) or stored procedures (programs that are run on the database server rather than in an application program). A wide range of users, including database administrators and application developers, carry out object management tasks.
- *Backup and recovery* is used to guard against events such as disk failure or major catastrophic events such as fire, vandalism, or earthquakes. Backup involves creating a copy of the data in a database and storing it on a different medium. Recovery involves rebuilding the database, and typically involves using a backup copy. Backup and recovery tasks are performed by database administrators, often with operational support from system operators.
- *Testing an application program* involves testing the database access functionality of an application program. This is typically carried out by application developers in a test environment before an application program is deployed into a production environment.

When using DB2, tasks such as those described earlier can be carried out using one or more of the following methods:

- *Using a command line interface.* DB2 users can run SQL statements and DB2 utility functions interactively. For example, a DB2 user could interactively create a new relational table named CUSTOMER with columns named CUSTOMER_ID, FIRST_NAME, LAST_NAME, and COMPANY_NAME by entering the following statement:

```
CREATE TABLE CUSTOMER
(CUSTOMER_ID CHARACTER (10),
FIRST_NAME VARCHAR (40),
LAST_NAME VARCHAR (40),
COMPANY_NAME VARCHAR (50))
```

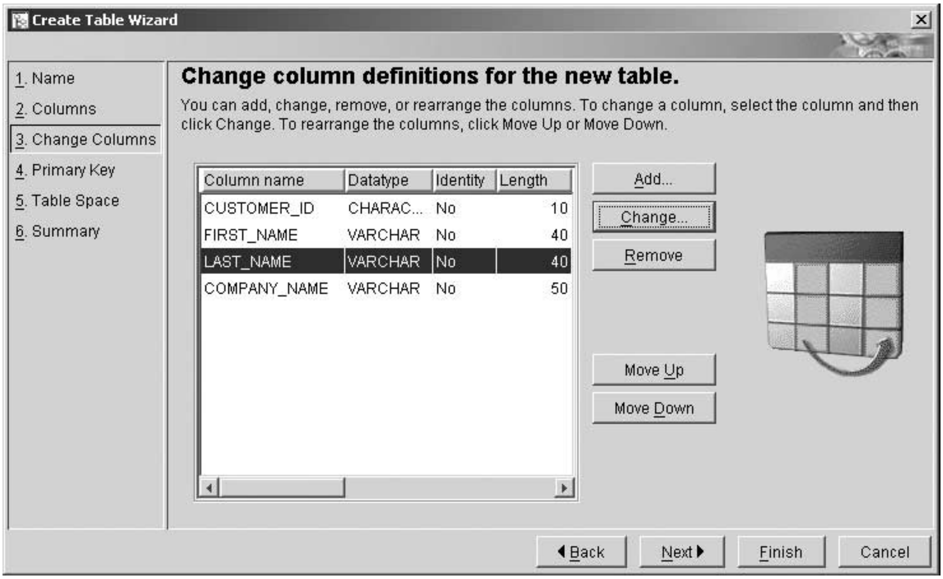
- *Using graphical tools.* With these tools, users can carry out database administration and application development tasks without using the command line. For example, the wizard in Figure 1 could be used to create the CUSTOMER table.
- *Using a programming interface.* Application developers can access DB2 from within their application programs through a wide range of programming interfaces including embedded SQL, Open Database Connectivity, and Java interfaces such as Java Database Connectivity.

The audience for our early UCD work relied heavily on graphical tools to administer the database. (The audience analysis that led to this conclusion is described in Section 5.1. Audience Analysis, Task Analysis, and Scenario Modeling.) The following sections describe the graphical tools and the design approach that was used to develop these tools.

3.2. Overview of the DB2 Graphical Tools

Much of the DB2 UCD program has focused on supporting user tasks through the graphical tools. The following list describes the core graphical tools that have been developed through the UCD process and the tasks they support.

- The *Control Center* is the central graphical tool for working with DB2 database objects. The Control Center lists DB2 objects and object types organized in a tree structure alongside a more detailed view of the selected item, as shown in Figure 2.



Create Table Wizard

1. Name
2. Columns
3. Change Columns
4. Primary Key
5. Table Space
6. Summary

Change column definitions for the new table.

You can add, change, remove, or rearrange the columns. To change a column, select the column and then click Change. To rearrange the columns, click Move Up or Move Down.

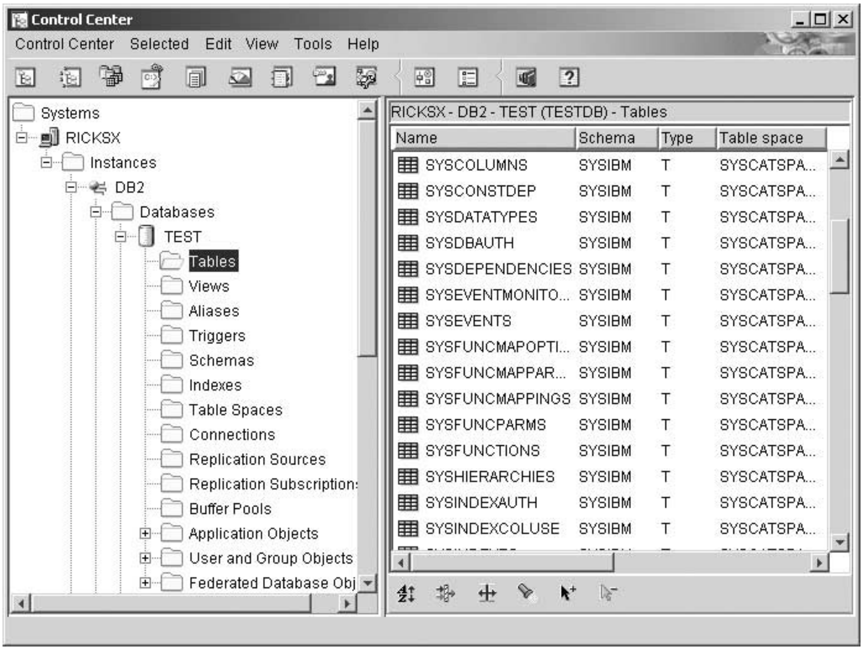
Column name	Datatype	Identity	Length
CUSTOMER_ID	CHARAC...	No	10
FIRST_NAME	VARCHAR	No	40
LAST_NAME	VARCHAR	No	40
COMPANY_NAME	VARCHAR	No	50

FIGURE 1 A DB2 wizard for creating a table. Each of the steps in the wizard is listed along the left-hand side, and the details of the selected step are displayed on the right-hand side.

Users can select object types (such as "Tables") from the tree view and invoke actions (such as "Create table") against these object types.

- The *Command Center* lets users interactively issue and refine DB2 commands, SQL statements, and operating system commands. It also includes support for evaluating the efficiency of SQL statements.
- The *Script Center* lets users create, run, and schedule DB2 command scripts, SQL statement scripts, and operating-system command scripts. In contrast to the Command Center, the Script Center is not intended for interactive execution of individual commands but rather for batch submission of commands in a script.
- The *Journal* is the central location for examining current status information as well as historical information about completed activities.
- The *Information Center* provides quick access to the DB2 product manuals, various sample programs, and other sources of DB2 information on the Web.
- The *Alert Center* notifies users of error and warning conditions. It is intended to be a central location for quickly viewing information that may warrant immediate action by the user.

The tools just described were originally released as part of DB2 Version 5, the first release that included an integrated set of graphical administration tools. (The UI design approach used for the original design of these tools in Version 5 is de-



RICKSX - DB2 - TEST (TESTDB) - Tables

Name	Schema	Type	Table space
SYSCOLUMNS	SYSIBM	T	SYSCATSPA...
SYSCONSTDEP	SYSIBM	T	SYSCATSPA...
SYSDATATYPES	SYSIBM	T	SYSCATSPA...
SYSDBAUTH	SYSIBM	T	SYSCATSPA...
SYSDEPENDENCIES	SYSIBM	T	SYSCATSPA...
SYSEVENTMONITO...	SYSIBM	T	SYSCATSPA...
SYSEVENTS	SYSIBM	T	SYSCATSPA...
SYSFUNCMAPOPTI...	SYSIBM	T	SYSCATSPA...
SYSFUNCMAPPAR...	SYSIBM	T	SYSCATSPA...
SYSFUNCMAPPINGS	SYSIBM	T	SYSCATSPA...
SYSFUNCPARMS	SYSIBM	T	SYSCATSPA...
SYSFUNCTIONS	SYSIBM	T	SYSCATSPA...
SYSHIERARCHIES	SYSIBM	T	SYSCATSPA...
SYSINDEXAUTH	SYSIBM	T	SYSCATSPA...
SYSINDEXCOLUSE	SYSIBM	T	SYSCATSPA...

FIGURE 2 The DB2 Control Center. The left-hand pane of the window contains a tree structure that represents the hierarchy of DB2 objects and object types. Selecting an object type in the tree structure results in all instances of that object type being displayed in the right-hand pane of the window. For example, selecting the Tables object type results in a list of tables being shown in the right-hand pane.

scribed in Section 6. UI ARCHITECTURE AND DESIGN.) Subsequent releases have seen the addition of specialized tools such as the Stored Procedure Builder (an edit–compile–debug environment for developing stored procedures) and the Data Warehouse Center (for defining data warehousing processes). Each of these specialized tools supports a task grouping that maps to a specialized user role. For example, the Stored Procedure Builder supports application development tasks, specifically those related to developing stored procedures.

3.3. Design Approach for the DB2 Graphical Tools

The high-level design of the graphical tools provides users with the Control Center, the Command Center, and the other core tools to support a wide range of commonly performed tasks that span various user roles. The specialized tools, such as the Stored Procedure Builder and Data Warehouse Center, were each designed to support a more narrow set of tasks and roles. This approach of providing a combination of general-purpose core tools alongside specialized tools was intended to strike a balance between two extremes:

- The all-in-one approach. This approach would provide a single general-purpose tool packed with support for all tasks for all user roles. The drawback of this approach is the complexity of the resulting tool.
- The unbundled approach. This approach would provide a large number of separate tools, each supporting a specific task or very small grouping of tasks (e.g., one tool for creating tables and another tool for backing up tables). One of the key drawbacks to this approach is the high burden of user navigation between all these tools.

Achieving a balance between these two extremes was an integral part of the UI architecture effort for DB2, described in Section 6. UI ARCHITECTURE AND DESIGN.

4. SETTING THE DIRECTION FOR DB2 VERSION 5

Given the background provided by the preceding sections, we now describe the motivation, objectives, and scope for the DB2 Version 5 UCD project. This project was the first application of UCD to DB2.

4.1. Motivation for Improving Ease of Use in DB2

In 1995, prior to the start of the DB2 Version 5 project, DB2 was a relative newcomer to the workstation RDBMS market. It had quickly established a reputation for functionality, reliability, and robust performance, all of which were important foundations for an RDBMS product. However, it had not yet established a reputation for ease of use due to, among other factors, the lack of an integrated set of graphical tools. One trade press review of DB2 Version 2[2] for the Microsoft Windows NT operating system was titled "IBM's DB2 for Windows NT Looks Powerful, Not Pretty," a poignant reference to the product's ease-of-use shortcomings at the time (Taschek, 1995). Another trade press review of DB2 (Dyck, 1997) described the DB2 Version 2 UI in unflattering terms.

Two IBM products that existed prior to DB2 Version 5 were IBM DataHub® and IBM DataPropagator® Relational. Customers frequently used these products in conjunction with DB2. DataHub, a graphical database administration tool, provided the ability for users to perform many tasks without using a command line or SQL statements. DataPropagator Relational replicated changes in users' data, often between a central host database and client–server databases. Although all three products were often used together by the same database administrator, there was little integration between them despite the fact that they were all developed by the same company. Their installation, configuration, terminology, and UIs were not well integrated.

DB2, DataHub, and DataPropagator Relational were enjoying success in the marketplace as individual products. However, product management realized that to continue this success IBM would have to start delivering integrated solutions rather than separate products. Therefore, the decision to integrate these three products was made. A cross-product, multidisciplinary "UI architecture" team with representatives from all three products' software development and human factors teams was brought together in late October 1995 to examine the integration alternatives.

The first attempt was an "on-the-glass" integration—all three products' existing UIs and code bases would be maintained, but would be launched from the DataHub UI. However, this approach was quickly rejected after being prototyped. This shallow integration only highlighted the inconsistencies between the products and would not be competitive. Instead, in January 1996, the UI architecture team and product management decided to integrate the products tightly and to design a completely new set of integrated graphical tools. In addition, a commitment was made to use the IBM UCD process to develop this integrated solution.

4.2. UCD Objectives

The IBM UCD process begins with setting the project objectives. We believe that one of the key success factors in the UCD effort for DB2 Version 5 was clear agreement, articulation, and documentation of these objectives. After the objectives were discussed, agreed on, and articulated to the broader development community, we found they provided a clear sense of direction throughout the project and often helped to resolve design issues. The objectives were defined using the following dimensions:

[2] DB2 Version 2 was the major release that immediately preceded DB2 Version 5.

- *Operating system.* Microsoft Windows NT was selected as the key operating system for our UCD effort. This choice was motivated in large part by the high growth rate of the Windows NT RDBMS marketplace during the Version 5 time frame. This high growth rate presented opportunities for establishing new business for DB2. Furthermore, Windows NT RDBMS users were reputed to have higher expectations for ease of use than RDBMS users of other operating systems. In particular, ease of use of graphical tools was thought to be particularly important for users of Windows NT. Although we did not carry out formal studies to validate the high ease-of-use expectations of Windows NT RDBMS users, informal observations throughout our UCD activities for Version 5 convinced us that this reputation was well founded. Consequently, we felt that focusing on Windows NT provided us with the most challenging (and, consequently, most fruitful) target for focusing our efforts.
- *Target market.* We coupled our focus on the Windows NT operating system with a focus on small- and medium-sized RDBMS organizations, where Windows NT dominated the market. These organizations rely on small- and medium-sized RDBMS value-added resellers[3] (VARs) to provide packaged or customized RDBMS applications along with ongoing services. As a result, small- and medium-sized Windows NT VARs became our primary market focus.
- *Strategic intent.* It is important to note that the strategic intent behind targeting the small- and medium-sized RDBMS Windows NT market was to set a high ease-of-use standard for DB2's graphical tools. Our marketing and sales efforts continued to be focused on many other market segments, particularly large enterprises. In fact, we identified the enterprise RDBMS marketplace as our secondary target market, and included some users from enterprise organizations in our UCD activities. However, the small- and medium-sized RDBMS customers were deemed to be the most demanding from a graphical tools ease-of-use perspective and thus were the best choice for achieving our UCD objectives in DB2 Version 5.
- *Competition.* For our primary market (small- and medium-sized RDBMS Windows NT VARs), we identified a key competitor (hereafter referred to as Competitor X). This competitor held both a leading marketshare and was the generally acknowledged ease-of-use leader at the time. For our secondary target market (large enterprises), we identified a different competitor (hereafter referred to as Competitor Y). This competitor had leading marketshare in this secondary target market.
- *Ease-of-use objectives.* The primary ease-of-use objective was to equal or exceed Competitor X, as measured by overall user satisfaction[4] in head-to-head design validation tests for a core set of RDBMS tasks. During the course of the project, a secondary objective emerged on the topic of "channel attractiveness." This objective was in response to what we learned about the purchase decision priorities of VAR decision

[3] A value-added reseller (VAR) is typically a company that takes a commercially available product, adds its own "value" (typically in the form of an application for the product), and then resells it as a packaged solution. For example, a VAR may develop and market an application for managing medical records along with the database product that is used by the application.

[4] Overall user satisfaction (our working ease-of-use indicator) was measured by questionnaires administered for each design validation participant at the end of each test session. In addition to overall satisfaction, we gathered user satisfaction data on various components of overall satisfaction including ease of learning, feedback, terminology, and documentation.

makers. (These priorities are described in Section 5.1. Audience Analysis, Task Analysis, and Scenario Modeling.) Furthermore, we devoted some effort to ensuring we achieved higher ease of use than Competitor Y for large enterprises.

4.3. Project Scope

Based on the direction spelled out by the UCD objectives, we deployed the UCD process (as described in Vredenburg et al., 2002) in its entirety. The scope of the project is illustrated by the following characteristics:

- Over 1000 hr of user feedback was gathered in laboratory sessions.
- Staff from the following disciplines were directly involved: software development (i.e., programmers), user feedback, interaction design, information development (i.e., writers of user assistance such as online help and user guides), visual design, marketing, and service. At the peak of the project, for every user feedback specialist or interaction designer there were approximately four software developers assigned to work on the UCD focus areas.
- Staff were located across two sites: San Jose, California; and Toronto, Ontario, Canada.
- The time frame of the project spanned almost 2 years, with the key activities happening in the period of January 1996 to June 1997.

The project was carried out under the typical time constraints, competitive pressures, and business realities of a large-scale software-product development project.

5. UCD USER FEEDBACK ACTIVITIES FOR DB2 VERSION 5

This section provides a brief description of the DB2 Version 5 UCD activities, with a focus on describing significant innovations to the standard UCD methodology.

5.1. Audience Analysis, Task Analysis, and Scenario Modeling

The objectives described in Section 4.2. UCD Objectives provided the basis for starting our UCD user feedback activities. Our first step was to focus on understanding the target audience, their tasks, and scenarios.[5]

We started by conducting an audience analysis of small- and medium-sized Windows NT VARs. This consisted of exploratory research such as reviewing VAR trade press articles, interviewing IBM staff familiar with this marketplace, and conducting site visits to VAR workplaces to interview staff on the topic of their roles, responsibilities, and characteristics. As a result of this preliminary research we identified three types of users to target for our UCD activities: (a) VAR decision

[5] By *scenario* we refer to the task flow (i.e., scenarios were used to described how component tasks were sequenced).

makers with a broad view of the business, products, and services of their organizations; (b) RDBMS-oriented application developers who work for VARs; and (c) database administrators who work for VARs. We also came to better understand the characteristics of our audience, including the high importance of using graphical tools for working with databases.

Subsequently, three task analysis sessions were held, one for each of the targeted user types (VAR decision makers, application developers, and database administrators). Each of these sessions was held in a usability laboratory, spanned a full 8-hr day, and involved a group of 15 to 20 users. We used a LAN-based (local area network) groupware tool that helps conduct such task analysis sessions very efficiently (as described by Boshes, 1995, and by Vredenburg et al., 2002). The typical agenda for such sessions includes (a) a collection of tasks performed by users, (b) a prioritization of tasks (through ratings of importance), (c) identification of task categories, and (d) grouping of tasks into categories. Following the task analysis, a series of scenario modeling sessions would typically be held to organize the tasks into scenarios and to identify the goals, inputs, outputs, and bottlenecks of the scenarios.

The VAR decision-maker task analysis led to a better understanding of the criteria used by VARs to establish their intent to purchase. Specifically, we learned that VARs base much of their purchase decision on nonproduct issues such as education, service, marketing, and sales support. (Such issues are often classified by VARs as "channel-attractiveness" issues.) This understanding of VARs led to the formulation of a secondary UCD objective that focused on channel attractiveness. Further details of how we addressed this objective are given in Section 5.4. Design Walkthrough.

The task analysis sessions for the application developers and database administrators differed from the standard LAN-based groupware task analysis methodology practiced by IBM at that time. These task analysis sessions were integrated with scenario modeling into a 1-day session rather than being held in separate sessions. The following factors helped achieve this integration of sessions:

- Customization of the LAN-based groupware tool. This customization enabled the output of the task analysis to be automatically generated (during a lunch break) as input information for the scenario modeling activity.
- Dividing the users into small groups (three or four users per group) and carrying out numerous scenario modeling sessions in parallel using these small groups.

This approach to combining task analysis and scenario modeling sessions helped us produce results faster than if these sessions were held separately. We strived to generate task and scenario data as soon as possible because these data were required to carry out various other UCD activities, and the sooner these UCD activities were completed then the more of an impact they would have on the product.

One of the key outputs of the task analysis sessions was identification of the most important tasks for our target audience. In all, 15 database tasks were identified as our top priority tasks, and they formed the basis for subsequent UCD activities. These tasks included installing the client and server components, configuring a client, creating a table, creating a database, backing up and recovering a database, optimizing a query, and configuring the performance of the database server.

5.2. Competitor Evaluation

Two expert-user evaluations of Competitor X were held, one for database administration tasks (a 5-day session) and the other for application development tasks (a 3-day session). The unique aspect of these evaluations was that they were conducted using LAN-based groupware that allowed members of various design disciplines to collaboratively enter observations and product attribute ratings as the expert user stepped through tasks. In addition, each day of these sessions had 2 hr set aside for members of one design discipline (e.g., visual design or information development) to explore discipline-specific competitive product attributes. We believe that involving a broad range of design disciplines in this fashion helped optimize the influence of this activity on design.

5.3. Iterative Evaluation

Various iterative evaluation sessions were held, ranging from low-fidelity and early-prototype explorations to high-fidelity, task-based evaluations. These sessions occurred over an 8-month period and, at their peak, were being conducted on a weekly basis. The Toronto and San Jose locations were linked together digitally for many of these sessions using remote collaboration software tools that, for example, enabled a designer in San Jose to interact with the same prototype being evaluated by a user in Toronto.

5.4. Design Walkthrough

The UCD design walkthrough methodology is intended to compare the overall IBM solution to that of the key competitor and to assess, among other factors, the intent to purchase. Given the focus of our VAR target market on nonproduct issues in the purchase decision (as described in Section 5.1. Audience Analysis, Task Analysis, and Scenario Modeling), we decided against holding a typical task-based walkthrough of the product interfaces. Instead, we held a design walkthrough that focused on channel-attractiveness issues such as business proposition, product proposition, terms and conditions, education, technical support, marketing activities, and sales support. This walkthrough was carried out using LAN-based groupware and involved a head-to-head comparison of the DB2 offering versus that of Competitor X.

5.5. Design Validation

We carried out an extensive set of design validation tests on beta-level code drivers, comparing DB2 against Competitor X for the top database administration tasks. These end-of-cycle tests proved to be valuable for uncovering ease-of-use issues that did not surface during the earlier phases of the project, and in evaluating our success in achieving the DB2 Version 5 ease-of-use objectives.

The design validation was carried out in two rounds of testing:

1. The first round involved user testing of DB2 and Competitor X. This round of testing uncovered various user issues that were then translated into changes to the product (also known as *fixes*). These tests were carried out after beta-level code drivers became available, but while we still had opportunities to implement late changes to the product prior to its release.
2. The second round involved user testing of DB2 only. The purpose of this round was to evaluate the efficacy of the fixes and, therefore, this round of testing was carried out after the fixes from the first round were implemented. We used this round to determine our *fix rate,* which is the proportion of user problems that were genuinely fixed, as determined by objective user data. The second round occurred while the product was undergoing its final system testing and was no longer open to UI changes.

An important aspect of the design validations was an aggressive fix rate target, which involved fixing 100% of Severity 1 problems (users could not complete the task), 100% of Severity 2 problems (users had major problems completing the task), and 90% of Severity 3 problems (users had minor problems). Some lower severity problems were not completely fixed in Version 5 and were, instead, addressed in following versions of the product. However, the vast majority of user problems identified during design validation were fixed before the product was released. Achieving this fix rate prior to product release involved very close collaboration across disciplines in resolving user problems.

As described earlier in Section 4.2. UCD objectives for DB2 Version 5, the primary ease-of-use objective was to equal or exceed Competitor X in overall user satisfaction. The design validation results indicated that DB2 scored statistically significantly higher than Competitor X in ratings for overall user satisfaction, and thus the primary ease-of-use objective was achieved. In addition, the results showed that DB2 equaled or exceeded Competitor X in task-specific user satisfaction ratings[6] for over 85% of the tasks covered by the design validations. These metrics provided a valuable yardstick by which to measure our success before we shipped the product.

5.6. Beta Feedback

A beta test of DB2 Version 5 was conducted to obtain early feedback on the product as it was used in customer environments. Feedback was gathered using Web-based surveys of beta users. In addition, beta users had the option of turning on an instrumentation feature that recorded user interactions. Several users selected this option and then sent us their instrumentation logs. These logs were used to evaluate usage

[6] A task-specific user satisfaction rating is a measure of users' satisfaction with how well the product (i.e., DB2 or Competitor X) supports a specific task. This is in contrast to overall user satisfaction, which measures users' satisfaction across all tasks.

patterns of the product and identify potential usability issues. (This approach to using instrumentation logs is discussed in Swallow, Hameluck, & Carey, 1997). These vehicles for providing beta feedback were used to gather user satisfaction data and user problem reports, and to understand how the various product components were being used.

6. UI ARCHITECTURE AND DESIGN

All of the UCD activities described in the previous section—with the exception of the design walkthrough, which focused on channel attractiveness—were carried out in support of the UI design of the graphical tools. This section explores the key aspects of the UI design approach for DB2 Version 5 and how the design work was tightly integrated with the UCD user feedback activities.

6.1. UI Architecture Team

The UI architecture team was composed of four interaction designers and three software developers from the major component teams: DB2, DataHub, and DataPropagator Relational. The goal of this cross-site, multidisciplinary team was to develop and maintain a long-term, cross-release vision for the DB2 graphical tools and to oversee the design of individual UI components for Version 5. The UI architecture team worked very closely together throughout the various phases of the project to produce a UI design that would meet the ease-of-use objectives while accounting for implementation considerations.

6.2. UI Architecture Approach

The UCD user feedback activities provided the UI architecture team with an understanding of user tasks, key scenarios, and the usability of the competition. The team used this knowledge to develop a holistic UI architecture that formed the foundation for the current and subsequent versions of the DB2 graphical tools. This architecture was then used as a basis for the detailed design phase of the project by defining, at a high level, the major UI components, task flow, integration of components, and design guidelines.

The architecture provided a long-term conceptual view of the UI. Its scope went beyond what was possible to implement in the current release, thus building extensibility into the UI design. A prototype was used to illustrate the architecture and to gather user feedback during the early iterative evaluation sessions.

Given a time-frame of 6 months for design activities and an estimate that the UI was destined to contain over 200 windows, it was necessary to assign the detailed design of many components to project members outside the architecture team. Design guidelines and the prototype were used together to educate component designers on where and how their components would fit into the overall UI structure, and to guide their detailed component designs. For example, for the designer responsible for the table component, the architecture spelled out that the "delete table" function was to be invoked from a menu item in the pop-up menu for tables listed in the right-hand pane of the Control Center. The architecture would also inform the developer that a confirmation message must be displayed when the delete table function is invoked by the user. This approach gave project members a holistic understanding of the UI even though they may have only been designing a small part. It also enforced consistency of interaction style across components.

6.3. Documented Component Design Specifications

The detailed component design specifications were formally documented, reviewed, and approved. The design specifications of each major window or view included the following:

- A description of the purpose of the window.
- A listing of menus and their actions (e.g., the table component was to have a menu with the rename, copy, and delete actions).
- Details of drag-and-drop support (e.g., it was specified whether a table could be deleted by dragging its icon and dropping it onto the trash bin).
- Visual layouts of the windows including text, graphics, and interaction mechanisms such as push buttons, list boxes, checkboxes, and radio buttons.

An interaction designer, software developer, information developer, and visual designer were identified to work on the design for each of the components. During this phase of detailed design, errors and omissions in the UI architecture were corrected and communicated to component designers. On approval of a design specification, coding (by software developers) would begin.

6.4. Formal Design Change Phase

After the design specification was approved, no changes were allowed without having a formal design change request (DCR) reviewed and approved by the UI architecture team, information development, and project management across both sites. Furthermore, differences between code implementation and the design specification were flagged as defects by the test team and by the UI architecture team. In fact, the UI architecture team carried out regular reviews of the implemented UI, thus maintaining the integrity of the design. The DCR process was also part of the design validation stage (described earlier) in that user problems that were generated during head-to-head tests against Competitor X were opened as DCRs. These design validation DCRs were given top priority at the late stages of the project as development resources were allocated.

6.5. UI Architecture and Design Process Flow

These aspects of our UI architecture and design approach are captured in Figure 3, which illustrates how the key user feedback activities that influenced design were integrated with UI design work.

The software development process for Version 5 followed a process that paralleled the UI architecture and design activities. Specifically, the software internals architecture (i.e., definition of how the software worked internally) happened in parallel to “1. Architecture” in Figure 3; software implementation (i.e., code development) happened in parallel to “2. Detailed design and implementation”; and software testing happened in parallel to “3. Design change.”

7. BEYOND VERSION 5: EVOLUTION OF THE DB2 UCD PROGRAM

UCD has continued to be applied to the development of DB2 in the releases since Version 5, and these releases have seen considerable ease of use improvements in their respective objective areas. This section describes how UCD evolved during the post Version 5 releases, and Section 8. DID UCD MAKE A DIFFERENCE? describes the external validation received for the DB2 UCD program starting with Version 5 and continuing into the subsequent releases.

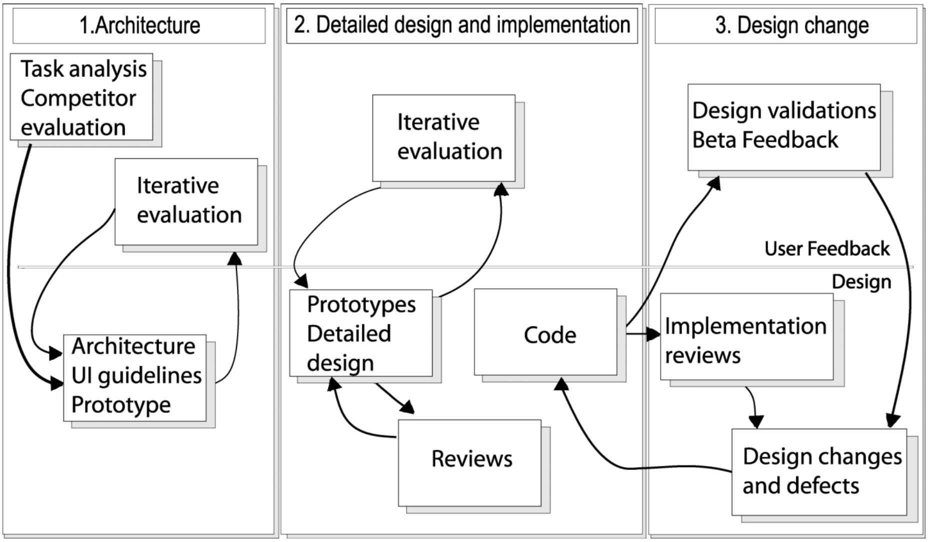

1.Architecture
2. Detailed design and implementation
3. Design change
Task analysis Competitor evaluation
Iterative evaluation
Architecture UI guidelines Prototype
Iterative evaluation
Prototypes Detailed design
Code
Reviews
Design validations Beta Feedback
User Feedback
Design
Implementation reviews
Design changes and defects

FIGURE 3 Integration of user feedback activities with UI design work for DB2 Version 5. Rectangles in the upper part of the figure (e.g., the rectangles labeled “Iterative evaluation”) represent user feedback activities, and rectangles in the lower part of the figure (e.g., the rectangle labeled “Code”) represent UI design activities and products of the design process. The directional lines between rectangles represent information flow.

7.1. Evolution of Ease-of-Use Objectives

One of the central elements of IBM's UCD approach is the focus on ease-of-use objectives for each release. Rather than trying to make incremental improvements across the entire range of user tasks or only targeting the most problematic tasks, specific focus areas are identified in each release and these areas receive a greater proportion of UCD resources than others. In Version 5, we focused on the top 15 database administration tasks and on channel attractiveness in small- and medium-sized RDBMS businesses that use the Windows NT operating system. Although we did not ignore aspects of the user experience outside of these tasks and target market, we devoted significantly more resources toward these focus areas to ensure they were usable and competitive.

Subsequent releases of DB2 have had different ease-of-use objectives and target markets based on the release content and business objectives. We have modified our target market and platforms, our user audience and tasks, and even the roles and responsibilities of the UCD practitioners.

Version 6: DB2 for OS/390 connectivity and common administration. The IBM DB2 for OS/390 product is a mainframe RDBMS that serves as a centralized data store for many large enterprises. In these large enterprises, it is common to have workstation databases (such as DB2 for Windows) that contain a subset of enterprise data. Data movement and replication technology ensures that changes to enterprise data are reflected in the workstation databases.

With the integration of replication functionality into DB2 Version 5, it was necessary to represent DB2 for OS/390 database objects (such as tables and databases) in the Control Center for the purpose of setting up data replication. However, other common DB2 for OS/390 actions (such as creating, altering, and deleting objects) were not available from this same UI. It was a natural progression to add support for DB2 for OS/390 administration tasks to the Control Center so that the DB2 family of database products could be administered across a variety of operating systems from one UI. Therefore, the ease-of-use objective for DB2 Version 6 was focused on providing easier connectivity and common administration across the mainframe and workstation environments. This objective was operationally defined in a similar fashion to the objective for Version 5—to equal or exceed a baseline solution as measured by user satisfaction ratings in design validation tests.

Mainframe database administrators tend to be different than the database administrators targeted in Version 5. They are comfortable with command-line UIs and work with very large numbers of database objects (several hundred tables in a UI view is not unusual). This change in audience and target market was reflected in the UCD activities we undertook in Version 6.

Version 7: "Out-of-box" experience for data warehouse administrators. A data warehouse is a specialized database used primarily to support analysis and decision making (Mattison, 1996). Analysts use data warehouse analytical tools to solve problems in application areas such as assessing credit risk and identifying sales trends.

In DB2 Version 7, the data warehousing capabilities of IBM's Visual Warehouse product and some related analytical tools were integrated into DB2. Visual Warehouse enjoyed success as a stand-alone product. However difficulties in installation, configuration, and initial usage caused it to be eliminated from many of the 3-day evaluation periods common in its target market. This 3-day evaluation period typically starts with opening the software package and continues through to creating successful analyses and reports on sample data. Tasks during this evaluation period include installation and configuration, familiarization with the product's capabilities and UI, creating a simple data warehouse from sample data, and loading data into an analytical application. Collectively, these tasks are known as the out-of-box experience.

The ease-of-use objective for Version 7 was focused on a successful out-of-box experience for administrators and analysts. More specifically, our objective was to equal or exceed the prime competitor as measured by user satisfaction ratings in design validation tests for the out-of-box experience. Our target market was line-of-business organizations that run data warehouse systems.

7.2. Evolution of Roles of HCI practitioners

As the DB2 product and its ease-of-use objectives have evolved over time, so has the role of the user feedback specialists and interaction designers. In Version 5, there was a difference in how these roles were staffed between the San Jose and Toronto sites. In Toronto, the roles of user feedback specialists and interaction designers were separated into different positions within the same human factors department. Although there was considerable interaction and overlap of responsibilities, the primary responsibilities were differentiated. On the other hand, human factors staff at the San Jose site assumed the role of both user feedback specialist and interaction designer.

There are many inherent advantages to the specialization of HCI skills, including more opportunity to develop an area of expertise and greater objectivity in evaluating designs. However, two key factors led to the integration of design and evaluation responsibilities for the human factors staff across both sites.

First, we found that peak workloads varied for the two disciplines over the course of a release. For example, most formal user feedback activity occurs either very early (as in the case of task analysis) or late in the development cycle (as in the case of design validations). In contrast, most interaction design activity is focused in the early and middle parts of the development cycle. Therefore, integration of design and evaluation responsibilities enabled us to staff peak workload periods more effectively by drawing on a larger pool of available and skilled staff.

Second, we found that integrating design and evaluation responsibilities helped develop valuable, complementary skills in human factors staff. This cross training resulted in more effective usability evaluations and recommendations because the evaluators were more familiar with the designs and tasks.

We have also seen an evolution of ownership of the UI design specifications over time. In Version 5 of DB2, human factors staff were responsible for the high-level architecture and design and for a small percentage of the low-level, detailed design specification. In addition, they owned the UI design guidelines and reviewed all detailed design specifications. This arrangement was largely a result of the resource that was available at the time and the large number of new UIs that needed to be designed. In Versions 6 and 7, as the number of interaction designers increased (through additional resources and the cross training of user feedback specialists), human factors staff assumed more ownership of low-level UI design specifications. As a result, designs began to demonstrate greater consistency and adherence to guidelines, and early user feedback on designs increased.

During Versions 6 and 7 we have seen human factors staff take on the role of facilitators and instructors of UI design and usability evaluation for the larger, multidisciplinary development community. Specifically, we have held internal education classes on the UI design process, design guidelines, common design mistakes, and iterative evaluations. These classes were attended by programmers, information developers, and visual designers. The goal of this education was not to displace the existing human factors staff, but rather to enhance the UCD skills of the entire development community, allowing them to be more effective participants in design and evaluation activities.

The size of the development resources for DB2 grew during the course of Versions 6 and 7. The human factors staffing increased in direct proportion to the overall development resource increase, thus maintaining the ratio of human factors staff to software developers established in Version 5.

7.3. The Role of High-Level Scenarios

In each subsequent version of DB2, we have seen an increased use of high-level scenarios to aid in UI design and usability evaluations. These high-level scenarios are defined through descriptions of

1. Target users, described in a fashion similar to that of Cooper's (1999) personas.
2. The end state (goal) that the target users are trying to achieve.
3. The product-independent task flow for achieving the end state, described at the level of user intentions (similar to essential use cases as described by Constantine & Lockwood, 1999).
4. The environment, which includes the hardware, software, and network configuration of the usage context, as well as the organizational culture in which the target users work.

These high-level scenarios differ from the scenarios described earlier in this article (e.g., in Section 5.1. Audience Description, Task Analysis, and Scenario Modeling) primarily in scope and depth of definition. The high-level scenarios cover a broader scope, such as the out-of-box experience covered in DB2 Version 7 (in contrast to the more narrowly focused scenario of installation and configuration in DB2 Version 5).

In addition, the high-level scenarios are defined through a rich set of descriptions of target users, end states, task flows, and environments (in contrast to DB2 Version 5 scenarios, which consisted of more narrowly defined task flows).

We have used high-level scenarios to ensure that we are addressing the larger business problems of our users rather than strictly focusing on user performance and satisfaction with specific UI components. By looking at the users' goals and complete task flow at a high level, we can understand how the component UI designs support the users' goals and how new function integrates with existing UIs. We can also identify problem areas that require additional focus or resources. These high-level scenarios define the focus of our design validation tests and help to ensure we are addressing all aspects of the user experience.

8. DID UCD MAKE A DIFFERENCE?

This section describes the external validation received for DB2 and for the UCD program. By external validation, we mean validation that comes from beyond the confines of UCD metrics such as overall user satisfaction ratings, task-specific user satisfaction ratings, fix rates, and other such data gathered as part of the UCD process. Instead, we describe broader based validation gathered from other sources such as the trade press and marketplace sales. Although these other sources may be less objective than UCD data, they reflect the reality in which industrial UCD programs must succeed to prove their worth.

8.1. Critical Success

Soon after the beta version of DB2 Version 5 became available, the product started to receive critical acclaim for its ease of use, and this acclaim continues to this day. Some of the quotes from the trade press are listed in the following. These quotes are a stark contrast to the "powerful, not pretty" label applied to DB2 by *PC Week* (Taschek, 1995, p. 1) prior to Version 5:

- "Installation on both client and server is mind-numbingly easy" (Whitehorn, 1977).
- "Improvements in ease of use and management make it worthwhile all by itself" (Stearns, 1997, p. 35).
- " … a vastly easier client setup procedure, integrated replication and a fresh new interface that's right on target" (Dyck, 1997, p. 1).
- "IBM's DB2 continues to provide top-notch database technology that now is easier to use and program" (Dyck, 1999, p. 14).
- "The latest version of DB2 bets on its added ease of use and e-commerce friendliness to penetrate deeper into the UNIX and Windows markets" (Apicella, 2000, p. 85).
- "We found that the package offers companies a comprehensive, well-integrated, highly programmable database" (Dyck, 2000, p. 19).

In addition to positive press reviews for ease of use (and other product attributes), DB2 has received various industry citations, such as the "IT [information technology] Leaders' Choice" for Open Systems DBMS (Burden, 1998), the winner of the "Database Grudge Match" (Whiting, 2000b), and the VARBusiness "Annual Report Card" award in the database category in 1997 (Bort, 1997) and in 2000 (Kaplan, 2000).

Although the press reviews do not represent the results of extensive user testing, they do provide evidence of perceived improvement in ease of use by proxies of the user population. Furthermore, many of the press reviews can be linked to the UCD investment. For example, the 1997 reviews make reference to client and server installation and they also elaborate on the ease of use of the new graphical tools. Both of these areas were the focus of the Version 5 UCD effort. Although we cannot prove that these reviews are a direct result of the UCD effort, we feel the evidence linking the reviews with the UCD effort is, nevertheless, compelling.

8.2. Business Success

The ease-of-use improvements in DB2 have clearly captured the mindshare of key influencers in the database marketplace. However, has that mindshare translated into business results? We begin to answer that question by examining the overall DB2 business results since DB2 Version 5 was released in September 1997.

- *Revenue growth.* DB2 revenue growth on distributed platforms grew at impressive rates. For the UNIX marketplace, DB2 grew its revenue at year-over-year rates of 69% in 1999 and 61% in 2000. For the Windows NT marketplace, DB2 grew its revenue at year-over-year rates of 125% in 1999 and 63% in 2000 (Gartner Dataquest, 2001).
- *Marketshare growth.* DB2 marketshare on distributed platforms also grew impressively. For the UNIX marketplace, DB2 grew its marketshare from 7.5% in 1998 to 10.5% in 1999 and 14.4% in 2000. For the Windows NT marketplace, DB2 grew its marketshare from 9.9% in 1998 to 15.2% in 1999 and 18.5% in 2000 (Gartner Dataquest, 2001).
- *Customer satisfaction.* IBM sponsors regular worldwide surveys of IBM customers (and the customers of our competition) to measure various aspects of customer satisfaction. These surveys have shown a significant improvement in DB2 usability ratings after the UCD program came into effect.

Thus, business results for DB2 have grown at an impressive rate since the UCD program was established. Although it is difficult to determine which portion of these business results are directly attributable to the UCD program, we feel it is reasonable to claim that the UCD program played a role in these improvements for the following reasons:

- The investment in the UCD program and its objectives formed a significant portion of investment of DB2 development resources during the period encompassing Versions 5 through 7.

- Ease of use is an important issue facing companies that use RDBMSs, as illustrated by Whiting (2000a). Therefore, we believe it is reasonable to assume that ease of use is an important factor in RDBMS purchase decisions.
- The investment in the UCD program has grown since the Version 5 release, a sign that decision makers in the DB2 organization have seen the value of this program through their information channels (such as discussions with customers and market research studies).

9. UCD SUCCESS FACTORS

In this section we identify a number of organizational, architectural, and practical factors that we believe have contributed to the ongoing success of the DB2 UCD program. We have also tried to provide insight into some of the problems we encountered and how they were addressed.

9.1. It Starts At The Top

Top-down product management commitment to UCD and ease-of-use objectives is paramount. DB2 product management has displayed a strong commitment to UCD throughout the course of the DB2 UCD program. At the start of Version 5, the Toronto DB2 development organization was reorganized in a way that concentrated most UI software developers into a single organization that was focused on the UCD process. Managers and developers from this organization played a major role in all user feedback activities and in the associated UI architecture and design efforts.

Resource allocations and product decisions consistently took UCD-related issues into account. A senior-level developer who reported directly to a senior software development manager was assigned as the full-time UCD project leader at the outset of the project, providing the vision and direction needed to get the cross-site, multidisciplinary team focused on the right things.

This top-down approach of leveraging UCD to drive decisions, resource allocations, and overall project direction is in contrast to a bottom-up approach, where people working “in the trenches” are expected to drive the UCD initiative. Our experience has shown that delegating this responsibility downward yields very limited results; product management needs to take on the responsibility for making UCD happen.

9.2. It Is Not Free

At the peak of Version 5, there was at least double the investment in development resources on the ease-of-use focus areas (graphical tools and installation) relative to previous releases. Tens of thousands of dollars were spent on recruiting users over the course of the project, and a significant amount of time was spent by all disciplines in attending user sessions, analyzing results, and incorporating user feedback into design. This investment in UCD has only increased over time. In other words, successful UCD does not come for free. A strong resource and financial commitment is required to supplement the leadership commitment.

Although UCD requires a significant investment, our experience is that the investment is more than justified. Key business factors such as improved customer satisfaction and accolades from the trade press have demonstrated a compelling return on investment for UCD.

9.3. Multisite Projects Can Work Just Fine

The two development labs involved throughout Versions 5 through 7 (in San Jose, California, and Toronto, Ontario) collaborated closely to produce an integrated product. Some of the factors that enabled our collaboration were the following:

- Full commitment to UCD from both sites.
- Working from the same UCD plan and objectives.
- Developing a UI architecture that provided a unifying vision for components across both sites.
- Creating a single design specification for the entire project.
- Collaborating closely on user feedback activities.
- Technological support for remote collaboration through the use of collaboration software for conducting usability evaluations and sharing information across sites.
- Authorizing extensive travel between the two sites.

At its best, we were able to leverage the multisite development effort to our advantage. For example, evening usability testing in Toronto could be observed by San Jose developers in the afternoon.

9.4. UCD Is Not Cast in Stone

As described earlier in this article, the standard UCD methodologies in Version 5 were adapted to better fit the needs of our project. This practice of adapting methodologies to meet our needs continued into Versions 6 and 7. For example, in Version 6 we conducted Web-based competitive evaluations; and in Version 7 we conducted design validations on site at customer locations. We did not stray from producing the types of user data required of the UCD process, nor did we stray from the key UCD principles. However, we took many liberties, without reservation, to ensure that our approach to gathering this data was adapted to our project environment. The result, we feel, was an effective deployment of UCD and a high degree of UCD buy-in from the development team.

9.5. It Pays to Plan Ahead

Many of the shortcomings of our initial UCD effort were centered around planning our work. We have planned and adapted our schedules to address these problems in subsequent releases.

In Version 5, much of the task analysis and competitor evaluation data was gathered after design work was well under way. Although the results of these sessions were used by designers, the information could have been used more effectively if we had gathered it earlier. Therefore, it is important to begin the UCD activities as early as possible, even before the project has a firm plan and direction. In subsequent versions of DB2, we have conducted more release-independent UCD activities. For example, we conduct competitive evaluations when a competitor ships a new release rather than wait until we reach the next product cycle.

Many of the features in the Version 5 design specification were eventually dropped from plan due to lack of resource. Therefore, our design did not match the reality of what could be implemented. In subsequent releases, we resized the development effort at key checkpoints throughout the development process, including the approval of the design specification, and then adjusted the design accordingly. Furthermore, the process for adding or dropping features and functions is now more controlled and requires multidisciplinary review.

9.6. A Holistic UI Architecture and Design

The UI architecture and design approach for DB2 Version 5 was an important factor in the overall ease-of-use success of the product. This approach recognizes UI architecture and design as distinct phases of the development process, as well as a distinct discipline that requires specialized skills. Fundamental to this approach is a holistic UI architecture and design. This approach is holistic in that the architecture provides the foundation for the current and subsequent versions of the lower level designs, as described in Section 6.2. UI Architecture Approach. As a result, all the components are designed from a “common blueprint,” resulting in (we believe) a more integrated and easy to use UI than would have been produced otherwise. This holistic approach was particularly important given the large number of people involved in the design of the UI.

9.7. Maintain the Specification … If At All Possible

At some point after the approval of the Version 5 design specification, we gave up on keeping it up to date with the approved design changes. This pragmatic decision was driven by lack of time and resource. However, this decision resulted in various communication challenges down the road that would likely not have occurred with an up-to-date specification.

In some cases, developers implementing components had to rely extensively on verbal communications and on reading through various design changes to understand what they were supposed to implement. Similarly, information developers were left without a definitive source for how the UI worked when it came time to write online help and other documentation. A maintained specification would address these and other issues, as well as leave behind a useful legacy for the following release.

9.8. Stay Focused on Things That Matter

In Version 5, we ended up with a UI that was rated higher than the key competitor for overall user satisfaction (as described in Section 5.5. Design Validation).

The press gave us flattering reviews (as described in Section 8.1. Critical Success). Also, customer satisfaction ratings showed significant usability improvements (as described in Section 8.2. Business Success). However, for the uninitiated, our UI may look rather unexciting, perhaps even plain. It does not have any major new UI widgets or a flashy look. In fact, it looks very much like a typical Windows application.

However, its strength lies in being optimized for supporting the tasks that users need to carry out on a day-to-day basis. We have common ways of scheduling jobs, capturing historical information, and filtering views. You can move across our UI components and to other Windows applications without entering into a new UI paradigm. Overall, we deliver an integrated UI that does the job quite well without many surprises, and that is exactly what our users want.

The key point here is to keep UCD projects focused on users and not on frills. We have found that the most effective approach to delivering a product that wins in the marketplace is to focus design on the UCD principles:

- Root design in an understanding of users and their tasks.
- Use multidisciplinary skills to design the user experience.
- Evaluate designs through user feedback.
- Focus on the competition.
- Leverage user feedback to drive plans and decisions.

This proven approach yields solid designs backed up by concrete user input rather than "cool" designs without substance.

10. ORGANIZATIONAL INFLUENCE OF THE UCD PROGRAM

This section describes the organizational influence of the DB2 UCD program and the influence it has had on other products within IBM.

10.1. Increased Awareness of Ease of Use and UCD

UCD and ease of use have become prominent topics in the IBM software development community. Executives from our CEO to our product managers have featured UCD and ease of use in their public addresses to employees. Site managers, products managers, and marketing managers have highlighted its importance. There has even been good-natured competition among product managers to determine who can most quickly install their product.

This top-down focus on ease of use has filtered its way down to the software development community. The number of developers taking UCD education classes has increased, the number of developers observing usability evaluations has increased, and we have seen an increase in participation in all UCD activities.

10.2. Concentration of UCD Investment

One of the lessons we learned from DB2 Version 5 was that sometimes a big improvement in ease of use requires a big investment. Although we apply the UCD process to all the products in the IBM Data Management Division, we have designated a few key ease-of-use projects where our UCD investment is more substantial. Rather than thinly spreading our resources in an attempt to cover all projects equally, we target dramatic results by concentrating UCD investment on these key projects.

10.3. Increased Focus on Integration

In DB2 Version 5, we integrated three separate products into one. In Version 6, we added connectivity and common administration for DB2 for OS/390. In Version 7, we integrated data warehousing capabilities. Even for IBM database products that do not have plans to integrate with DB2, we are striving for a common look and feel by using the DB2 UI style guidelines across our product portfolio. Also, as our users look for integrated solutions for their information technology problems, we will integrate DB2 with other solutions from IBM and its business partners.

11. THE FUTURE OF DB2 UCD

One of the exciting new frontiers for the DB2 UCD program is in the area of Self-Managing And Resource Tuning (SMART). This initiative is based on the realization that ease of use of graphical tools is only one part of the overall solution to the problem of increasingly complex RDBMS environments. Another (and important) part of the solution involves minimizing the need for user interaction for the most complex tasks, such as performance tuning.

SMART is focused on producing a database that "is designed to reduce the human intervention needed to run and maintain a database. The long-term goal for SMART is to provide DBAs [database administrators] the option not to get involved with the database" (Sullivan, 2001, p. 40). UCD is playing a role in identifying specific focus areas for SMART and in designing the associated user experience.

We expect that the DB2 UCD program will continue to evolve in many areas other than SMART. This is a reflection of the success we have found with applying the core UCD principles. With the historical perspective of over 6 years of process evolution and demonstrated success over multiple releases, DB2 remains firmly committed to its ongoing UCD program.

REFERENCES

Apicella, M. (2000). DB2's manageability–New release bolsters e-business capabilities with OLAP and XML support. *InfoWorld, 22*(17), 85.

Bort, J. (1997, October 15). IBM hits gold in database race, while Microsoft takes a tumble. *VARBusiness, 1318,* 233–235.

Boshes, M. (1995). User centered design task analysis. *Proceedings of the Sixth Annual GroupSystems Users' Conference,* 1–24.

Burden, K. (1998). IT leaders' choice [Electronic version]. *Computerworld, 32*. Retrieved January 13, 2002, from http://www.computerworld.com/cwi/story/0,1199,NAV47_STO4929,00.html

Constantine, L., & Lockwood, L. (1999). *Software for use.* New York: ACM Press.

Cooper, A. (1999). *The inmates are running the asylum.* Indianapolis, IN: SAMS.

Dyck, T. (1997). DB2 pulls out the big guns [Electronic version]. *PCWeek, 14*(11), 1–2. Retrieved January 13, 2002, from http://www.zdnet.com/eweek/news/0331/31db2.html

Dyck, T. (1999). IBM DB2 for the masses. *PC Week, 16*(21), 14.

Dyck, T. (2000). DB2 V7.1 covers the spectrum. *PC Week, 17*(16), 18–19.

Dykstra-Erickson, E. (2000). An insider's view of interface design. *Interactions, 7,* 8–86.

Dykstra-Erickson, E. (2001). Take two: An insider's view of interface design. *Interactions, 8,* 27–84.

Gartner Dataquest. (2001). *2000 Database Management Systems Software Market Share.* Stamford, CT: Author.

Kaplan, M. (2000). ARC databases: IBM beats out microsoft [Electronic version]. *VARBusiness, 1621,* 83. Retrieved January 13, 2002, from http://www.varbusiness.com/sections/technology/tech.asp?ArticleID=20731

Mattison, R. (1996). *Data warehousing: Strategies, technologies, and techniques.* New York: McGraw-Hill.

Stearns, T. (1997). DB2 challenges Oracle, Microsoft [Electronic version]. *LAN Times, 14*(8), 33–35.

Sullivan, T. (2001). DB2, 9i war looms large [Electronic version]. *InfoWorld, 23*(20), 40. Retrieved January 13, 2002, from http://www.infoworld.com/articles/hn/xml/01/05/14-/010514hndbwars.xml

Swallow, J., Hameluck, D., & Carey, T. (1997). User interface instrumentation for usability analysis: A case study. HCI & Telelearning Laboratories report, University of Waterloo, Canada. Retrieved January 13, 2002, from http://watserv1.uwaterloo.ca/~tcarey/casestudy.html

Taschek, J. (1995). IBM's DB2 for Windows NT looks powerful, not pretty. *PC Week, 13*(49), 1.

Ullman, J. D (1988). *Principles of database and knowledge-based systems* (Vol. 1). Rockville, MD: Computer Science Press, Inc.

Vredenburg, K., Isensee, S., & Righi, C. (2002). *User-centered design: An integrated approach.* Upper Saddle River, NJ: Prentice Hall.

Whitehorn, M. (1997, May–June). DB2 Universal Database v5.0 for NT. *Informations Week UK.*

Whiting, R. (2000a). Database administrators: A precious commodity. *InformationWeek, 54*. Retrieved January 13, 2002, from http://www.informationweek.com/815/databas3.htm

Whiting, R. (2000b). Database grudge match. *InformationWeek,* 45–56. Retrieved January 13, 2002, from http://www.informationweek.com/815/database.htm