APPENDIX

CD SELECTIONS

CHAPTER 1: INTRODUCTION TO SYSTEMS ANALYSIS AND DESIGN

Throughout this book, many new concepts about object-oriented systems analysis and design are introduced. As a way to make these new concepts more relevant, we apply them to a fictitious company called CD Selections. CD Selections is a chain of 50 music stores in California, with headquarters in Los Angeles. Annual sales last year were $50 million, and they have been growing at about 3 to 5 percent per year for the past few years. The firm has been interested in expanding their presence beyond California. Margaret Mooney, Vice President of Marketing, has become excited by and concerned about the rise of Internet sites selling CDs and sites such as iTunes that sell digital music. She believes that the Internet has great potential, but she wants to use it in the right way. Rushing into e-commerce without considering its effect on existing brick-and-mortar stores and the implications on existing systems at CD Selections could cause more harm than good.

Currently, CD Selections has a website that provides basic information about the company and about each of its stores (e.g., map, operating hours, phone number). The website was developed by an Internet consulting firm and is hosted by a prominent local Internet service provider (ISP) in Los Angeles. The IT department at CD Selections has become experienced with Internet technology as it has worked with the ISP to maintain the site; however, it still has a lot to learn when it comes to conducting business over the Web. Margaret is interested in investigating the possibility of creating an e-commerce site that will work with the current systems used by CD Selections. In future chapters, we revisit CD Selections to see how the concepts introduced in the individual chapters affect Margaret and the team developing a Web-based solution for CD Selections.

CHAPTER 2: PROJECT MANAGEMENT

In this chapter, we introduced how object-oriented systems development projects were managed. Specifically, we described how projects were identified and how the identification led to a system request. Next, we presented the three different types of feasibility analysis and how their results helped in selecting a project. After that, we reviewed a set of traditional project management tools that can be applied to planning and managing of an object-oriented systems development project and demonstrated employing use-case points as a method that can be used to estimate the effort it will take to develop an object-oriented system. We next discussed the use of evolutionary work breakdown structures and iterative workplans in conjunction with the Unified Process. We then covered the issues related to assigning the right people to the development team. Finally, we described topics associated with the environment and infrastructure management workflows of the Unified Process. In this installment of the CD Selections case, we see how Margaret and the development team work through all of these topics with regard to the Web-based solution that they hope to create.

Project Identification and System Request

At CD Selections, all potential IT projects are reviewed and approved by a project steering committee that meets quarterly. The committee has representatives from IT as well as from the major areas of the business. For Margaret, the first step was to prepare a system request for the committee. Using the system request template (see Figure 2-2) Margaret prepared a system request (see Figure 2-A). Of course, the sponsor is Margaret, and the business needs are to increase sales and to better service retail customers. Notice that the need does not focus on the technology, such as the need "to upgrade our Web page" The focus is on the business aspects: sales and customer service.

For now, the business requirements are described at a very high level of detail. In this case, Margaret's vision for the requirements includes the ability to help brick-and-mortar stores reach out to new customers. Specifically, customers should be able to search for products over the Internet, locate a retail store that contains the product, put a product on "hold" for later store pick-up, and order products that are not currently being stocked.

The business value describes how the requirements will affect the business. Margaret found identifying intangible business value to be fairly straightforward in this case. The Internet is a "hot" area, so she expects the Internet to improve customer recognition and satisfaction. Estimating tangible value is more difficult. She expects that Internet-ordering will increase sales in the retail stores, but by how much?

Margaret decided to have her marketing group do some market research to learn how many retail customers do not complete purchases because the store does not carry the item they are looking for. They learned that stores lose approximately 5% of total sales from "out-of-stocks and non-stocks." This number gave Margaret some idea of how much sales could increase from the existing customer base (i.e., about $50,000 per store), but it does not indicate how many new customers the system will generate.

Estimating how much revenue CD Selections should anticipate from new Internet customers was not simple. One approach was to use some of CD Selections' standard models for predicting sales of new stores. Retail stores average about $1 million in sales per year (after they have been open a year or two), depending upon location factors such as city population, average incomes, proximity to universities, and so on. Margaret estimated that adding the new Internet site would have similar effects of adding a new store. This would

System Request—Internet Order Project

Project sponsor: Margaret Mooney, Vice President of Marketing

Business Need: This project has been initiated to reach new Internet customers and to better serve existing customers using Internet sales support.

Business Requirements:

Using the Web, customers should be able to search for products and identify the brick-and-mortar stores that have them in stock. They should be able to put items on hold at a store location or place an order for items that are not carried or are not in stock. The functionality that the system should have is as follows:

- Search through the CD Selections inventory of products.
- Identify the retail stores that have the product in stock.
- Put a product on hold at a retail store and schedule a time to pick up the product.
- Place an order for products not currently in stock or not carried by CD Selections.
- Receive confirmation that an order can be placed and when the item will be in stock.

Business Value:

We expect that CD Selections will increase sales by reducing lost sales due to out-of-stock or non-stocked items and by reaching out to new customers through its Internet presence. We expect the improved services will reduce customer complaints, primarily because 50% of all customer complaints stem from out-of-stocks or non-stocked items. Also, CD Selections should benefit from improved customer satisfaction and increased brand recognition due to its Internet presence.

Conservative estimates of tangible value to the company include:

- $750,000 (75% of $1,000,000) in sales from new customers
- $1,875,000 (75% of $2,500,000) in sales from existing customers
- $50,000 in sales from customers not facing "out-of-stock or non-stocked" items

Special Issues or Constraints:

- The Marketing Department views this as a strategic system. This Internet system will add value to our current business model, and it also will serve as a proof-of-concept for future Internet endeavors. For example, in the future, CD Selections may want to sell products directly over the Internet.
- The system should be in place for the holiday shopping season next year.

FIGURE 2-A System Request for CD Selections

suggest on-going revenues of $1 million, give or take several hundred thousand dollars, after the website had been operating for a few years.

Together, the sales from existing customer ($2.5 million) and new customers ($1 million) totaled approximately $3.5 million. Margaret created conservative and optimistic estimates by reducing and increasing this figure by 25 percent. This created a possible range of values from $2,625,000 to $4,375,000. Margaret is conservative, so she decided to include the lower number as her sales projection.

Finally, Margaret wanted to impress on the project steering committee the importance of funding this specific project. To accomplish this, she made sure that the committee realized that the Marketing Department viewed the system that this project would produce as a strategic investment. And, she made certain that the committee realized, for the system to have an early success and an immediate impact, that the system really needed to be operational before the holiday shopping season next year. Consequently, she included this additional information as part of the system request.

Feasibility Analysis

Once Margaret and her Marketing group completed the system request, they submitted it to the steering committee for their next meeting. When the steering committee met, they placed the Internet Order project high on its list of projects. A senior systems analyst, Alec Adams, was assigned to help Margaret conduct a feasibility analysis because of his familiarity with CD Selections' sales and distribution systems. He also was an avid user of the Web and had been offering suggestions for the improvement of CD Selections' Web site.

Alec and Margaret worked closely together over the next few weeks on the feasibility analysis. Figure 2-B presents the executive summary page of the feasibility analysis; the report itself was about 10 pages long, and it provided additional detail and supporting documentation.

As shown in Figure 2-B, the project is somewhat risky from a technical perspective. CD Selections has minimal experience with the proposed application and the technology because the ISP had been managing most of the website technology to date. One solution may be to hire a consultant with e-commerce experience to work with the IT department and to offer guidance. Further, the new system would have to exchange order information with the company's brick-and-mortar order system. Currently, individual retail stores submit orders electronically, so receiving orders and exchanging information with the Internet systems should be possible.

The economic feasibility analysis includes refined assumptions that Margaret made in the system request. Figure 2-C shows the summary spreadsheet that lead to the conclusions on the feasibility analysis. Development costs are expected to be about $250,000. This is a very rough estimate, as Alec has had to make some assumptions about the amount of time it will take to design and program the system. These estimates will be revised after a detailed workplan has been developed and as the project proceeds[1]. Traditionally, operating costs include the costs of the computer operations. In this case, CD Selections has had to include the costs of business staff, because they are creating a new business unit, resulting in a total of about $450,000 each year. Margaret and Alec have decided to use a conservative estimate for revenues although they note the potential for higher returns. This shows that the project can still add significant business value, even if the underlying assumptions prove to be overly optimistic. The spreadsheet was projected over three years, and the ROI and break-even point were included.

The organizational feasibility is presented in Figure 2-B. There is a strong champion, well placed in the organization to support the project. The project originated in the business or functional side of the company, not the IT department, and Margaret has carefully built up support for the project among the senior management team.

This is an unusual system in that the ultimate end users are the consumers external to CD Selections. Margaret and Alec have not done any specific market research to see how well potential customers will react to the CD Selections system, so this is a risk.

An additional stakeholder in the project is the management team responsible for the operations of the traditional stores and the store managers. They should be quite supportive given the added service that they now can offer. However, Margaret must convince them that the Internet Sales System will not be viewed as a threat to stores' future sales. As such, Margaret and Alec need to make sure that the management team and store managers are included in the development of the system so that they can incorporate the system into their business processes.

[1] Some of the salary information may seem high to you. Most companies use a "full cost" model for estimating salary cost in which all benefits (e.g., health insurance, retirement, payroll taxes) are included in salaries when estimating costs.

Internet Order Feasibility Analysis Executive Summary

Margaret Mooney and Alec Adams created the following feasibility analysis for the CD Selections Internet Order System Project. The System Proposal is attached, along with the detailed feasibility study. The highlights of the feasibility analysis are as follows:

Technical Feasibility

The Internet Order System is feasible technically, although there is some risk.

CD Selections' risk regarding familiarity with Internet order applications is high

- The Marketing Department has little experience with Internet-based marketing and sales.
- The IT Department has strong knowledge of the company's existing order systems; however, it has not worked with Web-enabled order systems.
- Hundreds of retailers that have Internet Order applications exist in the marketplace.

CD Selections' risk regarding familiarity with the technology is medium.

- The IT Department has relied on external consultants and an Information Service Provider to develop its existing Web environment.
- The IT Department has gradually learned about Web systems by maintaining the current Web site.
- Development tools and products for commercial Web application development are available in the marketplace, although the IT department has little experience with them.
- Consultants are readily available to provide help in this area.

The project size is considered medium risk.

- The project team likely will include fewer than ten people.
- Business user involvement will be required.
- The project timeframe cannot exceed a year because of the holiday season implementation deadline, and it should be much shorter.

The compatibility with CD Selections' existing technical infrastructure should be good.

- The current Order System is a client-server system built using open standards. An interface with the Web should be possible.
- Retail stores already place and maintain orders electronically.
- An Internet infrastructure already is in place at retail stores and at the corporate headquarters.
- The ISP should be able to scale their services to include a new Order System.

Economic Feasibility

A cost–benefit analysis was performed; see attached spreadsheet for details. A conservative approach shows that the Internet Order System has a good chance of adding to the bottom line of the company significantly.

ROI over 3 years: 229 percent

Total benefit after three years: $3.5 million (adjusted for present value)

Break-even occurs: after 1.32 years

Intangible Costs and Benefits

- Improved customer satisfaction
- Greater brand recognition

Organizational Feasibility

From an organizational perspective, this project has low risk. The objective of the system, which is to increase sales, is aligned well with the senior management's goal of increasing sales for the company. The move to the Internet also aligns with Marketing's goal to become more savvy in Internet marketing and sales.

The project has a project champion, Margaret Mooney, Vice President of Marketing. Margaret is well positioned to sponsor this project and to educate the rest of the senior management team when necessary. To date, much of senior management is aware of and supports the initiative.

The users of the system, Internet consumers, are expected to appreciate the benefits of CD Selections' Web presence. And, management in the retail stores should be willing to accept the system, given the possibility of increased sales at the store level.

Additional Comments

- The Marketing Department views this as a strategic system. This Internet system will add value to our current business model, and it also will serve as a proof of concept for future Internet endeavors.
- We should consider hiring a consultant with expertise in similar applications to assist with the project.
- We will need to hire new staff to operate the new system, from both the technical and business operations aspects.

FIGURE 2-B Feasibility Analysis for CD Selections

	2012	2013	2014	Total
Increased sales from new customers	0	750,000	772,500	
Increased sales from existing customers	0	1,875,000	1,931,250	
Reduction in customer complaint calls	0	50,000	50,000	
TOTAL BENEFITS:	0	2,675,000	2,753,750	
PV of BENEFITS:	0	2,521,444	2,520,071	5,041,515
PV of ALL BENEFITS:	0	2,521,444	5,041,515	
Labor: Analysis and Design	42,000	0	0	
Labor: Implementation	120,000	0	0	
Consultant Fees	50,000	0	0	
Training	5,000	0	0	
Office Space and Equipment	2,000	0	0	
Software	10,000	0	0	
Hardware	25,000	0	0	
TOTAL DEVELOPMENT COSTS:	254,000	0	0	
Labor: Webmaster	85,000	87,550	90,177	
Labor: Network Technician	60,000	61,800	63,654	
Labor: Computer Operations	50,000	51,500	53,045	
Labor: Business Manager	60,000	61,800	63,654	
Labor: Assistant Manager	45,000	46,350	47,741	
Labor: 3 Staff	90,000	92,700	95,481	
Software Upgrades	1,000	1,000	1,000	
Software Licenses	3,000	1,000	1,000	
Hardware Upgrades	5,000	3,000	3,000	
User Training	2,000	1,000	1,000	
Communications Charges	20,000	20,000	20,000	
Marketing Expenses	25,000	25,000	25,000	
TOTAL OPERATIONAL COSTS:	446,000	452,700	464,751	
TOTAL COSTS:	700,000	452,700	464,751	
PV of COSTS:	679,612	426,713	425,313	1,531,638
PV of ALL COSTS:	679,612	1,106,325	1,531,638	
Total Project Benefits Costs :	(700,000)	2,222,300	2,288,999	
Yearly NPV:	(679,612)	2,094,731	2,094,758	3,509,878
Cumulative NPV:	(679,612)	1,415,119	3,509,878	
Return on Investment:	229.16%	(3,509,878/1,531,638)		
Break-even Point:	1.32 years	(Break-even occurs in year 2; (2,094,731 − 1,415,119]/2,094,731 = 0.32)		
Intangible Benefits:	Greater brand recognition Improved customer satisfaction			

FIGURE 2-C Economic Feasibility Analysis for CD Selections

Project Selection

The approval committee met and reviewed the Internet Sales System project along with two other projects–one that called for the implementation of a corporate Intranet and another that proposed in-store kiosks that would provide customers with information about the CDs that the store carried. Unfortunately, the budget would only allow for one project to be approved, so the committee carefully examined the costs, expected benefits, risks, and strategic alignment of all three projects. Currently, a primary focus of upper management is increasing sales in the retail stores and the Internet system and kiosk project best aligned with that goal. Given that both projects had equal risk, but that the Internet Order project expected a much greater return, the committee decided to fund the Internet Sales System.

Project Effort Estimation

Given the success of the system request being funded, Alec was very excited about managing the Internet Sales System project at CD Selections, but he realized that his project team would have very little time to deliver at least some parts of the system because the company wanted the application developed in time for the holiday season. Therefore, he decided that the project should follow an Enhanced Unified Process-based approach (see Figure 1-18). In this way, he could be sure that some version of the product would be in the hands of the users within several months, even if the completed system would be delivered at a later date.

As project manager, Alec had to estimate the project's effort and schedule–some of his least favorite jobs because of how tough it is to do at the very beginning of the project. But he knew that the users would expect at least general ranges for a product delivery date. He began by attempting to estimate the project's effort using use case points. Using the Use Case Point Template (see Figure 2-17) and the sample filled out worksheet (see Figure 2-18) Alec could estimate the effort to create the new system.

First, Alec had to sit down with Margaret and attempt to identify all of the different users that would interact with the system and to identify the different business processes that the system would support. At this point in time, Alec explained to Margaret that this was a very rough, but necessary estimate. Once they identified these, Alec categorized the different types of users into actors and identified the business processes as use cases. Once this was done, Alec and Margaret had to classify each actor and use case as being simple, average, or complex. In the case of the actors, Bricks and Mortar store and Distribution System had a well defined API. As such they were classified as simple actors. On the other hand, the Customer actor was classified as being complex. This gave an Unadjusted Actor Weight Total Value of 5. (See Figure 2-D).

Second, Alec and Margaret classified each use case based on the number of unique transactions that each had to handle. In this case, there were 2 simple use cases (Place InStore Hold and Place Special Order), 1 average use case (Create New Customer), and 2 complex use cases (Place Order and Checkout). Based on these, a value of 100 to the Unadjusted Use Case Weight Total was computed (See Figure 5-20).

Third, Alec computed a value of 55 for the Unadjusted Use Case Points. Fourth, he rated each of the technical complexity factors, rated each of the environmental factors, and computed the values for TCF and EF (See Figure 2-D). Fifth, using the Unadjusted Use Case Points and the TCF and EF values, Alec calculated a value of 67.5 for Adjusted Use Case Points. Sixth, based on the decision rule to determine whether to use 20 or 28 as the value of the person hours multiplier, Alec realized that he should use 28. Next, Alec was able to estimate the effort for the project to be 1,889.89 person hours. This equates to about 12 person months (1,889.89/160). In other words, it would take a single person working full time about 1 year to complete the project. However, Alec remembered the hurricane model (see Figure 2-22) and realized that at this point in the development, this estimate would have to be modified substantially since they have not created a requirements definition, system proposal, or a business process and functional model. Furthermore, since CD Selections had never built this type of system, based on his past experience with other systems, he suggested to simply double the estimate for right now and to redo it once they have a better understanding of the requirements.

Unadjusted Actor Weighting Table:

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	2	2
Average	External System using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	0	0
Complex	Human	3	1	3
		Unadjusted Actor Weight Total (UAW)		**5**

Unadjusted Use Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	2	10
Average	4–7 transactions	10	1	10
Complex	>7 transactions	15	2	30
		Unadjusted Use Case Weight Total (UUCW)		**50**

Unadjusted use case points (UUCP) = UAW + UUCW 55 = 5 + 50

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
T1	Distributed system	2.0	5	10.0	
T2	Response time or throughput performance objectives	1.0	5	5.0	
T3	End-user online efficiency	1.0	5	5.0	
T4	Complex internal processing	1.0	4	4.0	
T5	Reusability of code	1.0	3	3.0	
T6	Easy to install	0.5	3	1.5	
T7	Ease of use	0.5	5	2.5	
T8	Portability	2.0	4	8.0	
T9	Ease of change	1.0	3	3.0	
T10	Concurrency	1.0	3	3.0	
T11	Special security objectives included	1.0	5	5.0	
T12	Direct access for third parties	1.0	5	5.0	
T13	Special User training required	1.0	3	3.0	
			Technical Factor Value (TFactor)	**58.0**	

Technical complexity factor (TCF) = 0.6 + (0.01 * TFactor) 1.18 = 0.6 + (0.01 * 58)

Environmental Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	1	1.5	
E2	Application experience	0.5	2	1.0	
E3	Object-oriented experience	1.0	0	0.0	
E4	Lead analyst capability	0.5	3	1.5	
E5	Motivation	1.0	4	4.0	
E6	Requirements stability	2.0	4	8.0	
E7	Part time staff	–1.0	0	0.0	
E8	Difficulty of programming language	–1.0	4	–4.0	
			Environmental Factor Value (EFactor)	**12.0**	

Environmental factor (EF) = 1.4 + (–0.03 * EFactor) 1.04 = 1.4 + (–.03 * 12)
Adjusted use case points (UCP) = UUCP *TCF *ECF 67.5 = 55 * 1.18 * 1.04
Person hours multiplier (PHM) PHM = 28
Person hours = UPC * PHM 1,889.9 = 67.5 * 28

FIGURE 2-D Use-Case Points Estimation for the Internet Sales Systems

Creating and Managing the Workplan

Once the estimation was underway, Alec began to create an evolutionary work breakdown structure and iterative workplan to identify the tasks that would be needed to complete the system. He started by reviewing the Enhanced Unified Process phases and workflows (see Figure 1-18) and the evolutionary work breakdown structure template (see Figure 2-19). At this point in time, Alec does not know enough to create a complete workplan. Consequently, he has included as much detail as he knows to be correct (see Figure 2-E). For example, Alec feels confident about the estimation of time to create the requirements definition and to elicit the requirements. However, he does not know whether how long it will take to develop the functional, structural, or behavioral analysis models until after he has a better idea of the actual requirements. Until this determination can be made, any estimation as to the time required would be simply a guess. As time passes, Alec expects to know much more about the development process and will add much more detail to the workplan. (Remember that the development process and the project management processes are iterative and incremental in nature.)

Staffing the Project

Alec next turned to the task of how to staff his project. On the basis of his earlier estimates, it appeared that about 3 people would be needed to deliver the system by the holidays (24 person-months over 10 months of calendar time means 3 people, rounded up).

First, he created a list of the various roles that he needed to fill. He thought he would need a couple of analysts to work with the analysis and design of the system as well as an infrastructure analyst to manage the integration of the Internet Sales System

	Duration	Dependency
I. Business Modeling		
a. Inception		
1. Understand current business situation		
2. Uncover business process problems		
3. Identify potential projects		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
II. Requirements		
a. Inception		
1. Identify appropriate requirements analysis technique		
2. Identify appropriate requirements gathering techniques		
3. Identify functional and nonfunctional requirements		II.a.1, II.a.2
4. Analyze current systems		II.a.1, II.a.2
5. Create requirements definition		II.a.3, II.a.4
A. Determine requirements to track		
B. Compile requirements as they are elicited		II.a.5.A
C. Review requirements with sponsor		II.a.5.B

(Continued)

FIGURE 2-E
Evolutionary Work Breakdown Structure for the Inception Phase for CD Selections.

FIGURE 2-E
(Continued)

	Duration	Dependency
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
III. Analysis		
a. Inception		
1. Identify business processes		
2. Identify use cases		III.a.1
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
IV. Design		
a. Inception		
1. Identify potential classes		III.a
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
V. Implementation		
a. Inception		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
VI. Test		
a. Inception		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
VII. Deployment		
a. Inception		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
VIII. Configuration and change management		
a. Inception		
1. Identify necessary access controls for developed artifacts		
2. Identify version control mechanisms for developed artifacts		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		

	Duration	Dependency
IX. Project management		
a. Inception		
1. Create workplan for the inception phase		
2. Create system request		
3. Perform feasibility analysis		IX.a.2
A. Perform technical feasibility analysis		
B. Perform economic feasibility analysis		
C. Perform organizational feasibility analysis		
4. Identify project size		IX.a.3
5. Identify staffing requirements		IX.a.4
6. Compute cost estimate		IX.a.5
7. Create workplan for first iteration of the elaboration phase		IX.a.1
8. Assess inception phase		I.a, II.a, III.a IV.a, V.a, VI.a VII.a, VIII.a, IX.a, X.a, XI.a XII.a
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
X. Environment		
a. Inception		
1. Acquire and install CASE tool		
2. Acquire and install programming environment		
3. Acquire and install configuration and change management tools		
4. Acquire and install project management tools		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
XI. Operations and Support		
a. Inception		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
XII. Infrastructure Management		
a. Inception		
1. Identify appropriate standards and enterprise models		
2. Identify reuse opportunities, such as patterns, frameworks, and libraries		
3. Identify similar past projects		
b. Elaboration		
c. Construction		
d. Transition		
e. Production		

FIGURE 2-F
Staffing Plan for the Internet Sales System

Role	Description	Assigned To
Project Manager	Oversees the project to ensure that it meets its objectives in time and within budget.	Alec
Infrastructure Analyst	Ensures the system conforms to infrastructure standards at CD Selections. Ensures that the CD Selections infrastructure can support the new system.	Anne
Systems Analyst	Designs the information system—with a focus on interfaces with the distribution system.	Anne
Systems Analyst	Designs the information system—with a focus on the data models and system performance.	Brian
Programmer	Codes system.	Anne

Reporting Structure: All project team members will report to Alec.

Special Incentives: If the deadline for the project is met, all team members who contributed to this goal will receive a free day off to be taken over the holiday season.

with CD Selections' existing technical environment. Alec also needed people who had good programmer skills and who could be responsible for ultimately implementing the system. Anne and Brian are two analysts with strong technical and interpersonal skills (although Anne is less balanced, having greater technical than interpersonal abilities), and Alec believed that they were available to bring onto this project. He wasn't certain if they had experience with the actual Web technology that would be used on the project, but he decided to rely on vendor training or an external consultant to build those skills later when they were needed. Because the project was so small, Alec envisioned all of the team members reporting to him because he would be serving as the project's manager.

Alec created a staffing plan that captured this information, and he included a special incentive structure in the plan (see Figure 2-F). Meeting the holiday deadline was very important to the project's success, so he decided to offer a day off to the team members who contributed to meeting that date. He hoped that this incentive would motivate the team to work very hard. Alec also planned to budget money for pizza and sodas for times when the team worked long hours.

Before he left for the day, Alec drafted a project charter, to be fine-tuned after the team got together for its kickoff meeting (i.e., the first time the project team gets together). The charter listed several norms that Alec wanted to put in place from the start to eliminate any misunderstanding or problems that could come up otherwise (see Figure 2-G).

FIGURE 2-G
Project Charter for the Internet Order System

Project objective: The Internet order system project team will create a working Web-based system to sell CDs to CD Selections' customers in time for the holiday season.

The Internet order system team members will:

1. Attend a staff meeting each Friday at 2 PM. to report on the status of assigned tasks.
2. Update the workplan with actual data each Friday by 5 PM.
3. Discuss all problems with Alec as soon as they are detected.
4. Agree to support each other when help is needed, especially for tasks that could hold back the progress of the project.
5. Post important changes to the project on the team bulletin board as they are made.

Coordinating Project Activities

Alec wanted the Internet Sales System project to be well coordinated, so he immediately put several practices in place to support his responsibilities. First, he acquired the CASE tool used at CD Selections and set up the product so that it could be used for the analysis-phase tasks (e.g., drawing the functional, structural, and behavioral models). The team members would likely start creating diagrams and defining components of the system fairly early on. He pulled out some standards that he uses on all development projects and made a note to review them with his project team at the kickoff meeting for the system. He also had his assistant set up binders for the project deliverables that would start rolling in. Already he was able to include the system request, feasibility analysis, initial workplan, staffing plan, project charter, standards list, and risk assessment.

CHAPTER 3: REQUIREMENTS DETERMINATION

In this chapter, we introduced how the requirements are determined in object-oriented systems development projects. Specifically, we described what a requirement is, how to create a requirements definition, and a set of problems that can arise when determining requirements. Next, we reviewed three different requirements analysis strategies, along with a set of techniques that can be used in conjunction with the strategies. After that, we reviewed a set of generic requirements-gathering techniques and a couple of alternative techniques that can be used with an object-oriented system development project. Finally, we showed how the results of the requirements determination processes, along with an updated system request, feasibility analysis, and workplan, are organized into and documented by a system proposal. In this installment of the CD Selections case, we see how Alec and Margaret work through all of these topics with regards to the Web-based solution that they hope to create.

Once the CD Selections steering committee approved the system proposal and feasibility analysis, the project team began performing project management and analysis activities. In addition to the material described in the previous installment, these activities included gathering requirements using a variety of techniques, and analyzing the requirements that were gathered. Furthermore, Alec and Margaret decided to hire an Internet marketing and sales consultant, Chris Campbell, to advise Alec, Margaret, and the project team during the inception phase. Some highlights of the project team's activities are presented below.

Requirements Analysis Techniques

Margaret suggested that the project team conduct several joint application development (JAD) sessions with store managers, marketing analysts, and Web-savvy members of the IT staff. Together, the groups could work through some business process improvement (BPI) techniques and brainstorm how improvements could be made to the current order process using a new Web-based system.

Alec facilitated three JAD sessions that were conducted over the course of a week. Alec' past facilitation experience helped the 8-person meetings run smoothly and stay on track. First, Alec used technology analysis, and suggested several important Web technologies that could be used for the system. The JAD session generated ideas about how CD Selections could apply each of the technologies to the Internet order project. Alec had the group categorize the ideas into three sets: "definite" ideas that would have a good probability of providing business value; "possible" ideas that might add business value; and "unlikely" ideas.

Next, Alec applied informal benchmarking by introducing the Web sites of several leading retailers and pointing out the features that they offered on-line. He selected some sites based on their success with Internet sales, and others based on their similarity to the vision for CD Selections' new system. The group discussed the features that were common across most retailers versus unique functionality, and they created a list of suggested business requirements for the project team.

Requirements Gathering Techniques

Alec believed that it would be important to understand the order processes and systems that already existed in the organization because they would have to be closely integrated with the Web order system. Three requirements gathering techniques proved to be helpful in understanding the current systems and processes–document analysis, interviews, and observation.

First, the project team collected existing reports (e.g., order forms, screenshots of the on-line order screens) and system documentation (functional, structural and behavioral models) that shed light on the as-is system. They were able to gather a good amount of information about the brick-and-mortar order processes and systems in this way. When questions arose, they conducted short interviews with the person who provided the documentation for clarification.

Next, Alec interviewed the senior analysts for the order and inventory systems to get a better understanding of how those systems worked. He asked if they had any ideas for the new system, as well as any integration issues that would need to be addressed. Alex also interviewed a contact from the ISP and the IT person who supported CD Selections' current website–both provided information about the existing communications infrastructure at CD Selections and its Web capabilities. Finally, Alex spent a half-day visiting two of the retail stores and observing exactly how the order and hold processes worked in the brick-and-mortar facilities.

Requirements Definition

Throughout all of the above activities, the project team collected information and tried to identify the business requirements for the system from the information. As the project progressed, requirements were added to the requirements definition and grouped by requirement type. When questions arose, they worked with Margaret, Chris, and Alec to confirm that requirements were in scope. The requirements that fell outside of the scope of the current system were typed into a separate document that would be saved for future use.

After gathering and documenting the requirements, the requirements definition was distributed to Margaret, two marketing employees who would work with the system on the business side, and several retail store managers. This group then met for a two-day JAD session to clarify, finalize, and prioritize business requirements[2].

The project team spent time creating functional, structural and behavioral models (Chapters 4, 5, and 6) that depicted the objects in the future system. Members of marketing and IT departments reviewed the documents during interviews with the project team. Figure 3-A shows a portion of the final requirements definition and Figure 3-B represents the requirements in the form of a concept map.

System Proposal

Alec reviewed the requirements definition and the other deliverables that the project team created during the inception phase. Given Margaret's desire to have the system operating before next year's Christmas season, Alec decided to timebox the project, and he determined what functionality could be included in the system by that schedule deadline (see Chapter 2). He suggested that the project team develop the system in three versions rather than attempting to develop a complete system that provided all the features initially. The first version, to be operational well before the holidays, would implement a "basic" system that would have the "standard" order features of other Internet retailers. The second version, planned for late spring or early summer, would have several features unique to CD Selections. The third version would add more "advanced" features, such as the ability to listen to a sample of music over the Internet, to find similar CDs, and to write reviews.

[2] This JAD session was not originally planned. As such, the workplan (see Figure 2-E) should be modified

Nonfunctional Requirements

1. Operational Requirements

   1.1 The Internet sales system will draw information from the main CD information database, which contains basic information about CDs (e.g., title, artist, ID number, price, quantity in inventory). The Internet sales system will not write information to the main CD information database.

   1.2 The Internet sales system will store orders for new CDs in the special order system and will rely on the special order system to complete the special orders generated.

   1.3 A new module for the in-store system will be written to manage the “holds” generated by the Internet sales system. The requirements for this new module will be documented as part of the Internet sales system because they are necessary for the Internet sales system to function.

2. Performance Requirements

   No special performance requirements are anticipated.

3. Security Requirements

   No special security requirements are anticipated.

4. Cultural and Political Requirements.

   No special cultural and political requirements are anticipated.

Functional Requirements

1. Maintain CD Information

   1.1 The Internet sales system will need a database of basic information about the CDs that it can sell over the Internet, similar to the CD database at each of the retail stores (e.g., title, artist, ID number, price, quantity in inventory).

   1.2 Every day, the Internet sales system will receive an update from the distribution system that will be used to update this CD database. Some new CDs will be added, some will be deleted, and others will be revised (e.g., a new price).

   1.3 The electronic marketing (EM) manager (a position that will need to be created) will also have the ability to update information (e.g., prices for sales).

2. Maintain CD Marketing Information

   2.1 The Internet sales system provides an additional opportunity to market CDs to current and new customers. The system will provide a database of marketing materials about selected CDs that will help Web users learn more about them (e.g., music reviews, links to Web sites, artist information, and sample sound clips). When information about a CD that has additional marketing information is displayed, a link will be provided to the additional information.

   2.2 Marketing materials will be supplied primarily by vendors and record labels so that we can better promote their CDs. The EM manager of the marketing department will determine what marketing materials will be placed in the system and will be responsible for adding, changing, and deleting the materials.

3. Place Order

   3.1 Customers will access the Internet sales system to look for CDs of interest. Some customers will search for specific CDs or CDs by specific artists, whereas other customers will want to browse for interesting CDs in certain categories (e.g., rock, jazz, classical).

   3.2 When the customer has found all the CDs he or she wants, the customer will "check out" by providing personal information (e.g., name, e-mail, address, credit card), and information regarding the order (e.g., the CDs to purchase, and the quantity for each item).

   3.3 The system will verify the customer's credit card information with an online credit card center and either accept the order or reject it.

   3.4 Customers will also be able check to see if their preferred stores have the CDs in stock. They will use zip code to find stores close to their location. If the CD is available at a preferred store, a customer can immediately place a hold on the CD in stock and then come into the store and pick it up.

   3.5 If the CD is not available in the customer’s preferred store, the customer can request that the CD be special ordered to that store for later pickup. The customer will be notified by e-mail when the requested CD arrives at the requested store; the CD will be placed on hold (which will again expire after 7 days). This process will work similarly to the current special order systems already available in the regular stores.

   3.6 Alternatively, the customer can mail order the CD (see requirement 4).

4. Fill Mail Orders

   4.1 When a CD is mail-ordered, the Internet sales system will send the mail order to the mail order distribution system.

   4.2 The mail-order distribution system will handle the actual sending of CDs to customers; it will notify the Internet sales system and e-mail the customer.

   4.3 Weekly reports can be run by the EM manager to check the order status.

FIGURE 3-A CD Selections Requirements Definition

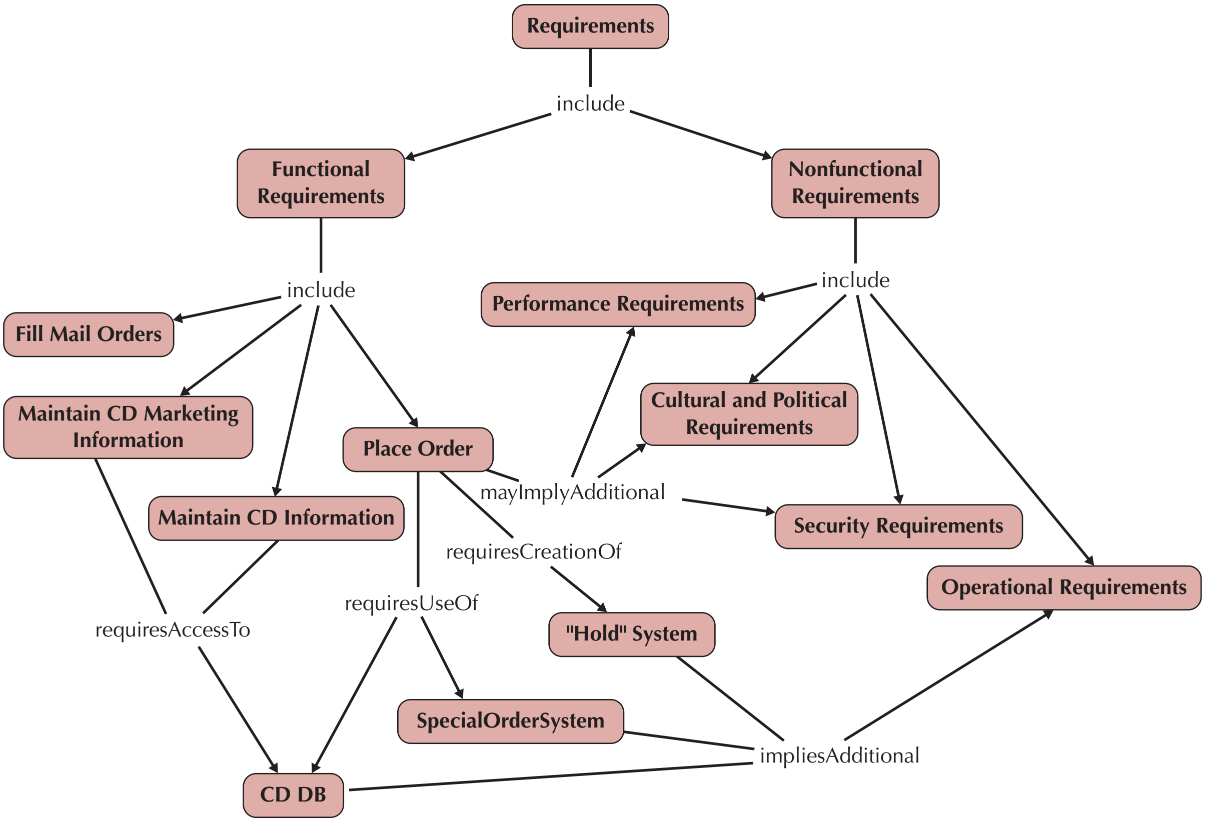


FIGURE 3-B Concept Map Requirements Model

Based on the requirements definition, Alec revised the workplan accordingly, and he worked with Margaret and the folks in marketing to review the feasibility analysis and update it where appropriate. Furthermore, Alec and Margaret realized that they had missed both actors and use cases. Consequently, they went back and revised the project effort estimation (see Figure 3-C). At this point in time, the effort estimation went from about 12 person months to over 20 person months. As Alec explained to Margaret, until we complete a rough functional model of the business processes, this estimation is still fairly volatile. However, since Alec, Margaret, and the development team have a reasonable understanding of the functional requirements, they decided to stay with the original effort estimation of 24 person months (Remember, Alec doubled the estimation.). Using the System Proposal Template in Figure 3-16, Alec combined all of the deliverables from the project a System Proposal and submitted to the steering committee for approval. Figure 3-D shows the outline of the CD Selections System Proposal. Margaret and Alec met with the committee and presented the highlights of what was learned during the inception phase and the final concept of the new system. Based on the proposal and presentation, the steering committee decided that they would continue to fund the Internet Sales System.

Unadjusted Actor Weighting Table:

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	2	2
Average	External System using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	1	2
Complex	Human	3	2	6
		Unadjusted Actor Weight Total (UAW)		**10**

Unadjusted Use Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	3	15
Average	4–7 transactions	10	1	10
Complex	>7 transactions	15	4	60
		Unadjusted Use Case Weight Total (UUCW)		**85**

Unadjusted use case points (UUCP) = UAW + UUCW 95 = 10 + 85

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
T1	Distributed system	2.0	5	10.0	
T2	Response time or throughput performance objectives	1.0	5	5.0	
T3	End-user online efficiency	1.0	5	5.0	
T4	Complex internal processing	1.0	4	4.0	
T5	Reusability of code	1.0	3	3.0	
T6	Easy to install	0.5	3	1.5	
T7	Ease of use	0.5	5	2.5	
T8	Portability	2.0	4	8.0	
T9	Ease of change	1.0	3	3.0	
T10	Concurrency	1.0	3	3.0	
T11	Special security objectives included	1.0	5	5.0	
T12	Direct access for third parties	1.0	5	5.0	
T13	Special User training required	1.0	3	3.0	
			Technical Factor Value (TFactor)	**58.0**	

Technical complexity factor (TCF) = 0.6 + (0.01 * TFactor) 1.18 = 0.6 + (0.01 * 58)

Environmental Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	1	1.5	
E2	Application experience	0.5	2	1.0	
E3	Object-oriented experience	1.0	0	0.0	
E4	Lead analyst capability	0.5	3	1.5	
E5	Motivation	1.0	4	4.0	
E6	Requirements stability	2.0	4	8.0	
E7	Part time staff	–1.0	0	0.0	
E8	Difficulty of programming language	–1.0	4	–4.0	
			Environmental Factor Value (EFactor)	**12.0**	

Environmental factor (EF) = 1.4 + (–0.03 * EFactor) 1.04 = 1.4 + (–.03 * 12)
Adjusted use case points (UCP) = UUCP *TCF *ECF 116.584 = 95 * 1.18 * 1.04
Person hours multiplier (PHM) PHM = 28
Person hours = UPC * PHM 3,264.352 = 116.584 * 28

FIGURE 3-C Use-Case Points Estimation for the Internet Sales Systems

1. Table of Contents

2. Executive Summary

   To be completed once everything else is done.

3. System Request

   See Figure 2-A.

4. Workplan

   See Figure 2-E.

5. Feasibility Analysis

   See Figures 2-B and 2-C.

6. Requirements Definition

   See Figures 3-A and 3-B.

7. Functional Model

   To be completed in the future (see Chapter 4).

8. Structural Models

   To be completed in the future (see Chapter 5).

9. Behavioral Model

   To be completed in the future (see Chapter 6).

Appendices

A. Effort Estimate

   See Figure 3-C.

B. Staffing Plan

   See Figure 2-F.

C. Project Charter

   See Figure 2-G.

FIGURE 3-D
Outline of the CD Selections System Proposal

CHAPTER 4: BUSINESS PROCESS AND FUNCTIONAL MODELING

In this chapter, we introduced how business processes are identified, modeled, and documented using the functional models of the UML. Specifically, we described how the functional requirements of business processes are identified by use cases and use-case diagrams. We described how activity diagrams model business processes and we described how use-case descriptions are used to more fully document the business processes. Finally, we described how to verify and validate the evolving representations of the business processes contained in the functional models. The basic functional and non-functional requirements for the CD Selections Internet Sales System were developed previously. At this point in time, you should go back and carefully review these requirements (see Figures 2-A, 2-B, 2-C, 2-E, 3A, 3B, and 3-C). In this installment of the CD Selections case, we see how Alec and Margaret work through all of these topics with regard to the Web-based solution that they hope to create.

Business Process Identification with Use-Cases and Use-Case Diagrams

As a first step toward developing a model of the functional requirements, Alec and his team decided to model these high-level business processes as use cases and to draw a use case diagram showing the interaction between business processes and the systems environment and how the business processes interact among themselves. To begin the business process identification process, Alec and the team went back and reviewed the requirements definition (see Figure 3-A).

The first business process identified in the requirements definition was Maintain CD Information. The Distribution System triggers this business process when it distributes new information for use in the CD database. Besides the Distribution System, another stakeholder was identified: EM Manager. Vendors trigger the second business process, Maintain CD Marketing Information, when CD Selections receive new marketing materials. An again, it seemed to the project team that the Electronic Marketing (EM) Manager would be an interested stakeholder. The third business process, Place Order, is much more interesting. In this case, the customer triggers the process. Again, the EM Manager is an interested stakeholder. This process has many more inputs. The final business process, Fill Mail Orders, seems to deal with the distribution system, customers, a credit card center, and the EM Manager. Furthermore, unlike the other business processes, it is unclear what exactly triggers its execution.

Once the team felt comfortable with their understanding of the requirements, they began the modeling process by setting the scope of the project. To begin with, they felt that the subject boundary should be drawn in such a manner that anything that is not part of CD Selections' Internet Sales System, such as the vendors, credit card center, and customers should be identified as primary actors. Therefore, these were considered outside of the scope of the system. The other potential actors identified could be the distribution system, EM Manager, and the current CD Selections stores. Upon closer review of Figure 3-A, Alec and Margaret felt that the distribution system and the current CD Selections stores should be outside the scope of the Internet Sales system. Consequently, they also should be identified as primary actors. In the case of the EM Manager, Alec and Margaret believed that the EM Manager should be considered as part of the Internet Sales System and therefore should not be identified as a primary actor. Remember, primary actors are only those that can be viewed as being outside of the scope of the system. The decision on whether the EM Manager, the current CD Selections stores, or the distribution system is inside or outside of the system is somewhat arbitrary. From a customer's perspective, the distribution system and the current CD Selections stores could be seen as being inside of the overall system and it could be argued that the EM Manger is a primary user of the Internet Sales System. At this point in the process, it was important to make a decision and to move on. During the process of writing the detailed use cases, there would be ample opportunities to revisit this decision to determine whether the set of use cases identified are necessary and sufficient to describe the requirements of the Internet Sales System for CD Selections. As you can see, based on the above, finding the systems boundaries and listing the primary actors are heavily intertwined.

Based on the above, Alec and his team decided that the four high-level business processes should be modeled as use cases. However, upon closer review of and reflection on the Place Order process, the team felt that the Fill Mail Order business process seemed to be best treated as a part of the Place Order process instead of a separate business process (see Point 3.6 of the Functional Requirements in Figure 3-A). Based on these decisions, Alec created a first cut use case diagram of the Internet Sales System. He followed the guidelines in the textbook. First, he decided where to place the use cases on the diagram and drew them. Second, based on the actors that would interact with the different use cases, he decided where to place them in the diagram and drew them there. Third, he drew the subject boundary to portray the scope of the system. And finally, he drew the associations between the actors and the use cases to portray the interactions between the system processes and the systems environment. Figure 4-A portrays the first cut use case diagram created by Alec.

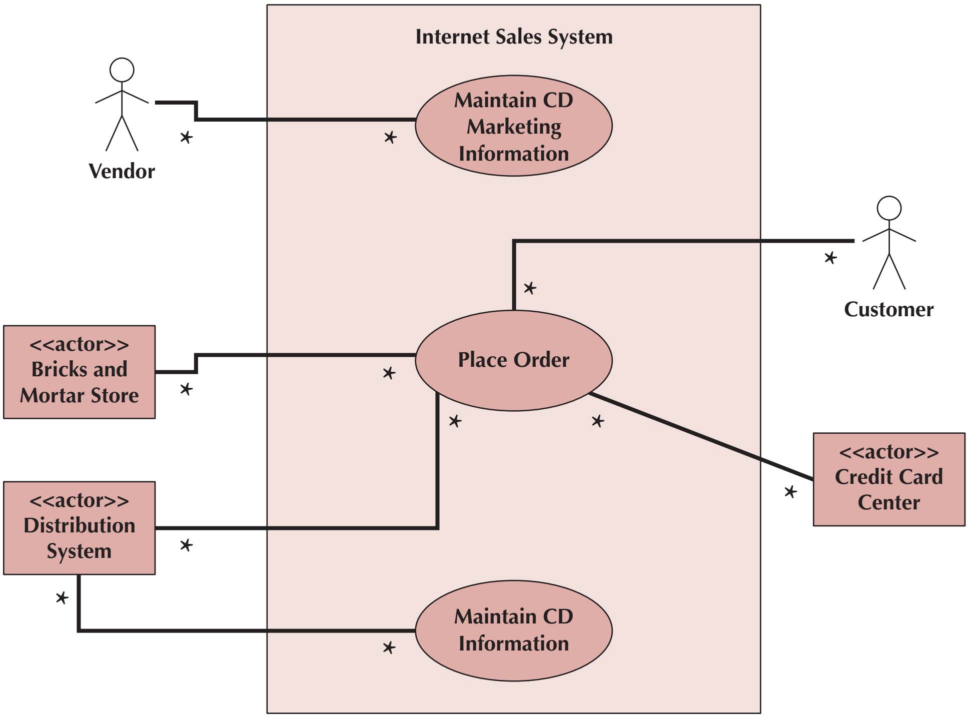

Internet Sales System
Vendor
Maintain CD Marketing Information
Customer
<<actor>> Bricks and Mortar Store
Place Order
<<actor>> Credit Card Center
<<actor>> Distribution System
Maintain CD Information
*

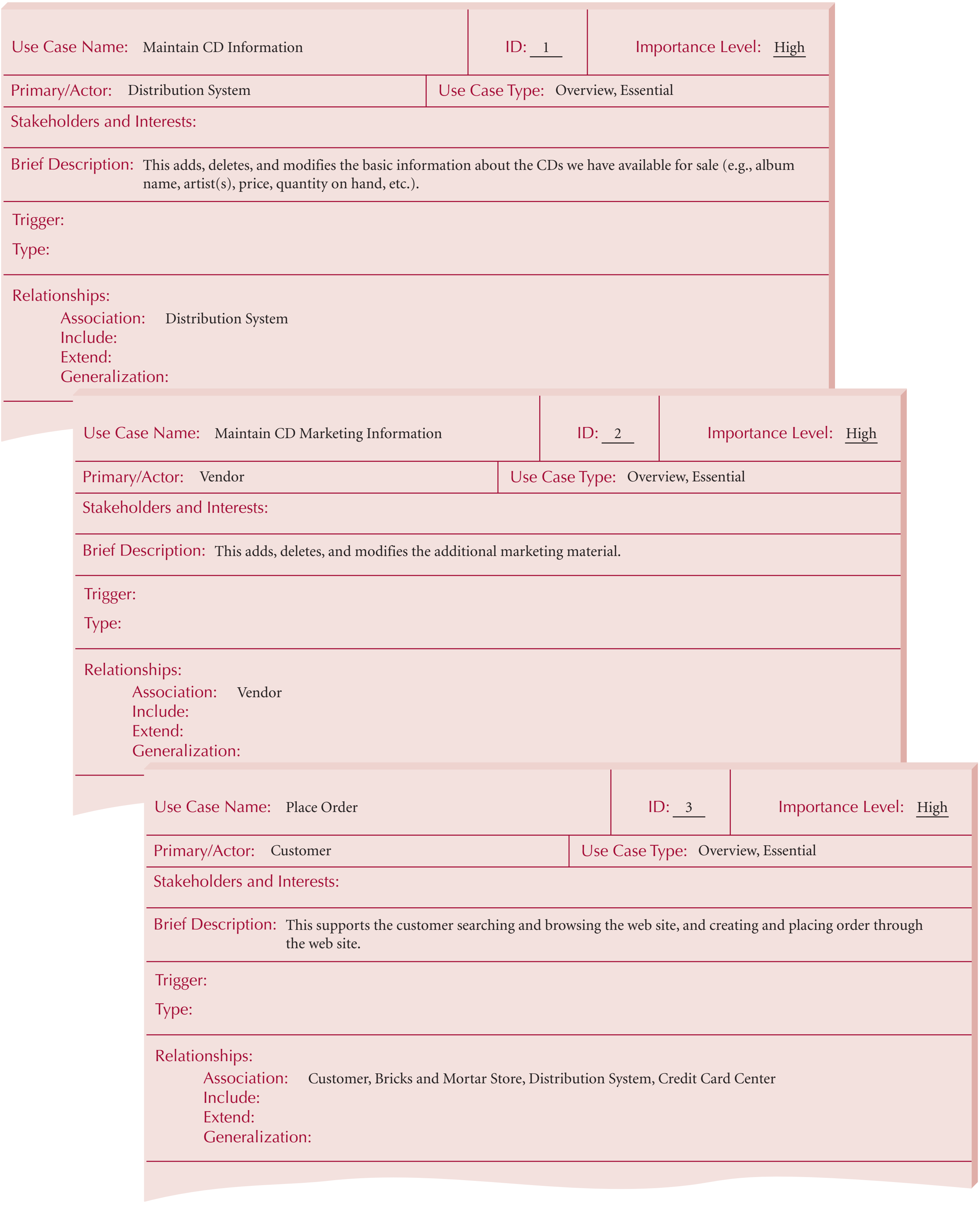
FIGURE 4-A

First-Cut Use Case Diagrams for CD Selections

At this point in the process, the project team began writing the overview use cases for the three high-level business processes: Maintain CD Information, Maintain CD Marketing Information, and Place Order. Remember, that an overview use case only has five pieces of information: use case name, ID number, primary actor, type, and a brief description. Having drawn the use case diagram, the team has already identified the primary actors and has associated the actors with the three use cases. Furthermore, since they are just beginning the development process, all three use cases type will be Overview and Essential. Since the ID numbers are simply used for identification purposes (i.e., they act as a key in a database), their values can be assigned in a sequential manner. This only left the team with two pieces of information for each use case to write. The use case name should be an action verb/noun phrase combination (e.g., Make Appointment). In the CD Selections Internet Sales situation, Maintain CD Information, Maintain CD Marketing Information, and Place Order seem to capture the essence of each of the use cases. Finally, a brief description was written to describe the purpose of the use case or the goal of the primary actor using the use case. Even though the description can range from a sentence to a short essay, the team only wanted to capture the primary issues in the use case and to make them explicit. Finally, the team carefully reviewed the current set of use cases. Take a moment to review the use cases and make sure you understand them. Based on the descriptions of all three use cases, the team felt that these three were a good basic representation of the primary business processes in the system. Figure 4-B portrays the overview, essential use cases for the Maintain CD Information, Maintain CD Marketing Information, and Place CD Orders use cases.

Business Process Modeling with Activity Diagrams

Reviewing the functional requirements described in Figure 3-A, the overview, essential use cases and the first cut Use Case Diagram, Alec sat down with Margaret to prioritize the three use cases. Based on their meeting, it was decided that the development team should focus on the most difficult and largest of the use cases first: Place Order. Consequently, Alec and the development team decided to carefully review point 3 of the functional requirements (see Figure 3-A). Upon doing this, the team identified a set of additional sub-processes that needed to be addressed: Search/Browse CDs (see point 3.1), Checkout (see point 3.2), Verify Credit Card Information (see point 3.3), Place in Store Hold (see point 3.4), Place Special Order (see point 3.5), and Fill Mail Order (see points 3.6 and 4). By carefully reviewing these sub-processes, the team realized that there needed to be a sub-process associated with the Checkout process that would support the creation of new customers. Upon further discussion and reflection, the team also decided to fold the Verify Credit Card Information sub-process back into the Checkout sub-process. The logic behind this decision was that this sub-process was really simply an action in the Checkout sub-process that simply sent a message to a Credit Card Center. Consequently, it did not make sense to factor it out as a separate sub-process. After the team had completed this process, Alec decided to create an activity diagram that portrayed the logical flow through the Place CD Order use case (business process). Following the process to draw an activity diagram described in the textbook, Alex decided to model the six sub-processes (Search/Browse CDs, Checkout, Create New Customer, Place in Store Hold, Place Special Order, and Fill Mail Order) as activities. Next, he identified the three decisions that needed to be modeled (to place an order or not, to create a new customer or not, and whether the customer wanted to place a special order, an in store hold, or a mail order). He then identified the control flows necessary to link the activities and control nodes together. The resulting activity



Use Case Name: Maintain CD Information	ID: 1	Importance Level: High
Primary/Actor: Distribution System	Use Case Type: Overview, Essential	

Stakeholders and Interests:

Brief Description: This adds, deletes, and modifies the basic information about the CDs we have available for sale (e.g., album name, artist(s), price, quantity on hand, etc.).

Trigger:

Type:

Relationships:
Association: Distribution System
Include:
Extend:
Generalization:

Use Case Name: Maintain CD Marketing Information	ID: 2	Importance Level: High
Primary/Actor: Vendor	Use Case Type: Overview, Essential	

Stakeholders and Interests:

Brief Description: This adds, deletes, and modifies the additional marketing material.

Trigger:

Type:

Relationships:
Association: Vendor
Include:
Extend:
Generalization:

Use Case Name: Place Order	ID: 3	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests:

Brief Description: This supports the customer searching and browsing the web site, and creating and placing order through the web site.

Trigger:

Type:

Relationships:
Association: Customer, Bricks and Mortar Store, Distribution System, Credit Card Center
Include:
Extend:
Generalization:

FIGURE 4-B Overview of the three Major Use Cases (Business Processes) for CD Selections

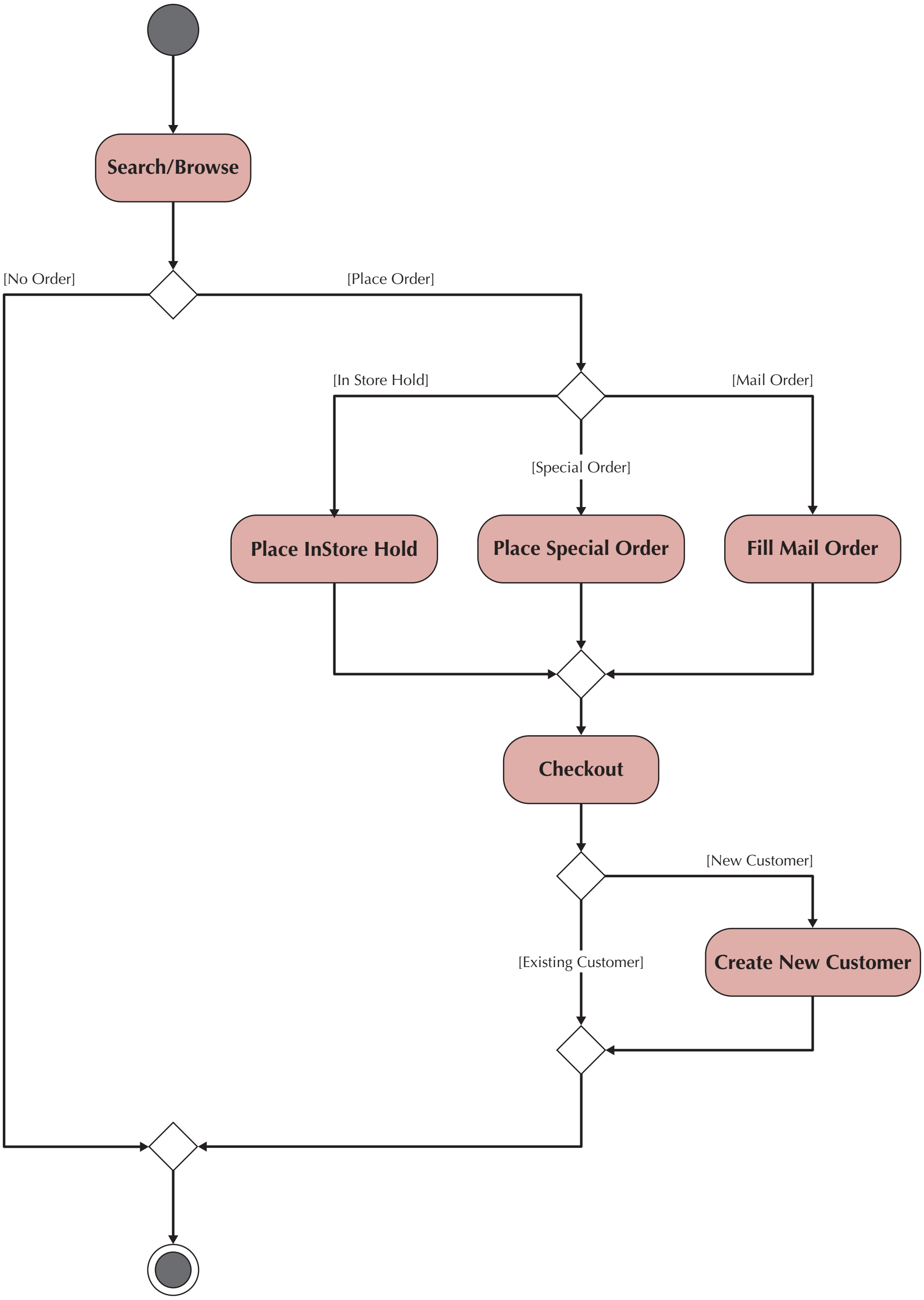


FIGURE 4-C Activity Diagram for the Place Order Use Case for CD Selections

diagram representing the Place Order use case (business process) is portrayed in Figure 4-C. Finally, the team decided to go back and modify the Use Case diagram to reflect these changes. In this case, the team decided to model each of the sub-processes as a separate use case (see Figure 4-D). This then required the team to go back and

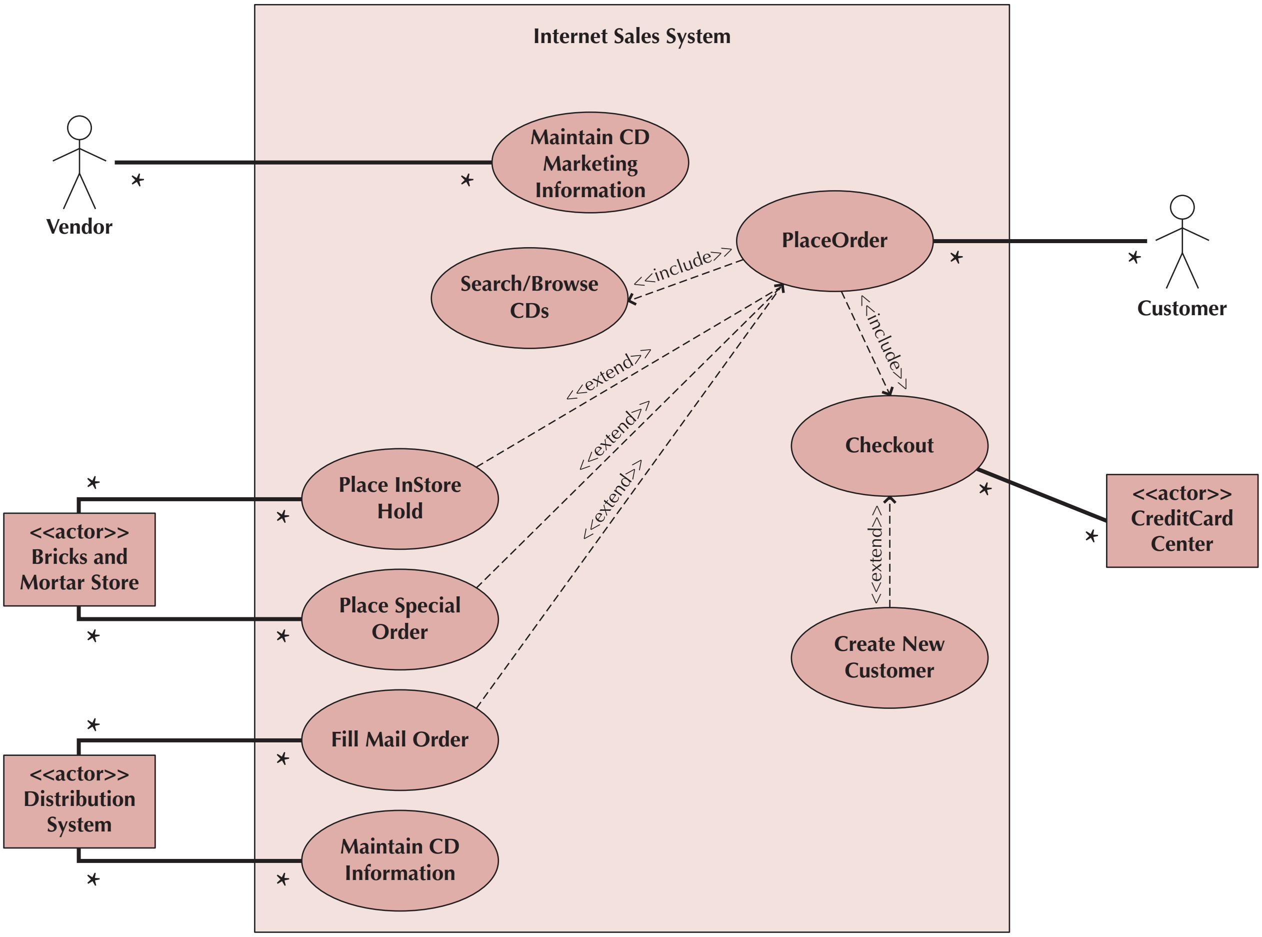
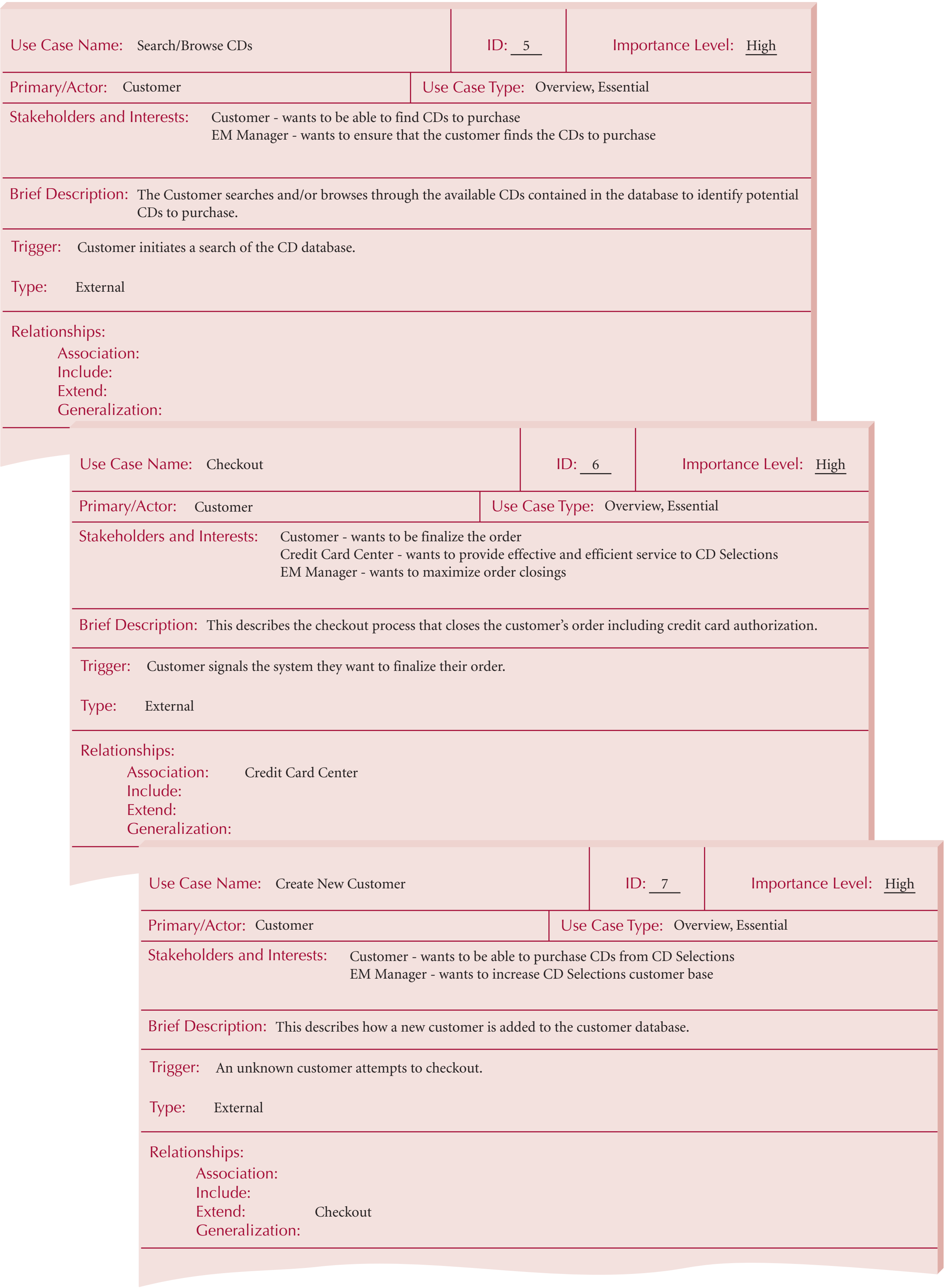


FIGURE 4-D Use Case Diagram for CD Selections

create an overview, essential use case description for each of the newly identified use cases (see Figure 4-E).

Business Process Documentation with Use Cases and Use Case Descriptions

Based on their review of the overview, essential use case descriptions, the project team decided that the Place Order use case was the most interesting. Based on this decision, the team began the conversion of the overview essential use case description to a detail, essential use case description. At this point in time, in addition to the information already contained in the overview, essential use case description (see Figure 4-B), the project team had the information necessary to fill in the stakeholders and interests and the trigger and type. In this case, the stakeholders were the customer actor and the EM Manager. The trigger was the customer visiting the web site to place an order. Since the customer is an actor, the type of trigger would be external. If the EM Manager had triggered the use case, the type would have been internal since the EM Manager was deemed earlier to be part of the system, i.e., the EM Manager is possibly an internal actor. Finally, it is possible for temporal triggers to exist where the use case would be triggered by the system's clock.



Use Case Name: Search/Browse CDs	ID: 5	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests: Customer - wants to be able to find CDs to purchase
EM Manager - wants to ensure that the customer finds the CDs to purchase

Brief Description: The Customer searches and/or browses through the available CDs contained in the database to identify potential CDs to purchase.

Trigger: Customer initiates a search of the CD database.

Type: External

Relationships:
Association:
Include:
Extend:
Generalization:

Use Case Name: Checkout	ID: 6	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests: Customer - wants to be finalize the order
Credit Card Center - wants to provide effective and efficient service to CD Selections
EM Manager - wants to maximize order closings

Brief Description: This describes the checkout process that closes the customer's order including credit card authorization.

Trigger: Customer signals the system they want to finalize their order.

Type: External

Relationships:
Association: Credit Card Center
Include:
Extend:
Generalization:

Use Case Name: Create New Customer	ID: 7	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests: Customer - wants to be able to purchase CDs from CD Selections
EM Manager - wants to increase CD Selections customer base

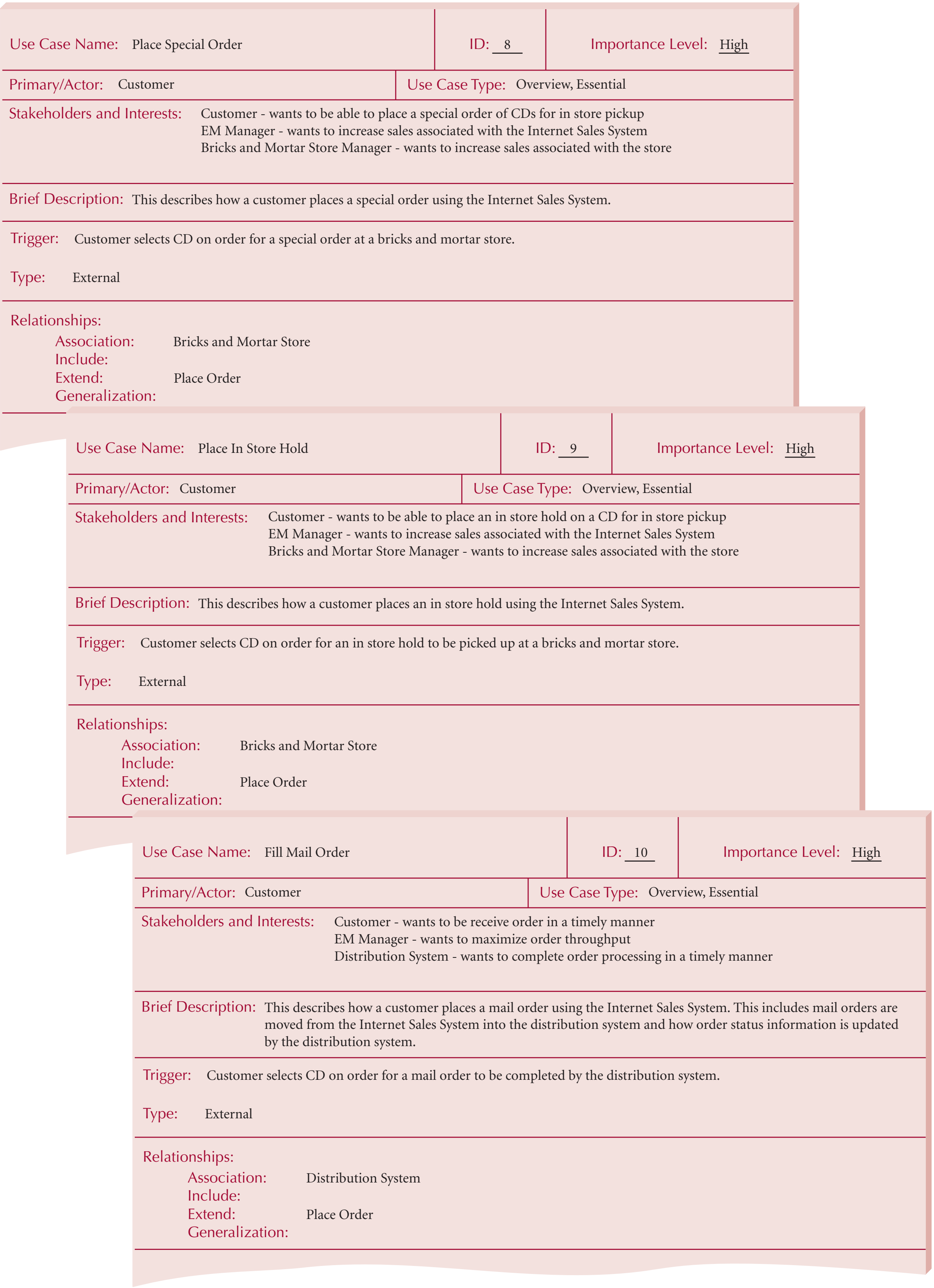
Brief Description: This describes how a new customer is added to the customer database.

Trigger: An unknown customer attempts to checkout.

Type: External

Relationships:
Association:
Include:
Extend: Checkout
Generalization:

FIGURE 4-E Overview of the Six Newly Identified Use Cases for CD Selections



Use Case Name: Place Special Order	ID: 8	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests: Customer - wants to be able to place a special order of CDs for in store pickup
EM Manager - wants to increase sales associated with the Internet Sales System
Bricks and Mortar Store Manager - wants to increase sales associated with the store

Brief Description: This describes how a customer places a special order using the Internet Sales System.

Trigger: Customer selects CD on order for a special order at a bricks and mortar store.

Type: External

Relationships:
Association: Bricks and Mortar Store
Include:
Extend: Place Order
Generalization:

Use Case Name: Place In Store Hold	ID: 9	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests: Customer - wants to be able to place an in store hold on a CD for in store pickup
EM Manager - wants to increase sales associated with the Internet Sales System
Bricks and Mortar Store Manager - wants to increase sales associated with the store

Brief Description: This describes how a customer places an in store hold using the Internet Sales System.

Trigger: Customer selects CD on order for an in store hold to be picked up at a bricks and mortar store.

Type: External

Relationships:
Association: Bricks and Mortar Store
Include:
Extend: Place Order
Generalization:

Use Case Name: Fill Mail Order	ID: 10	Importance Level: High
Primary/Actor: Customer	Use Case Type: Overview, Essential	

Stakeholders and Interests: Customer - wants to be receive order in a timely manner
EM Manager - wants to maximize order throughput
Distribution System - wants to complete order processing in a timely manner

Brief Description: This describes how a customer places a mail order using the Internet Sales System. This includes mail orders are moved from the Internet Sales System into the distribution system and how order status information is updated by the distribution system.

Trigger: Customer selects CD on order for a mail order to be completed by the distribution system.

Type: External

Relationships:
Association: Distribution System
Include:
Extend: Place Order
Generalization:

The project team then needed to gather and organize the information needed to define the Place Order use case in more detail. Specifically, they needed to begin writing the Normal Flow of Events (see Figures 4-11 and 4-14 for examples). This was done based on the results of the earlier analyses described in Chapter 3, as well as through a series of JAD meetings with the project sponsor and the key marketing department managers and staff who would ultimately operate the system.

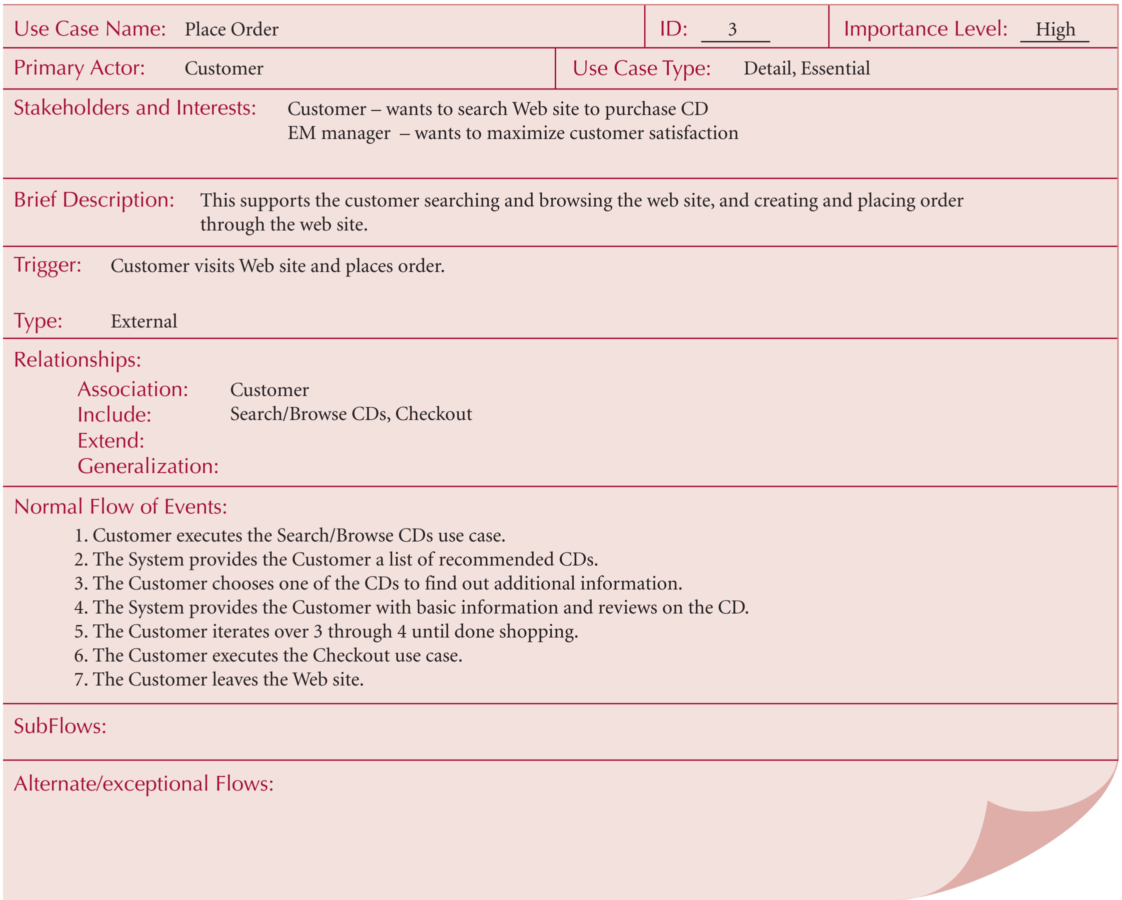
The goal at this point is to describe how the chosen use case operates: Place Order. Using the activity diagram as a starting point, Alec and the team decided to visualize placing a CD order over the Web and to think about how other electronic commerce Web sites work–that is, role play. As they role played the Place Order use case, they realized that after the customer connected to the Web site, they probably begin searching, perhaps for a specific CD, perhaps for a category of music, but in any event, they entered some information for their search. The Web site then should present a list of CDs matching their request along with some basic information about the CDs (e.g., artist, title, and price). If one of the CDs is of interest to them, they might seek more information about it, such as the list of songs, liner notes, reviews, etc. Once they found a CD they like, they will add it to their order and perhaps continue looking for more CDs. Once they are done–perhaps immediately–they will "check out" by presenting their order with information on the CDs they want and giving additional information such as mailing address, credit card, etc.

When the team wrote the use case's Normal Flow Of Events, they paid close attention to the seven guidelines described in the textbook. Alec realized that that the first step was to present the customer with the home page or a form to fill in to search for an album. Even though this is technically correct, this type of step was very "small" compared to the other steps that followed[3]. It was analogous to making the first step "hand the user a piece of paper." At this point, the team was only looking for the three to seven major steps. Based on the role playing and the application of the earlier principles (see Figure 4-12), the team successfully identified a set of steps (see Figure 4-F)

The first major step performed by the system is to respond to the customer's search inquiry, which might include a search for a specific album name or albums by a specific artist. Or, it might be the customer wanting to see all the classical or alternative CDs in stock. Or, it might be a request to see the list of special deals or CDs on "sale." In any event, the system finds all the CDs matching the request, and shows a list of CDs in response. The user will view this response, and perhaps will decide to seek more information about one or more CDs. He or she will click on it, and the system will provide additional information. Perhaps the user will also want to see any extra marketing material that is available as well. The user will then select one or more CDs for purchase, decide how to take delivery of each CD, and perhaps continue with a new search. These steps correspond to events 1 through 7 in 4-F.

The user may later make changes to the CDs selected, either by dropping some or changing the number ordered. At some point the user will "checkout" by verifying the CDs he or she has selected for purchase, and providing information about him or herself (e.g., name, mailing address, credit card). The system will calculate the total payment, and verify the credit card information with the credit card center. At this point in the transaction, the system will send an order confirmation to the customer, and the customer typically leaves the web site. Figure 4-F shows the use case at this point. Note that the Normal Flow of Events has been added to the form, but nothing else has changed. At this point in time, the Place Order use case had eight events. Given the purpose of this use case, this seemed to be a reasonable number of events.

[3] Since it is so small, it violates the fourth principle (see Figure 4-12).



Use Case Name: Place Order | ID: 3 | Importance Level: High

Primary Actor: Customer | Use Case Type: Detail, Essential

Stakeholders and Interests: Customer – wants to search Web site to purchase CD
EM manager – wants to maximize customer satisfaction

Brief Description: This supports the customer searching and browsing the web site, and creating and placing order through the web site.

Trigger: Customer visits Web site and places order.

Type: External

Relationships:
Association: Customer
Include: Search/Browse CDs, Checkout
Extend:
Generalization:

Normal Flow of Events:
1. Customer executes the Search/Browse CDs use case.
2. The System provides the Customer a list of recommended CDs.
3. The Customer chooses one of the CDs to find out additional information.
4. The System provides the Customer with basic information and reviews on the CD.
5. The Customer iterates over 3 through 4 until done shopping.
6. The Customer executes the Checkout use case.
7. The Customer leaves the Web site.

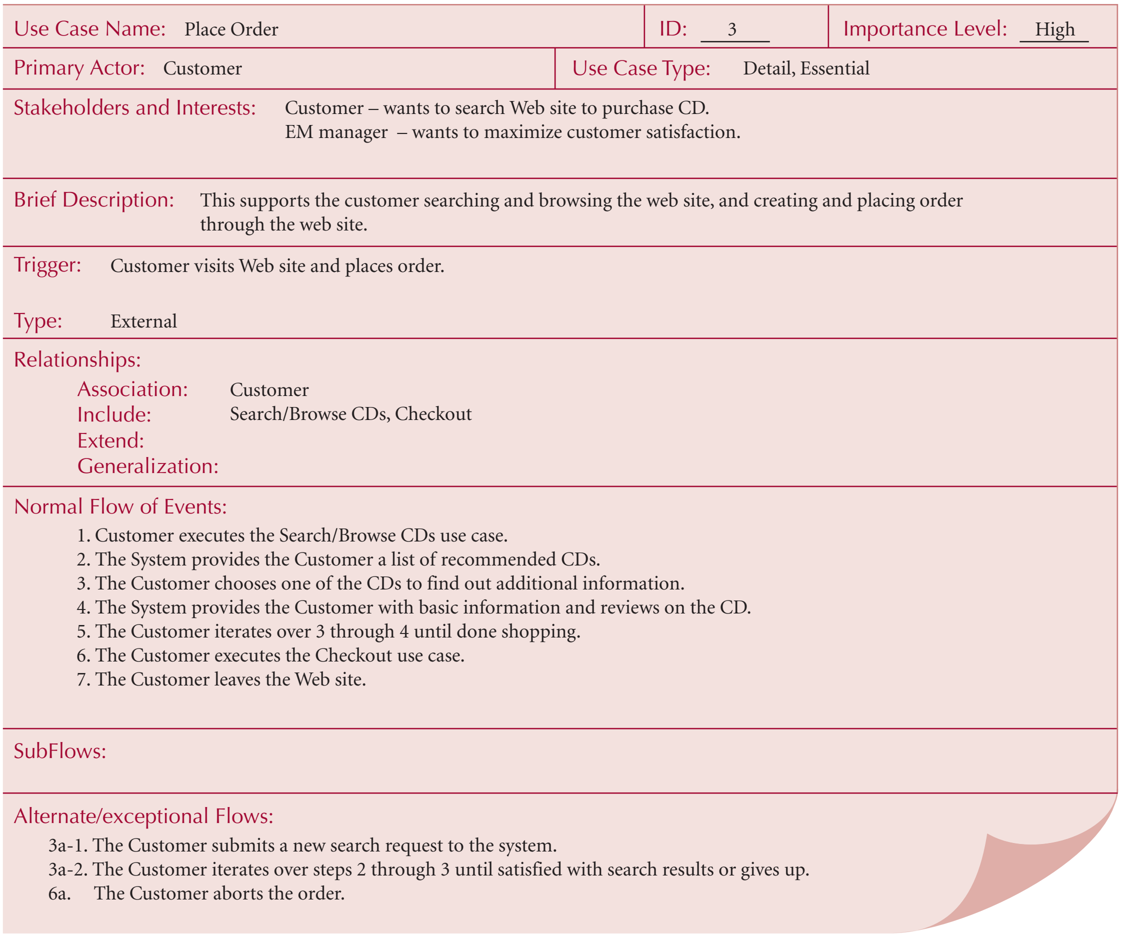
SubFlows:

Alternate/exceptional Flows:

FIGURE 4-F Normal Flow of Events of Places Order Use Case

The next step in writing a use case deals with alternate or exceptional flows. (Note: Remember the Normal Flow of Events only captures the typical set of events that end in a successful transaction.) With the Place Order use case, the development team defined success as a new order being placed. However, the team identified two sets of events that were exceptions to the normal flow. First, event 3 assumed that the list of recommended CDs were acceptable to the customer. However, as one of the team members pointed out, that is an unrealistic assumption. As such, two exceptional flows have been identified and written (3a-1 and 3a-2 in Figure 4-G) to handle this specific situation. Second, a customer may want to abort the entire order instead of going through the checkout process. In this case, exceptional flow 6a was created. Next, the team carefully reviewed the detailed, essential use case description for the Place Order use case (see Figure 4-G). The final step that Alec and the team performed was to repeat the entire process for each of the remaining use cases.

Once the team had completed evolving the overview, essential use cases to detail, essential use cases, Alec and his team had to go back through all of the functional representations of the business process to guarantee that the functional model was consistent, i.e., they had to verify and validate the business processes and functional models. Using Figure 4-16 as a



Use Case Name: Place Order	ID: 3	Importance Level: High
Primary Actor: Customer	Use Case Type: Detail, Essential	

Stakeholders and Interests: Customer – wants to search Web site to purchase CD.
EM manager – wants to maximize customer satisfaction.

Brief Description: This supports the customer searching and browsing the web site, and creating and placing order through the web site.

Trigger: Customer visits Web site and places order.

Type: External

Relationships:
Association: Customer
Include: Search/Browse CDs, Checkout
Extend:
Generalization:

Normal Flow of Events:
1. Customer executes the Search/Browse CDs use case.
2. The System provides the Customer a list of recommended CDs.
3. The Customer chooses one of the CDs to find out additional information.
4. The System provides the Customer with basic information and reviews on the CD.
5. The Customer iterates over 3 through 4 until done shopping.
6. The Customer executes the Checkout use case.
7. The Customer leaves the Web site.

SubFlows:

Alternate/exceptional Flows:
3a-1. The Customer submits a new search request to the system.
3a-2. The Customer iterates over steps 2 through 3 until satisfied with search results or gives up.
6a. The Customer aborts the order.

FIGURE 4-G Places Order Use Case with Alternate/exceptional Flows

guideline, Alec and his team checked each activity diagram against the use case descriptions and use case diagram and they checked the to be sure that each use case in the use case diagram had a completed use case description.

Finally, based on the current functional model, Alec revised the workplan again and he worked with Margaret and the folks in marketing to review the feasibility analysis and update it where appropriate. Furthermore, Alec and Margaret realized that the number of use cases had changed. Consequently, they went back and revised the project effort estimation (see Figure 4-H). At this point in time, the effort estimation went from about over 20 person months to approximately 24 months. Based on this new estimate, both Alec and Margaret were glad that Alec had originally doubled the estimation of 12 months. As Alec told Margaret, “sometimes it is better to be lucky, then to be good.” Given the fact that the team has never built this type of system before, they were indeed lucky. In the future, Alec and Margaret realized that they would have to do a better job at estimating. But, with experience they felt that they would indeed be better in being able to estimate the number and complexity of the actors and use cases in this type of system. And that this would improve their estimation ability without having to revert to Alec’s let’s double the estimate approach.

Unadjusted Actor Weighting Table:

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	2	2
Average	External System using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	1	2
Complex	Human	3	2	6
			Unadjusted Actor Weight Total (UAW)	**10**

Unadjusted Use Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	3	15
Average	4–7 transactions	10	1	10
Complex	>7 transactions	15	5	75
			Unadjusted Use Case Weight Total (UUCW)	**100**

Unadjusted use case points (UUCP) = UAW + UUCW 110 = 10 + 100

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
T1	Distributed system	2.0	5	10.0	
T2	Response time or throughput performance objectives	1.0	5	5.0	
T3	End-user online efficiency	1.0	5	5.0	
T4	Complex internal processing	1.0	4	4.0	
T5	Reusability of code	1.0	3	3.0	
T6	Easy to install	0.5	3	1.5	
T7	Ease of use	0.5	5	2.5	
T8	Portability	2.0	4	8.0	
T9	Ease of change	1.0	3	3.0	
T10	Concurrency	1.0	3	3.0	
T11	Special security objectives included	1.0	5	5.0	
T12	Direct access for third parties	1.0	5	5.0	
T13	Special User training required	1.0	3	3.0	
			Technical Factor Value (TFactor)	**58.0**	

Technical complexity factor (TCF) = 0.6 + (0.01 * TFactor) 1.18 = 0.6 + (0.01 * 58)

Environmental Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	1	1.5	
E2	Application experience	0.5	2	1.0	
E3	Object-oriented experience	1.0	0	0.0	
E4	Lead analyst capability	0.5	3	1.5	
E5	Motivation	1.0	4	4.0	
E6	Requirements stability	2.0	4	8.0	
E7	Part time staff	−1.0	0	0.0	
E8	Difficulty of programming language	−1.0	4	−4.0	
			Environmental Factor Value (EFactor)	**12.0**	

Environmental factor (EF) = 1.4 + (−0.03 * EFactor) 1.04 = 1.4 + (−.03 * 12)

Adjusted use case points (UCP) = UUCP *TCF *ECF 134.992 = 110 * 1.18 * 1.04

Person hours multiplier (PHM) PHM = 28

Person hours = UPC * PHM 3,779.776 = 134.992 * 28

FIGURE 4-H Use-Case Points Estimation for the Internet Sales Systems

CHAPTER 5: STRUCTURAL MODELING

In the previous chapter's installment of the CD Selections case, we saw how Alec, Margaret, and the team worked through building functional models of the business processes (see Figures 4-A through 4-G) contained in their evolving Web-based solution. In this chapter, we introduced how structural models using CRC cards and class and object diagrams could be created, verified, and validated. In this installment of the CD Selections case, we see how Alec and Margaret work through creating, verifying, and validating the structural models of the Web-based solution based on the one completed detailed, essential use case: Place Order (see Figure 4-G). Even though we are using just one of the use cases for our example, you should remember that to create a complete structural model all use cases should be used.

Create CRC Cards

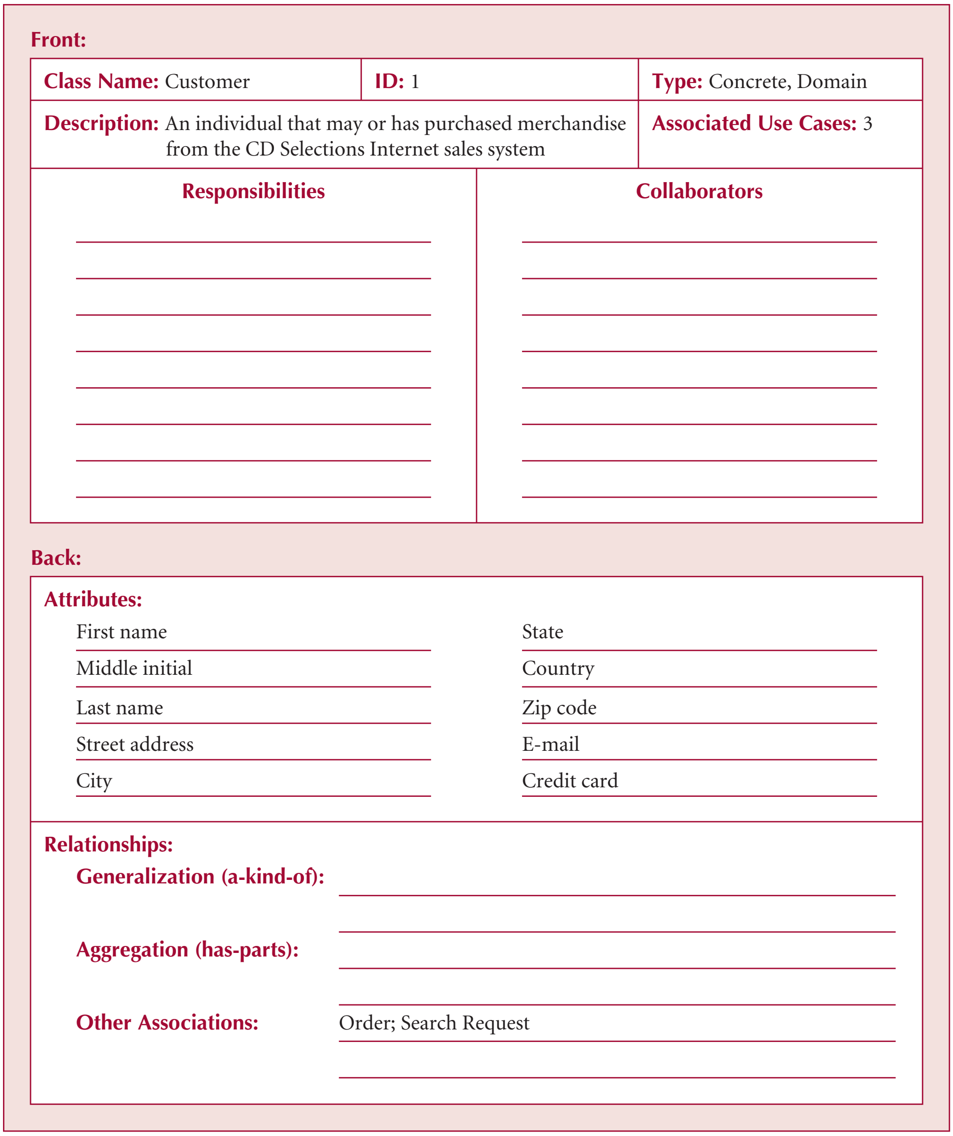
The first step Alec and the team was to create the set of CRC cards by performing textual analysis on the use cases. To begin with, Alec chose the Place Order use case (see Figure 4-G). He and his team then used the textual analysis rules (see Figure 5-1) to identify the candidate classes, attributes, operations, and relationships. Using these rules on the Normal Flow of Events, they identified Customer, Search Request, CD, CD List, and Review as candidate classes. They uncovered three different types of search requests: Title Search, Artist Search, and Category Search. By applying the textual analysis rules to the Brief Description, an additional candidate class was discovered: Order. By reviewing the verbs contained in this use case, they saw that a Customer places an Order and that a Customer makes a Search Request.

To be as thorough as possible, Alec and his team also reviewed the original requirements used to create the use case. The original requirements are contained in Figure 3-A. After reviewing this information, they identified a set of attributes for the Customer (name, address, e-mail, and credit card) and Order (CDs to purchase and quantity) classes and uncovered additional candidate classes: CD Categories and Credit Card Center. Furthermore, they realized that the Category Search class used the CD Categories class. Finally, they also identified three subclasses of CD Categories: Rock, Jazz, and Classical. Alec's goal, at this point in time, was to be as complete as possible. As such, he realized that they may have identified many candidate classes, attributes, operations, and relationships that may not be included in the final structural model. Regardless, the current list of candidate classes included: Customer, Order, Search Request, Title Search, Artist Search, Category Search, CD, CD List, Review, CD Categories, Rock, Jazz, Classical, and Credit Card Center.

Review CRC Cards

The second step for Alec and his team was to carefully review the current set of CRC cards to determine if they had missed any potential candidate classes, attributes, operations, or relationships. Furthermore, the team then used the current candidate classes and a common object list as input to brainstorming additional candidate classes, attributes, operations, and relationships. For example, Alec asked the team members to take a minute and think about what information they would like to keep about CDs. The information that they thought of was a set of attributes, for example, title, artist, quantity on hand, price, and category.

He then asked them to take another minute and think about the information that they should store about orders and an order's responsibilities. The responsibilities they identified were a set of operations, including calculate tax, calculate extension price, calculate shipping, and calculate total. Currently, the attributes (CDs to purchase and quantity) of Order implied that a customer should be allowed to order multiple copies of the same CD and allow different CDs to be ordered on the same order. However, the current structural model did not allow this. As such, they created a new class that was



Front:

Class Name: Customer	**ID:** 1	**Type:** Concrete, Domain
Description: An individual that may or has purchased merchandise from the CD Selections Internet sales system		**Associated Use Cases:** 3

Responsibilities	**Collaborators**

Back:

Attributes:

First name	State
Middle initial	Country
Last name	Zip code
Street address	E-mail
City	Credit card

Relationships:

Generalization (a-kind-of):

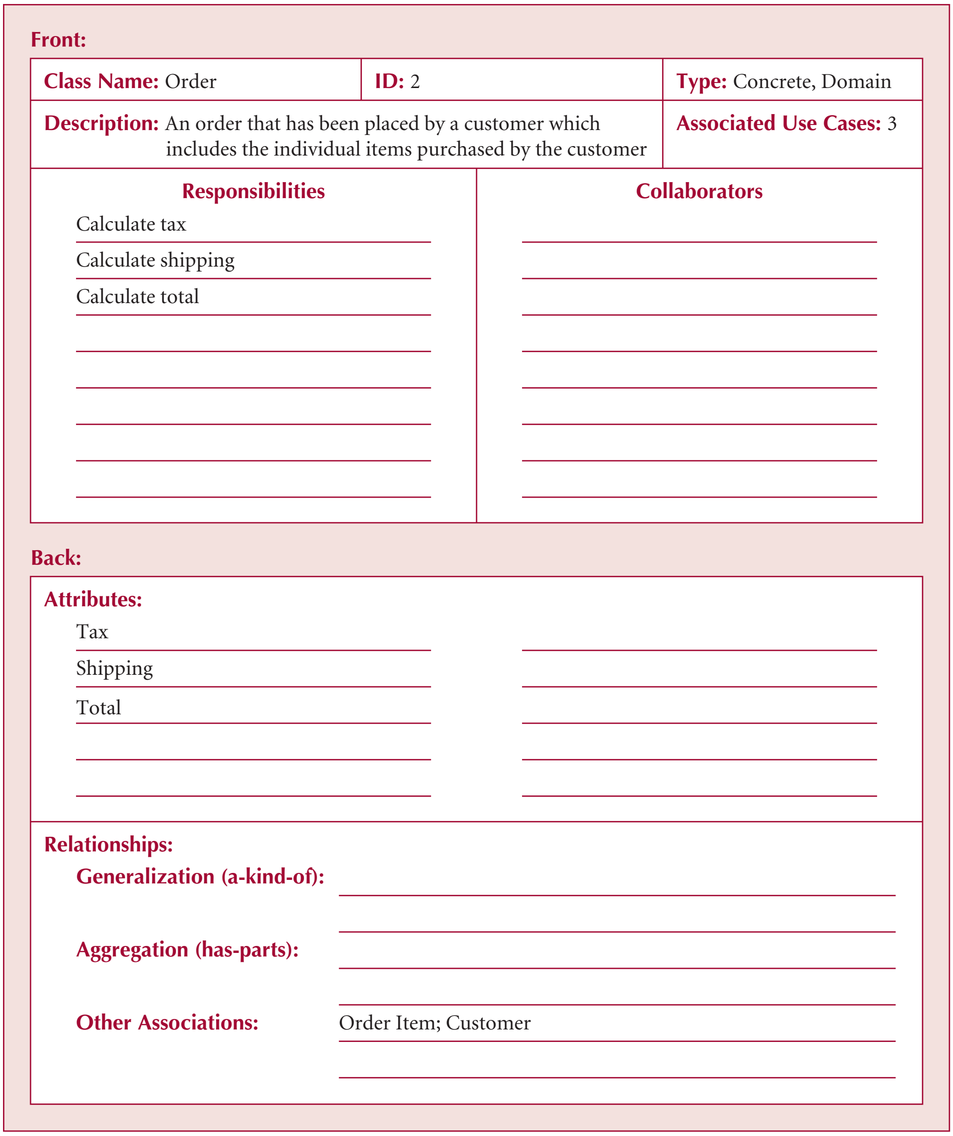
Aggregation (has-parts):

Other Associations: Order; Search Request

FIGURE 5-A Customer Class CRC Card

associated with both the Order class and the CD class: Order Line Item. This new class only had one attribute, quantity, but it had two relationships: one with Order and the other with CD.

When they reviewed the Customer class, they decided that the name and address attributes needed to be expanded; name should become last name, first name, and middle initial, and address should become street address, city, state, country, and zip code. The updated Customer class and Order class CRC cards are shown in Figures 5-A and 5-B, respectively. Once they completed the CRC cards for the Customer and Order class, the team completed the CRC cards for the remaining candidate classes: Order Line Item, Search Request, Title Search, Artist Search, Category Search, CD, CD List, Review, CD Categories, Rock, Jazz, Classical, and Credit Card Center.



Front:

Class Name: Order	**ID:** 2	**Type:** Concrete, Domain
Description: An order that has been placed by a customer which includes the individual items purchased by the customer		**Associated Use Cases:** 3

Responsibilities	Collaborators
Calculate tax	
Calculate shipping	
Calculate total	

Back:

Attributes:

Tax

Shipping

Total

Relationships:

Generalization (a-kind-of):

Aggregation (has-parts):

Other Associations: Order Item; Customer

FIGURE 5-B
Order Class CRC card

Role Play the CRC Cards

The third step was to role-play the classes recorded on the CRC cards. The purpose of this step was to validate the current state of the evolving structural model. Alec handed out the CRC cards to different members of his team. Using the CRC cards, they began executing the different use cases (see Figures 4-B, 4-E, and 4-G), one at a time, to see if the current structural model could support each use case or whether the use case caused the "system" to crash. Anytime the "system" crashed, there was something missing: a class, an attribute, a relationship, or an operation. They would then add the missing information to the structural model and try executing the use case again.

First, Alec and the team decided that the customer had requested the system to perform a search for all of the CDs associated with a specific artist. Based on the current CRC cards, the team felt that the system would produce an accurate list of CDs. They then tried to ask the system for a set of reviews of the CD. At this point in the exercise, the system crashed. The CRC cards did not have the Review class associated with the CD class. Therefore, there was no way to retrieve the requested information. This observation raised another question. Was there other marketing information that should be made available to the customer, for example, artist information and sample clips?

Next, the team realized that vendor information should be a separate class that was associated with a CD rather than an additional attribute of a CD. This was because vendors had additional information and operations themselves. If the team had modeled the vendor information as an attribute of CD, then the additional information and operations would have been lost. They continued role-playing each of the use cases until they were comfortable with the structural model's ability to support each and every one. Based on the above, the team created CRC cards for the newly identified classes: Mkt Info, Artist Info, Sample Clip, and Vendor.

Create Class Diagram

The fourth step was to create the class diagram from the CRC cards. Figure 5-C shows the current state of the evolving structural model as depicted in a class diagram based on the Places Order use case.

Review Class Diagram

The fifth step was to review the structural model for missing and/or unnecessary classes, attributes, operations, and relationships. At this point, the team challenged all components of the model (class, attribute, relationship, or operation) that did not seem to be adding anything useful to the model. If a component could not be justified sufficiently, then they removed it from the structural model. By carefully reviewing the current state of the structural model, they were able to challenge over a third of the classes contained in the class diagram (see Figure 5-C). It seemed that the CD categories, and their subclasses, were not really necessary. There were no attributes or operations for these classes. As such, the idea of CD categories was modeled as an attribute of a CD. The category attribute for the CD class was previously uncovered during the brainstorming step. Also, upon further review of the Search Request class and its subclasses, it was decided that the subclasses were really nothing more than a set of operations of the Search Request class. This was an example of process decomposition creeping into the modeling process. From an object-oriented perspective, we must always be careful to not allow this to occur. However, during the previous steps in the modeling process, Alec wanted to include as much information as possible in the model. He felt that it was more beneficial to remove this type of information after it had crept into the model than to take a chance on not capturing the information required to solve the problem.

Incorporate Patterns

The sixth step was to incorporate any useful patterns into the structural model (see Figure 5-5). Two patterns that could be useful in this case are the Transaction pattern and the Party pattern (see Figure 5-3). Using these two patterns, Alec and his team uncovered two subclasses of the Customer class, Person and Organization, and they raised the issue of the Place class. However, in this case, all transactions would be taking place in the same place, cyberspace, and therefore the team decided not to include it.

Review the Model

The seventh and final step was to carefully review the structural model. In this case, Alec and the team had to make sure that the CRC cards and the class diagram were in total agreement. Figure 5-D shows the Places Order use case view of the structural model as portrayed in a class diagram developed by Alec and his team. This version of the class diagram incorporates all of the modification described previously.

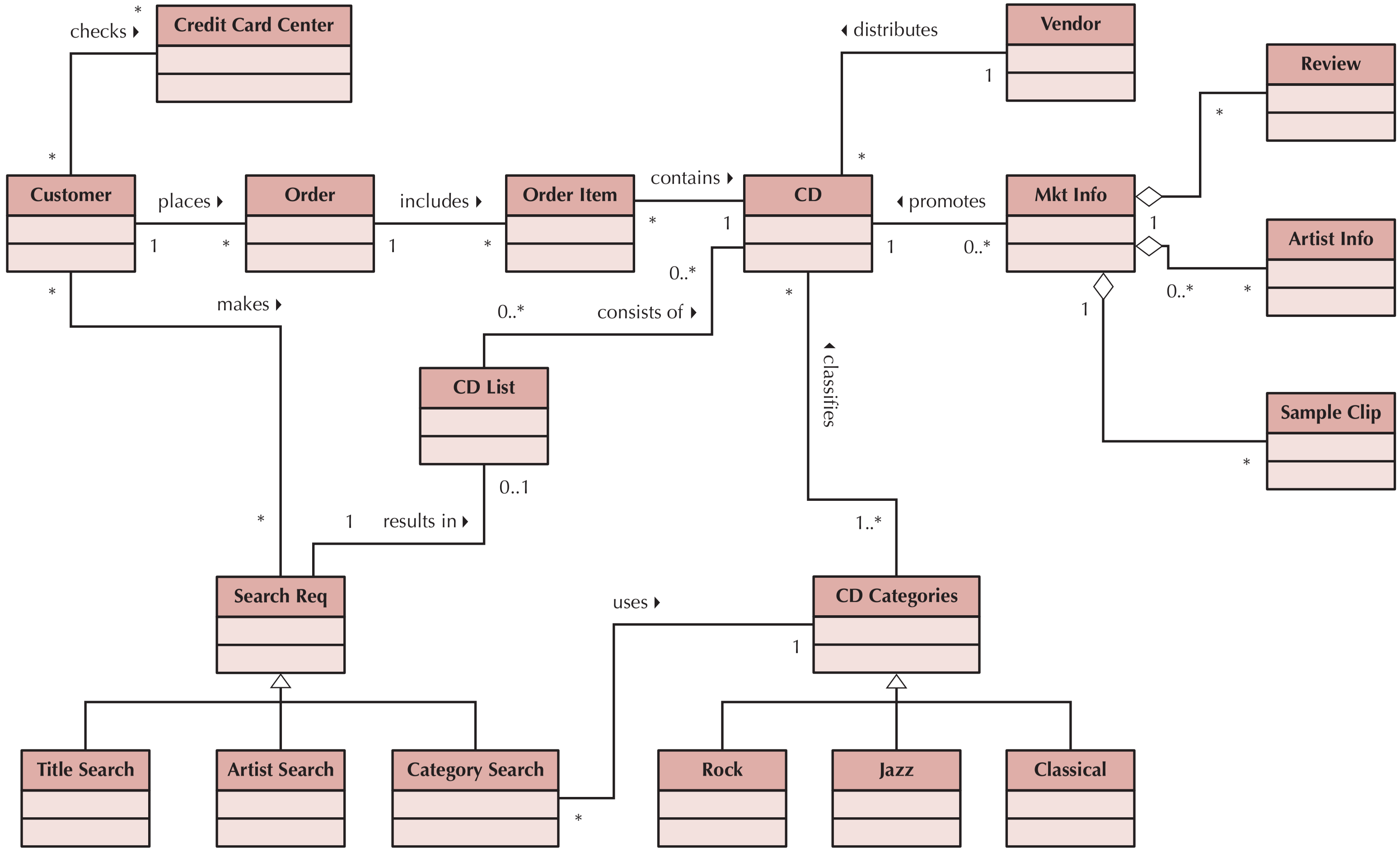


FIGURE 5-C Preliminary CD Selections Internet Sales System Class Diagram (Places Order Use Case View)

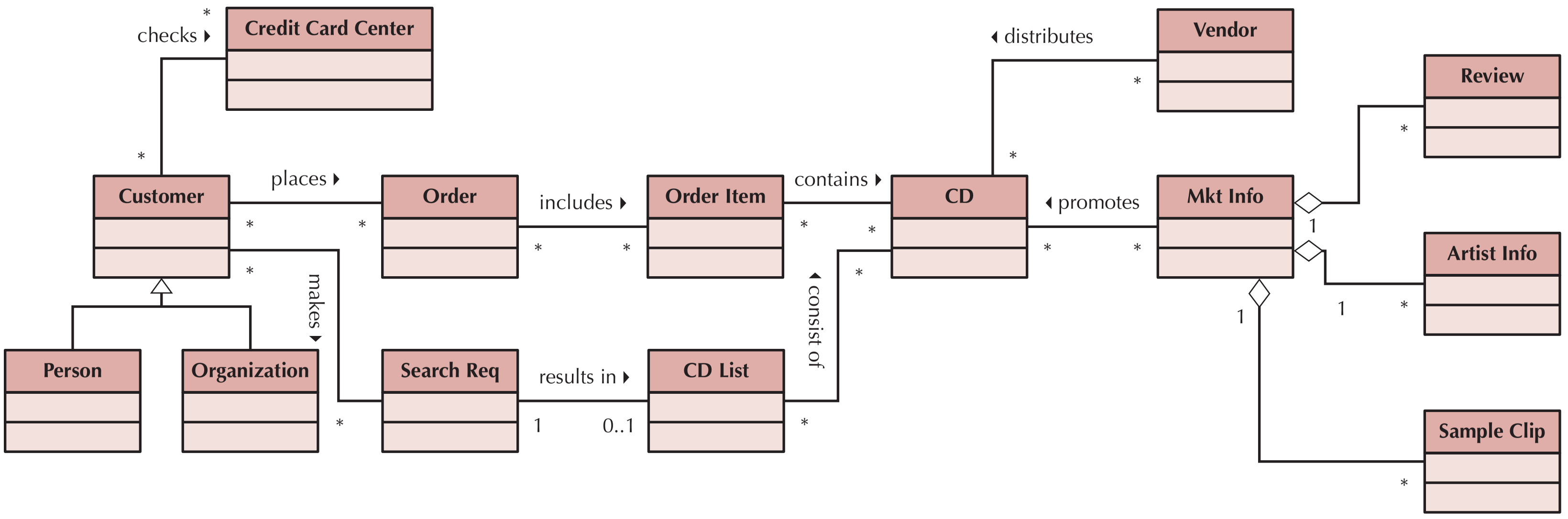


FIGURE 5-D CD Selections Internet Sales System Class Diagram (Places Order Use Case View)

CHAPTER 6: BEHAVIORAL MODELING

Since Alec, Margaret, and the team have now completed rough functional and structural models for their evolving Web-based solution, they have decided that it was time to move on and begin to create the behavioral models. Alec understood that in some ways, the behavioral models allow them to complete their understanding of the problem. In this installment of the CD Selections case, the team creates sequence diagrams, communication diagrams, behavioral state machines, and a CRUDE matrix. As in the previous installments, we see how the team goes about creating, verifying, and validating the behavioral models of the Web-based system they hope to implement. As in Chapter 5, you should realize that the team created behavioral models for all of the use cases and classes in the evolving system description. However, in the sections that follow, we only see the models associated with the Place Order use case and the Order class. The sections are organized in the same manner as the chapter: sequence diagrams, communication diagrams, behavioral state machines, and CRUDE matrix.

Sequence Diagrams

To begin with Alec decided that the team should follow the guidelines for creating sequence diagrams listed in Figure 6-4. Next, Alec decided that the team should follow six steps to create a sequence diagram described in the textbook. Consequently, Alec first needed to determine the context of the sequence diagram. He decided to use a scenario[4] from the Place Order use case that was created with the business process and functional modeling installment in Chapter 4 and illustrated in Figure 4-G. (Refer to the original use case for the details.) Figure 6-A lists the Normal Flow of Events that contains the scenario that this sequence diagram describes.

The second step was to identify the actors and objects that participated in the scenario being modeled. The classes associated with the Place Order use case are shown in Figure 5-D. For example, the classes used for the Place Order use case include Customer, CD, Marketing Information, Credit Card Center, Order, Order Item, Vendor, Search Request, CD List, Review, Artist Information, Sample Clip, Person, and Organization.

During the role playing of the CRC Cards, Anne, one of the analysts assigned to the CD Selections Internet System Development Team, asked whether a Shopping Cart class should be included. She stated that every Internet sales site she had been to had a shopping cart that was used to build an order. However, the instance of the Shopping Cart class only existed until either an order was placed or the shopping cart was emptied. Based on this obvious oversight, both the Place Order use case and the class diagram will have to be modified. Brian, another analyst, pointed out that the CDs themselves were

Normal Flow of Events:

1. Customer executes the Search/Browse CDs use case.
2. The System provides the Customer a list of recommended CDs.
3. The Customer chooses one of the CDs to find out additional information.
4. The System provides the Customer with basic information and reviews on the CD.
5. The Customer iterates over 3 through 4 until done shopping.
6. The Customer executes the Checkout use case.
7. The Customer leaves the Web site.

FIGURE 6-A
Normal Flow of Events of the Places Order Use Case

[4] Remember, as stated previously, a scenario is a single executable path through a use case.

going to have to be associated with some type of searchable storage. Otherwise, it would be impossible to find the CD in which the customer was interested. However, Alec decided that the CD List class would suffice for both the searchable storage and a temporary instance that would be created as a result of a search request. Alec pointed out to the team that this process was fairly typical in object-oriented development. The more the development team learned about the requirements, the more the models (functional, structural, and behavioral) would evolve. Alec reminded them that the important thing to remember was that an object-oriented development process was incremental and it iterated over all of the models. As such, as they understood the problem better, the team would most likely have to make changes to the functional and structural models already completed.

Based on the team's current understanding of the Place Order use case, they decided that instances of the Search Request, CD List, CD, Marketing Material, Customer, Review, Artist Information, Sample Clip, and Shopping Cart classes would be required to describe this scenario. Furthermore, they realized that the Customer actor interacted with the scenario. To complete this step, the team laid out the objects on the sequence diagram by drawing them, from left to right, across the diagram.

The third step was to set the lifeline for each object. To do this, they drew a vertical dotted line below each of the objects (aSR, aCDL, CDs, aCD, MI, aR, AI, SC, and anOrder) and the actor (aCustomer). They placed an X at the bottom of the lifelines for aCDL and aSC since they "go away" at the end of this process.

The fourth step was to add the messages to the diagram. By examining the steps in Figure 6-A, the team was able to determine the way in which the messages should be added to the diagram. Figure 6-B shows the diagram they created. Notice how they did not include messages back to Customer in response to "create SR" and "add CD." In these cases, the team assumed that aCustomer would receive response messages about the requested CD and inserted CD, respectively.

The fifth step was to add the execution occurrence to each object's and actor's lifeline. This was done by drawing a narrow rectangle box over top of the lifelines to represent when the objects (actors) are sending and receiving messages (e.g., in Figure 6-B aCustomer is active during the entire process, while aSR is only active at the beginning of the process (the top of the diagram)).

Finally, the CD Selections team validated the diagram by ensuring that the diagram accurately and completely represented the scenario of the Place Order use case being modeled. Figure 6-B portrays their completed sequence diagram.

Communication Diagrams

Brian, one of the analysts, pointed out to the team that sequence diagrams and communication diagrams essentially modeled the same things. As such, he felt that it was not worth the time for the team to do both. And since they had already completed the sequence diagrams, he really did not want to do the communication diagrams also. However, even though the diagrams are very similar in what they portray, Alec decided that it would be worth the team's time to build both. He felt that it could be possible that the different formats of the diagrams might uncover additional requirements. As such, the team also created communication diagrams.

Alec chose to create the communication diagrams by following the steps to create communication diagrams described in the textbook. Like creating sequence diagrams, the first step in the process was to determine the context of the communication diagram. Alec chose to start with the same scenario of the Place Order use case that he and the team had used previously when they created the sequence diagrams (see Figure 6-B).

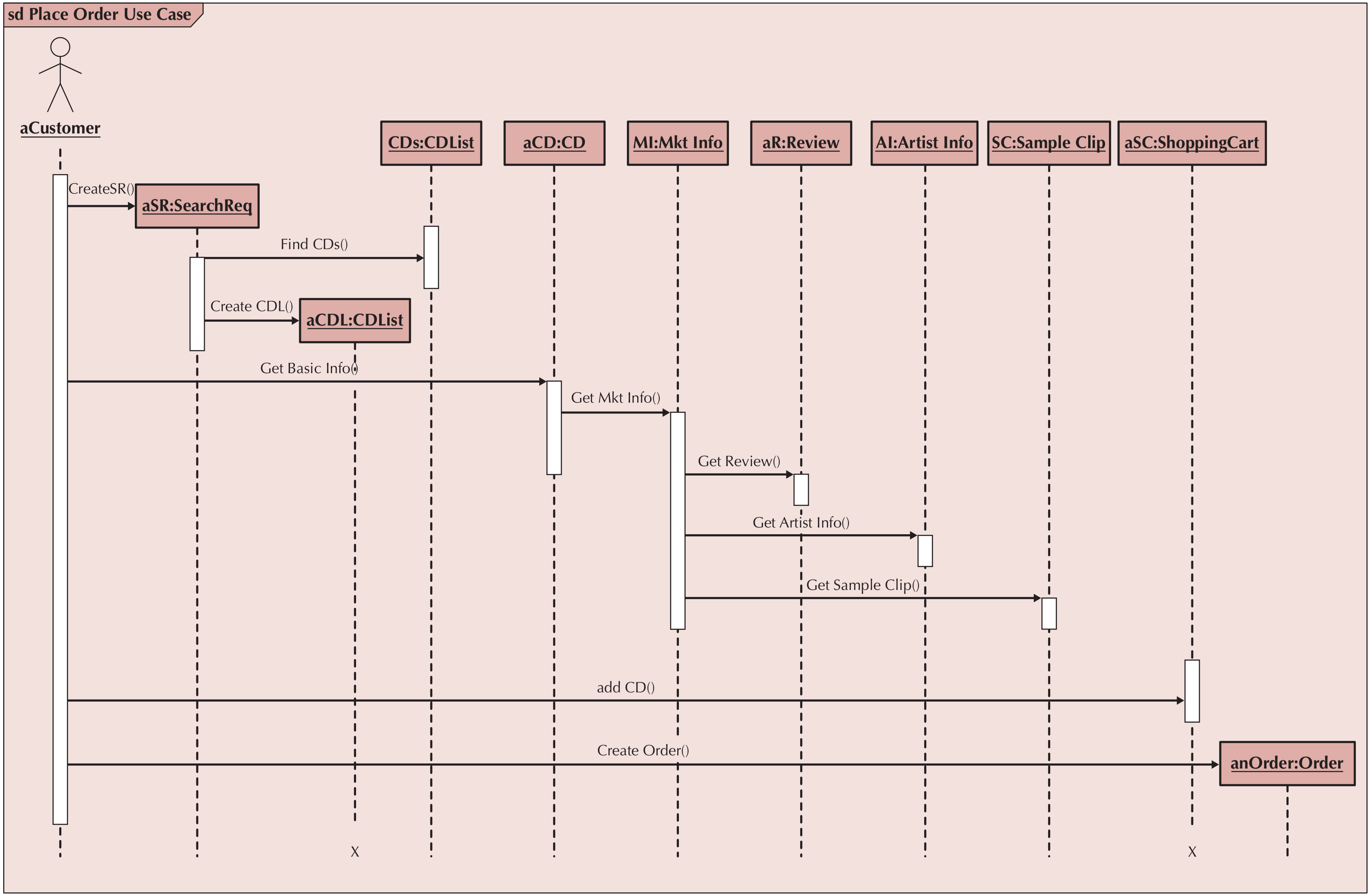


FIGURE 6-B Sequence Diagram for the Places Order Use Case

By executing the second step, the CD Selections team again identified the objects and the associations that link the objects together. Since they are using the same scenario as they did in the previously described sequence diagram, instances of the Search Request, CD List, CD, Marketing Material, Customer, Review, Artist Information, Sample Clip, and Shopping Cart classes should be the ones included. Also, the since the Customer actor interacts with the scenario, it should also be included. Furthermore, the team identified the associations between the objects (e.g., the instances of CD are associated with instances of Mkt Info, which, in turn, are associated with instances of Review, Artist Info, and Sample Clip).

During the third step, the team placed the objects on the diagram based on the associations that they have with the other objects in the collaboration. This was done to increase the readability, and hence, the understandability of the diagram (see Figure 6-C).

During the fourth step, the CD Selections team added the messages to the associations between the objects. For example, in Figure 6-C, the Create SR() message is the first message sent and the FindCDs() message is the second message sent. The aCustomer actor sends the Create SR() message to the aSR object, and the aSR object sends the FindCDs() message to the CDs object.

Finally, the CD Selections team executed the fifth and final step: validating the diagram. They accomplished this by ensuring that the scenario of the Place Order use case was accurately and completely represented by the diagram. See Figure 6-C for the completed communication diagram for this particular scenario of the Place Order use case. Furthermore, they compared the previously created sequence diagram (see Figure 6-B) with the communication diagram (see Figure 6-C) to ensure that that both diagrams were equivalent. The only difference between the two diagrams was the ease to portray the time ordering of the messages in the sequence diagram to represent how the objects interacted with each other in the communication diagram.

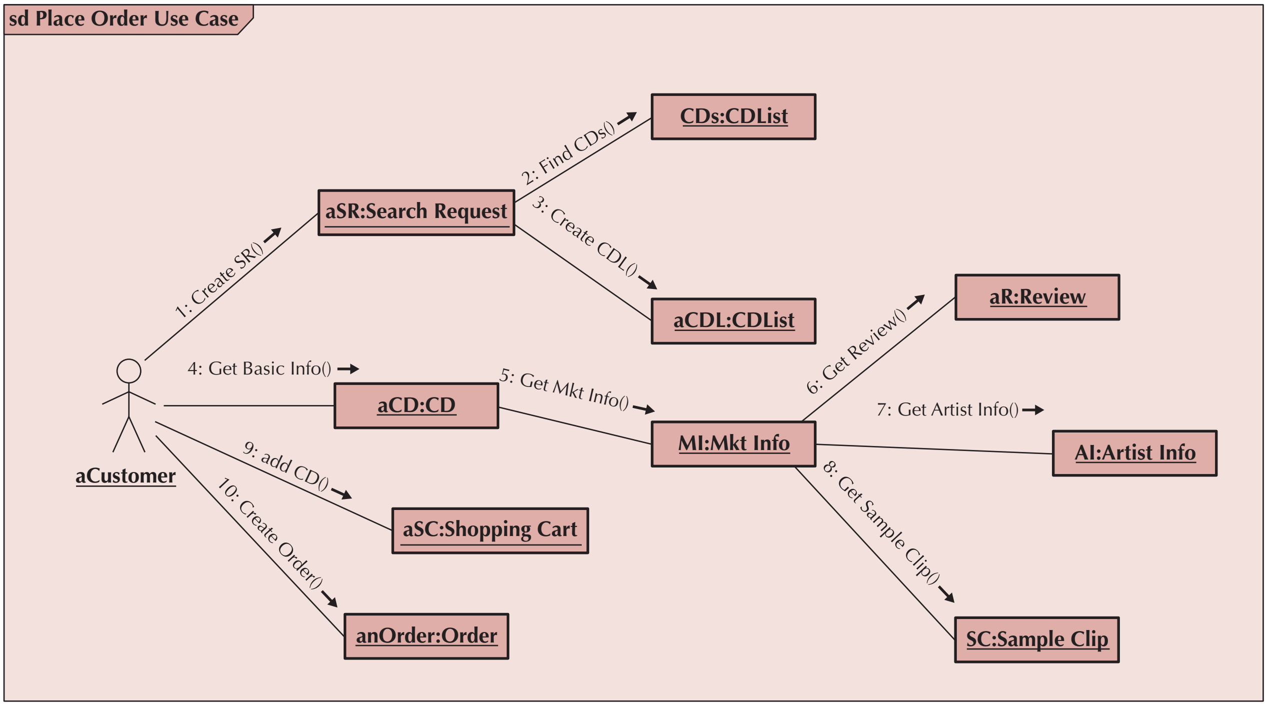


FIGURE 6-C Communication Diagram for the Places Order Use Case

Behavioral State Machines

As in the previous example diagrams, we focus our attention only on the Place Order use case. Alec decided to follow the guidelines for creating a behavioral state machine (see Figure 6-10) and to follow the five steps for creating behavioral state machines described in the textbook. Like the earlier diagrams, the first step was to determine the context for the behavioral state machine to be drawn. Upon reviewing the objects involved in the scenario described by the sequence diagram (see Figure 6-A) and the communication diagram (see Figure 6-B) Alec decided that the team should focus on the Order class.

The second step was to identify the various states that an order will go through during its lifetime. To enable the discovery of the initial, final, and stable states of an order, Alec and the development team interviewed a customer representative that dealt with processing customer orders on a regular basis. Based on this interview, the team uncovered the life of an order (see Figure 6-D) from start to finish, from an order’s perspective.

The third step is to determine the sequence of the states that an order object will pass through during its lifetime. Based on the order’s lifecycle portrayed in Figure 6-D, the team identified and laid out the states of the order on the behavioral state machine.

Next, the team identified the events, actions, and guard conditions associated with the transitions between the states of the order. For example, the event “Order is created” moves the order from the “initial” state to the “In process” state (see Figure 6-E). During the “Processing” state, a credit authorization is requested. The guard condition “Authorization = Approved″ prevents the order to move from the ″Processing″ state to the ″Placed″ state unless the credit authorization has been approved. Also, the guard condition ″Authorization = Denied″ prevents the order to move from the ″Processing″ state to the ″Denied″ state unless the credit authorization has been denied. As such, between the two guard conditions, the order is stuck in the processing state until the credit authorization has been either approved or denied.

The team finally validated the Order’s behavioral state machine by ensuring that each state was reachable and that it was possible to leave all states except for the final states. Furthermore, the team made sure that all states and transitions for the order had been modeled. At this point in time, one of the analysts on the team, Brian, suggested that possibly there were multiple types of orders being described in the behavioral state machine. Specifically, he thought that there were denied and accepted orders. Based on this discovery, he suggested that two new classes, for each subtype of order, be created. However, upon further review by the entire team, it was decided that adding these classes to the class diagram and modifying all of the other diagrams to reflect this change would not add anything to the understanding of the problem. Therefore, it was decided not to add the classes. However, in many cases, modeling the states that an object will go through during its lifetime may in fact uncover additional useful subclasses. Figure 6-E illustrates the behavioral state machine that the CD Selections team created for an order object[5].

1. The customer creates an order on the Web.
2. The customer submits the order once he or she is finished.
3. The credit authorization needs to be approved for the order to be accepted.
4. If denied, the order is returned to the customer for changes or deletion.
5. If accepted, the order is placed.
6. The order is shipped to the customer.
7. The customer receives the order.
8. The order is closed.

FIGURE 6-D
The Life of an Order

[5] If the development team had more carefully read our textbook, they would have seen that they could have reused the Order behavioral state machine in Figure 6-17.

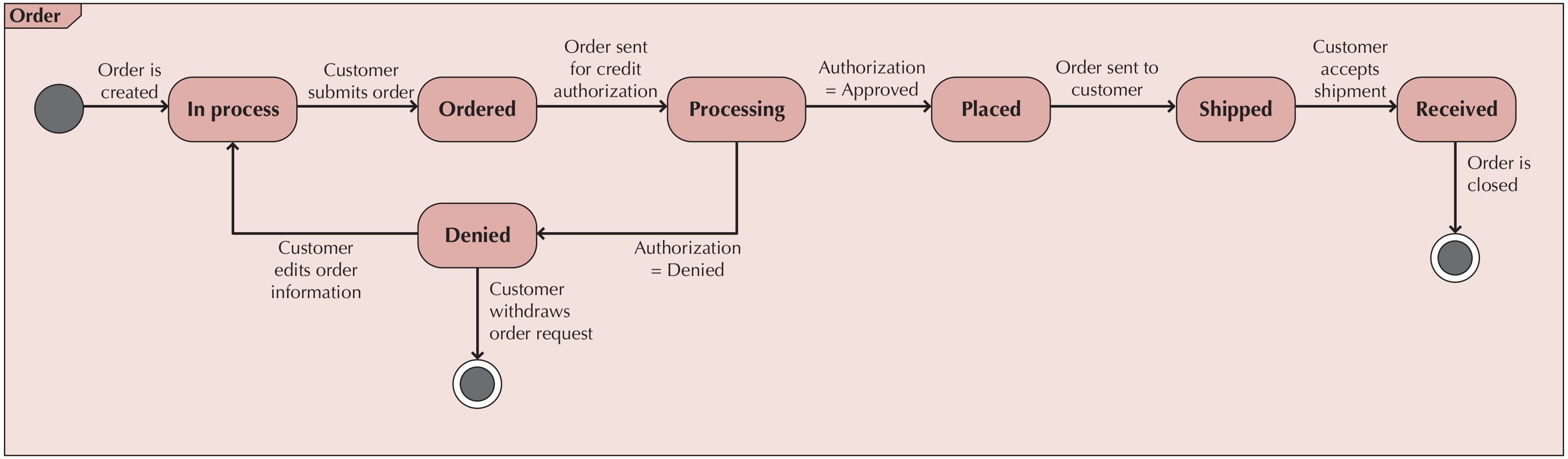


FIGURE 6-E Behavioral State Machine for the Order class

CRUDE Matrix

As an attempt to tie the functional, structural, and behavioral models together, Alec decided to create a CRUDE matrix. To accomplish this, Alec assigned Anne the task of creating the matrix. As in the previous examples, we have limited this example to the Place Order use case.

To begin with, Anne created a class-by-class matrix. She then placed a (C)reate, (R)ead, (U)pdate, (D)elete, or a (E)xecute in each cell of the matrix that represented an interaction between instances of the classes. For example in Figures 6-B and 6-C, an instance of SearchReq created an instance of CDList. Also, in Figures 6-B and 6-C, an instance of CD references an instance of MktInfo. In this case, an "R" was placed in the CD:MktInfo cell. Figure 6-F shows the CRUDE matrix that Anne created based on the Place Order use case.

Verifying and Validating the Behavioral Model

Once the team had completed all of the sequence diagrams, communication diagrams, behavioral state machines, and the CRUDE matrix, Alec had the team verify and validate the behavioral model. To accomplish this, Alec had the team use Figure 6-24 to identify where all of the common aspects of the different representations could be easily identified. For example, Alec pointed out that messages were not only contained in communication and sequence diagrams, but that they were also associated with the transitions in a behavioral state machines and the cell entries of the CRUDE matrix. Even though the team felt confident that the different representations were all consistent with each other, based on his past experience with teams that had short cut the verification and validation process, Alec insisted that all of the representations had to be verified and validated. Furthermore, Alec reminded them that they still needed to go back and modify the class diagram (see Figure 5-D) to include the Shopping Cart class. Consequently, the team still had quite a bit of work to do.

	Customer	SearchReq	CDList	CD	Mkt Info	Review	Artist Info	Sample Clip	Shopping Cart	Order
Customer		RU							U	C
SearchReq			CR							
CDList										
CD					R					
Mkt Info						U	U	U		
Review										
Artist Info										
Sample Clip										
Shopping Cart										
Order										

FIGURE 6-F CRUDE Matrix for the Places Order Use Case

CHAPTER 7: MOVING ON TO DESIGN

In the previous installments of the CD Selections case, we saw how Alec, Margaret, and the team had identified the functional and nonfunctional requirements (see Figure 3-A) and had completed the functional (see Figure 4-B, 4C, 4D, 4E, and 4G), structural (see Figures 5-A, 5-B, and 5-D), and behavioral models (Figure 6-B, 6-C, 6-E, and 6-F) of their evolving Web-based solution. However, before they can move into design, they realize that they needed to logically partition the model of the problem-domain. To do this, they have decided to create a package diagram that will represent an overview of the analysis models for their evolving system. In this section of the case, we see how Alec and his team prepared to move from an analysis, or problem-domain, orientation to a design, or solution-domain, orientation. Below, we will see that to get ready for this transition, Alec and his team first create a package diagram to partition the problem-domain layer. Next, they go through a verification and validation of all of the analysis models, and finally, they choose a design strategy to develop the actual design. As in the previous installments of the case, we only deal with the Place Order use case. However, you should remember that object-oriented systems development is holistic. Therefore, to be complete, Alec and his team had to complete the analysis models for all of the use cases associated with the case.

Packages and Package Diagrams

At this point in the development of the Internet Sales System for CD Selections, Alec wants to explicitly partition the evolving system. To do this, Alec decided to use packages to represent both the layers and partitions in each layer. Once he made this decision, he chose to follow the guidelines in Figure 7-21 and the steps for identifying packages and building package diagrams in the textbook. Since, at this point in the development, the team has only been focusing on analysis models, Alec decided that the team should only concentrate on identifying potential partitions on the Problem Domain Layer.

The second step, cluster classes together, was accomplished by reviewing the relationships among the different classes (see Figures 5-D, 6-C, and 6-F). Through this review process, the team saw that there were generalization, aggregation, various associations, and message sending relationships. They also saw the entries in the CRUDE matrix. Since they understood that classes in a generalization hierarchy should be kept together, they clustered the Customer, Person, and Organization classes together to form a partition. Brian pointed out that it is also preferred to keep classes together that participate in aggregation relationships. Based on aggregation relationships, they clustered the Mkt Info, Review, Artist Info, and Sample Clip classes together in a partition. Based on the association relationship and the message-sending pattern of activity between the CD and Mkt Info classes, Anne suggested that they should be in the same partition. Furthermore, since the Vendor class was only related to the CD class, Alec suggested that they be placed in the same partition. Finally, the development team decided to place the Order and Order Item classes together and they decided that the “temporary” classes used during the shopping process should be grouped together: Search Req, CD List, Shopping Cart, and SC Item classes together in their own partitions. If you remember, in the previous installation, Anne had recommended including a Shopping Cart class. Once they did that, they realized that they also needed to include a Shopping Cart Item class.

The third step was to model each of these partitions as packages. Figure 7-A shows the classes being contained in their respective packages. Observe that the Credit-Card center currently is not contained in any package.

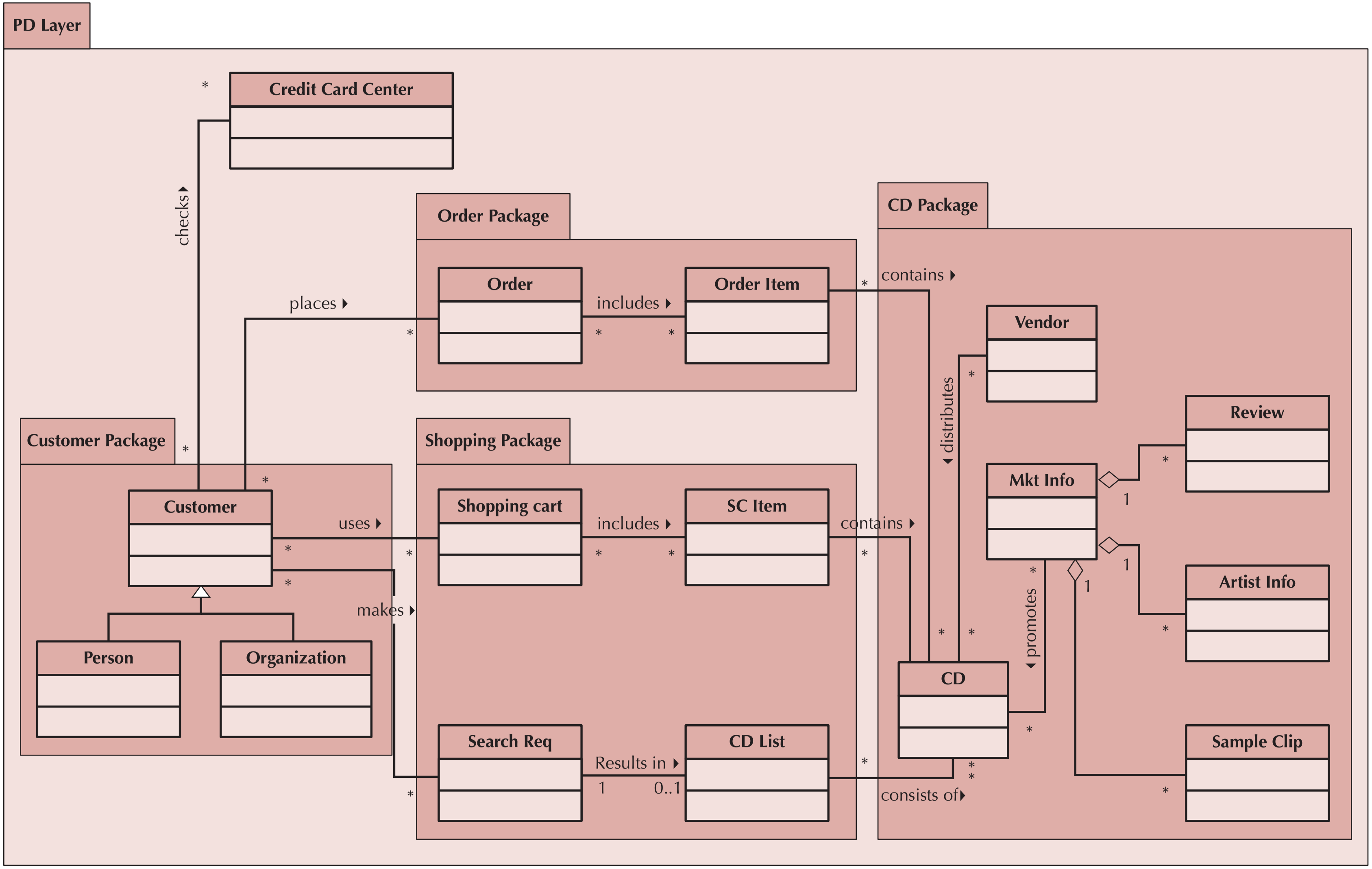


FIGURE 7-A Package Diagram of the PD Layer of CD Selections Internet Sales System

Next, Alec quickly identified four associations among the different packages: the Customer Package and the Order Package, the Customer Package and the Shopping Package, the Order Package and the CD Package, and the Shopping Package and the CD Package. He also identified an association between the Credit Card Center class and the Customer Package. Based on these associations, five dependency relationships were identified.

The fifth and final step was to place the dependency relationships on the package diagram. Again, to increase the understandability of the dependency relationships among the different packages, Alec decided to create a pure package diagram that only depicted the highest-level packages (and in this case the Credit Card Center class) and the dependency relationships (see Figure 7-B).

Verifying and Validating the Analysis Models

Upon completing the partitioning of the Problem Domain Layer, the team felt pretty good about what they had accomplished. However, based on his understanding of what was coming up next, Alec wanted to be sure that the analysis models–functional, structural, behavioral–and the partitions made sense, he decided that all of the work to date needed to go through a verification and validation step. To say the least, the team was not all that excited about this. In fact, Brian pointed out that the team had been verifying and validating everything as they went along. As such, he argued that this would

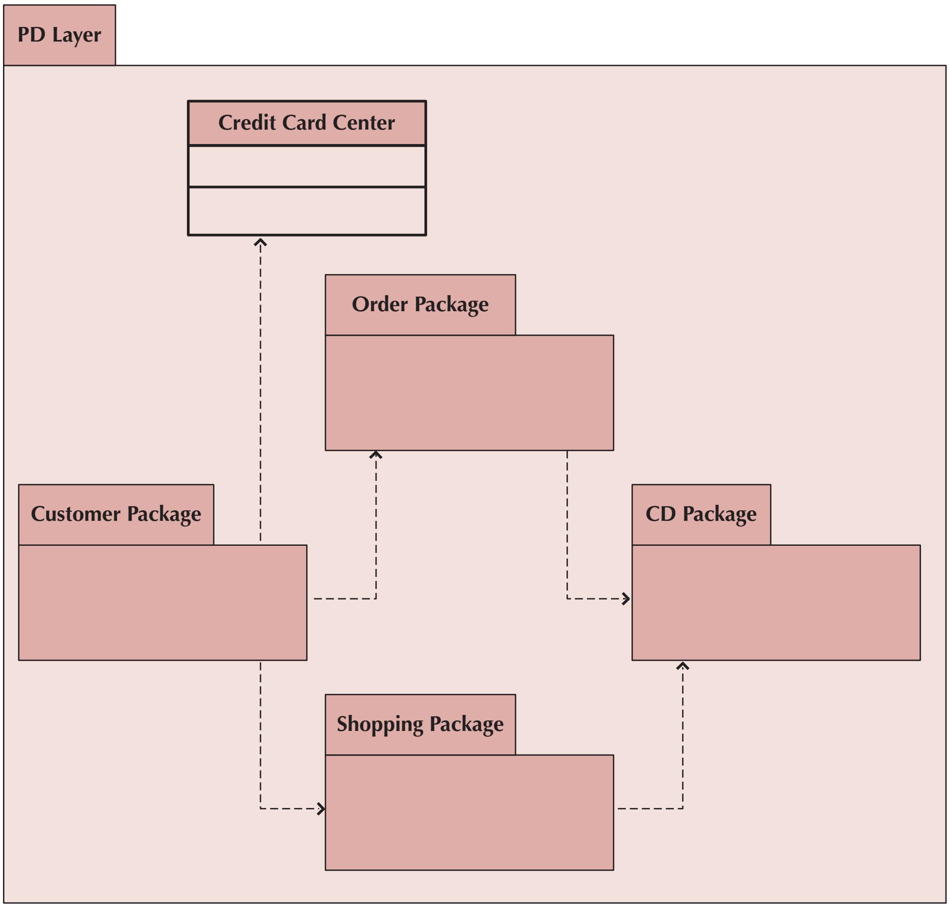


FIGURE 7-B
Overview Package Diagram of the PD Layer of CD Selections Internet Sales System

simply be a waste of time. But, Alec prevailed. He explained that in past projects, when they had not assured the quality of the Problem Domain Layer that teams had run into significant problems. These problems included the system not solving the right problem, significant cost overruns, and the system not being delivered on time. Since, this team was relatively inexperienced with the technology that they were about to use, Alec told the team, that there was not enough slack in the workplan to run into problems related to the analysis models. He suggested that the team perform a walkthrough with the analysis models and to ensure that all the relationships among the diagrams were fully tested (see Figures 7-1, 7-2, 7-8, 7-13, 7-16)).

The good news was that given all of the verification and validation that they had performed on the individual models, the team did not uncover any additional errors within the models. Brian sort of brought up his earlier point in a "I told you so" manner. However, Alec let him blow off a little steam and simply reminded the team that it was better to have done the verification and validation step now and not have to be sorry later. He pointed out that the other layers are mostly dependent on the Problem Domain Layer (see Figure 7-19) and any mistake not caught now could be very costly to catch it later.

Developing the Actual Design

Now that the team had verified and validated the analysis models, Alec had to decide on a design strategy. As he saw it, he had three different approaches that he could take with the new system: he could develop the entire system using development resources from CD Selections, he could buy a commercial Internet sales packaged software program (or a set of different packages and integrate them), or he could hire a consulting firm or service provider to create the system. Immediately, Alec ruled out the third option. Building Internet applications, especially sales systems, was important to the CD Selections' business strategy. By outsourcing the Internet Sales System, CD Selections would not develop Internet application development skills and business skills within the organization.

Instead, Alec decided that a custom development project using the company's standard Web development tools would be the best choice for CD Selections. In this way, the company would be developing critical technical and business skills in-house, and the project team would be able to have a high level of flexibility and control over the final product. Also, Alec wanted the new Internet Sales System to directly interface with the existing distribution system, and there was a chance that a packaged solution would not be able to integrate as well into the CD Selections environment.

There was one part of the project that potentially could be handled using packaged software: the shopping cart portion of the application. Alec realized that a multitude of programs have been written and are available (at low prices) to handle a customer's order transaction over the Web. These programs allow customers to select items for an order form, input credit card and billing information, and finalize the order transaction. Alec believed that the project team should at least consider some of these packaged alternatives so that less time had to be spent writing a program that handled basic Web tasks, and more time could be devoted to innovative marketing ideas and custom interfaces with the distribution system.

To help better understand some of the shopping cart programs that were available in the market and how their adoption could benefit the project, Alec created an alternative matrix that compared three different shopping cart programs to one another (see Figure 7-C). Although all three alternatives had positive points, Alec saw Alternative B (WebShop) as the best alternative for handling the shopping cart functionality for the new Internet sales

	Alternative 1: Shop-With-Me	Alternative 2: WebShop	Alternative 3: Shop-N-Go
Technical Feasibility	• Developed using C: very little C experience in-house • Orders sent to company using email files	• Developed using C and JAVA: would like to develop in-house JAVA skills • Flexible export features for passing order information to other systems	• Developed using JAVA: would like to develop in-house JAVA skills • Orders saved to a number of file formats
Economic Feasibility	• $150 initial charge	• $700 up front charge, no yearly fees	• $200/year
Organizational Feasibility	• Program used by other retail music companies	• Program used by other retail music companies	• Brand new application: few companies have experience with Shop-N-Go to date
Other Benefits	• Very simple to use	• Tom in IS support has had limited, but positive experience with this program • Easy to customize	
Other Limitations			• The interface is not easily customized

FIGURE 7-C Alternative Matrix for Shopping Cart Program

system. WebShop was written in JAVA, the tool that CD Selections selected as its standard Web development language; the expense was reasonable, with no hidden or recurring costs; and there was a person in-house who had some positive experience with the program. Alec made a note to look into acquiring WebShop as the shopping cart program for the Internet sales system.

CHAPTER 8: CLASS AND METHOD DESIGN

Up until now, Alec, Margaret, and the development team members have been focusing on being sure that they captured the underlying behavior and structure of the evolving system. During this installment, Alec instructs the team members to make sure that the connascence is minimized at all levels of the design, to identify any opportunities for reuse, to consider restructuring and optimizing the evolving specification. Furthermore, he instructed them to identify any and all constraints that need to be modeled. He also suggested that they define the invariants in a separate text file and to define the preconditions and postconditions for all public methods using contracts. Finally, he instructed the team to specify every method using the method specification form.

Alec and his team began the detailed object design process by reviewing the class and package diagram for the problem domain layer (see Figures 7-A and 7-B). Alec made it clear that the team should be aware of the cohesion, coupling, and connascence design criteria and to review the models with those in mind. Furthermore, he insisted that they look to see if there were any additional specifications necessary, any opportunities for reuse that could be exploited, and any further restructuring of the design. Alec assigned Anne to review all results and to look for any possible optimizations that could be implemented. Finally, since the implementation would be in Java, he asked Anne to also ensure that the design could be implemented in a single-inheritance-based language.

Upon their review, it was discovered that there were quite a few many-to-many (*..*) association relationships on the class diagram. Alec questioned whether this was a correct representation of the actual situation. Brian admitted that when they put together the class diagram, they had decided to model most of the associations as a many-to-many multiplicity, figuring that this could be easily fixed at a later point in time when they had more precise information. Alec also questioned why this issue was not addressed during the verification and validation step. However, he did not assign any blame at this point in time. Instead, since Brian was the team member that was most familiar with structural modeling and was the analyst in charge of the data management layer (see Chapter 9), Alec assigned him to evaluate the multiplicity of each association in the model and to restructure and optimize the evolving problem domain model.

Figure 8-A shows the updated version of the class diagram. As you can see, Brian included both the lower and upper values of the multiplicity of the associations. He did this to remove any ambiguity about the associations. Since there is a one-to-one relationship between the CD class and the Mkt Info class, Brian considered merging them into a single class. However, he decided that not all CDs would necessarily have any marketing information associated with them for the CD to be included in an order. Consequently, he reasoned that the Mkt Info associated with every CD would be optional and he should keep the Mkt Info separate. He also realized that, even though the team had attempted to verify and validate the structural model, they had not gotten the multiplicities correct in many places. For example, he recognized that he should have known that an artist could be associated with multiple CDs. Consequently, without changing the multiplicities, the Artist Info would have been duplicated for each CD with which the artist was associated.

Upon reviewing the new revised class diagram and since Brian had already spent quite a bit of time on the classes in the CD package, Alec assigned it to him. The classes in the CD package were CD, Vendor, Mkt Info, Review, Artist Info, and Sample Clip. Since Anne was going to have to review all classes and packages from a more technical perspective, Alec

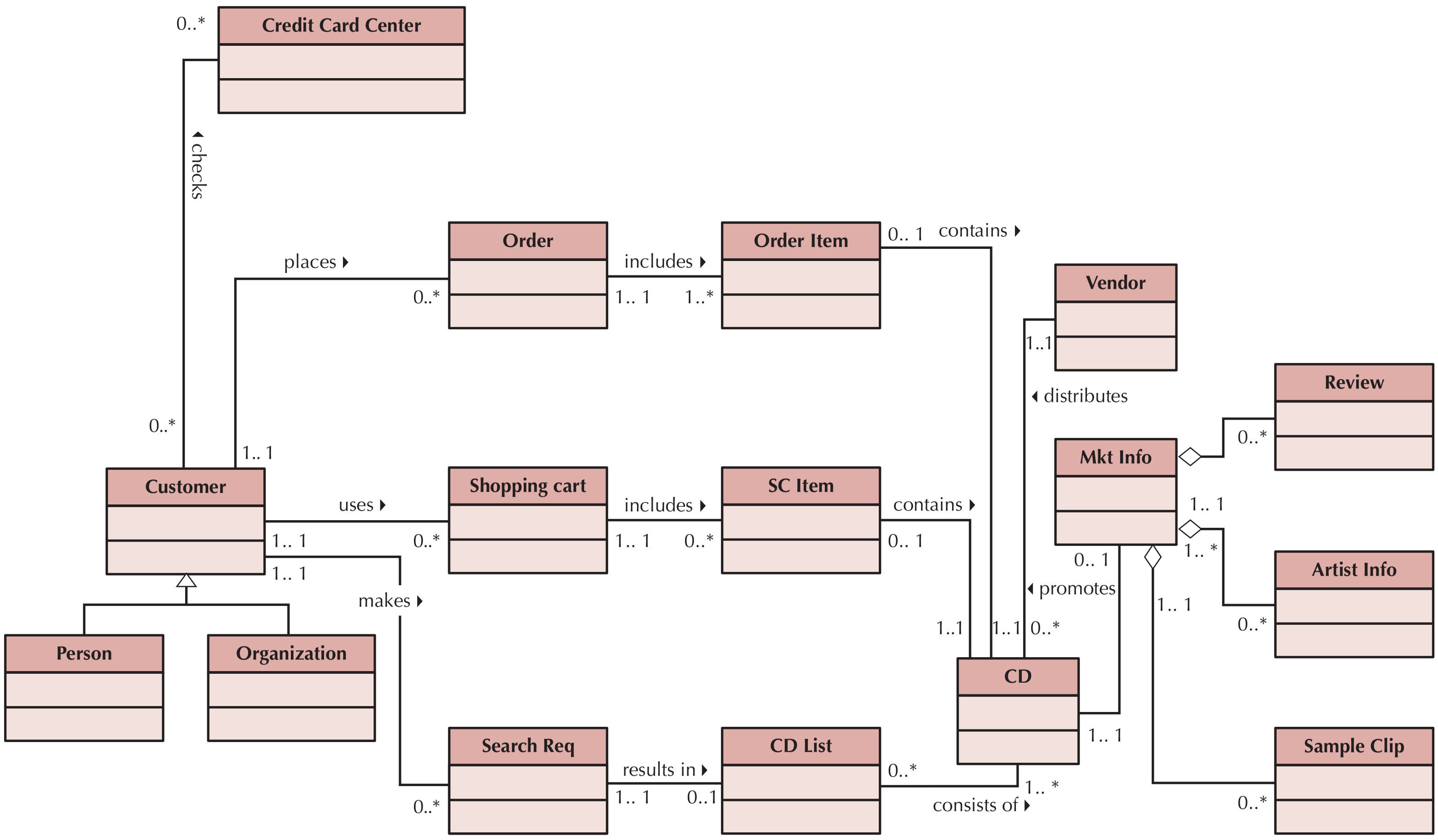


FIGURE 8-A Revised CD Selections Internet Sales System Class Diagram (Places Order Use Case View)

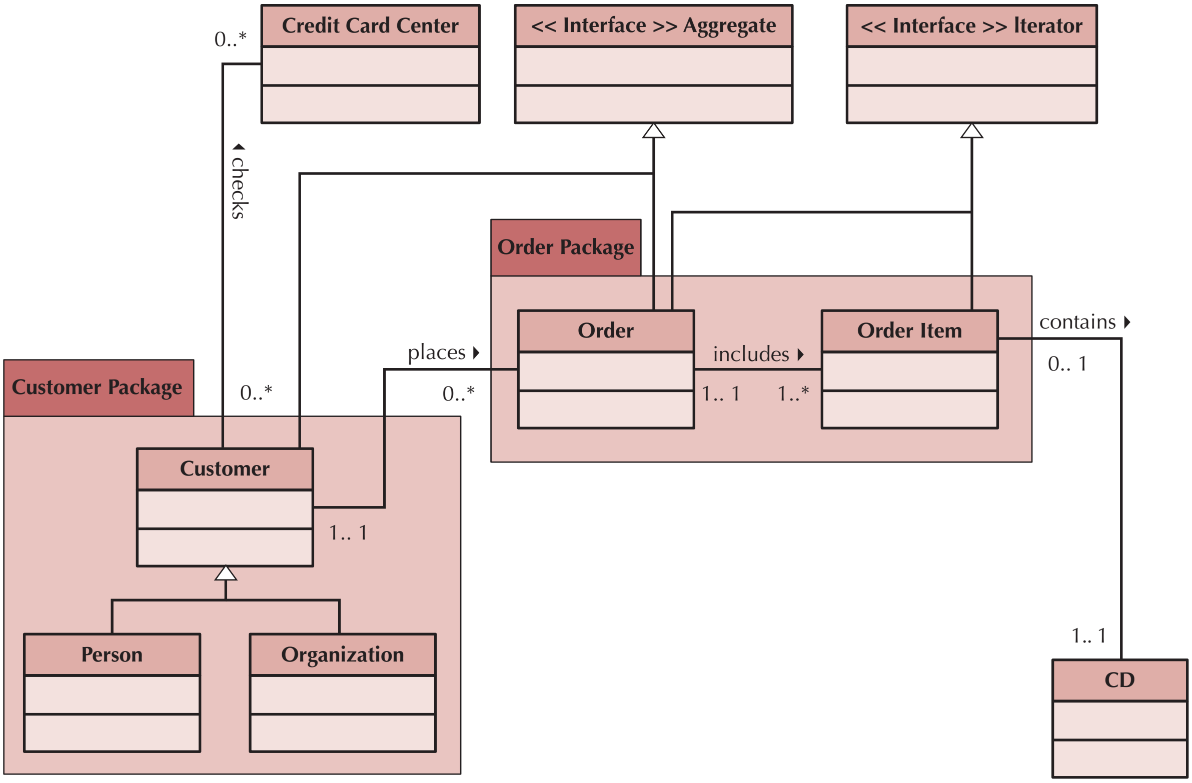
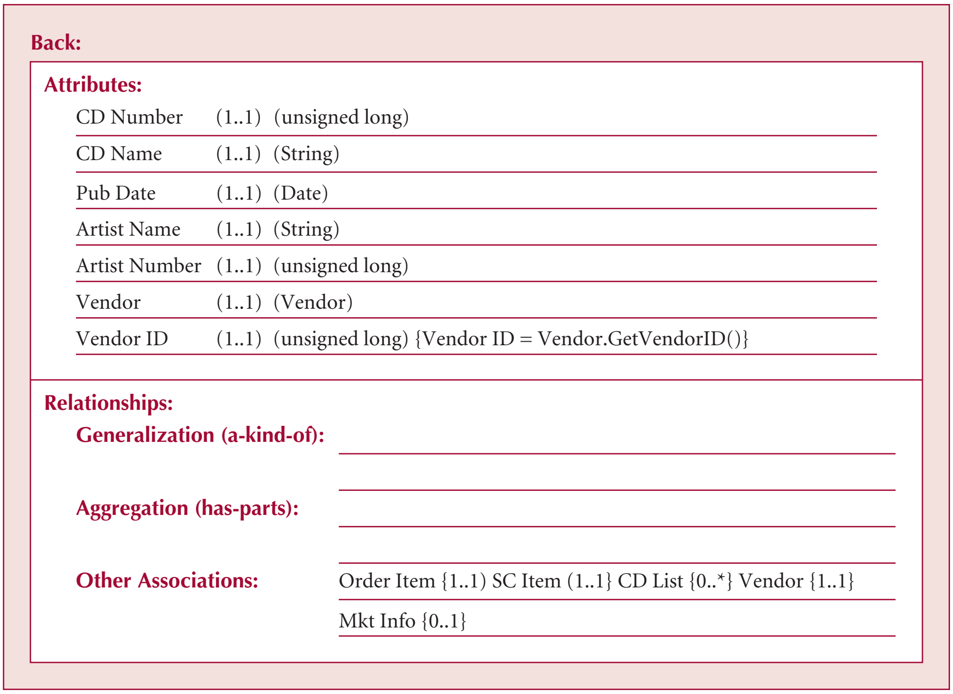


FIGURE 8-B Class Diagram for the Customer and Order classes in the Customer and Order Packages

decided to take on the Customer and Order packages himself (see Figures 7-A and 7-B). However, since the Shopping package could become rather technical, he assigned it to Anne.

Based on earlier projects, Alec suspected that there could be a set of patterns that should be useful in developing the Order Package. With a little work, Alec uncovered the Order, Product Order, and Product classes described in the textbook (see Figures 8-15, 8-19, and 8-20). By using these reusable classes, Alec was able to better define the Order, Order Item, and CD classes, respectively. He also discovered the Iterator and Whole-Part patterns (see Figures 8-13A, 8-13B, and 8-14). He decided that he should reuse the Iterator pattern in both the Customer and Order packages. He also recommended to Brian and to Anne to use the Product class as a basis for the CD class. Finally, Alec recommended to Brian to consider the Whole-Part pattern in the CD package for the Mkt Info class and to Anne, the use of the Iterator pattern for the Shopping Cart and CD List classes in the Shopping package. Alec hopes that by using these patterns the overall quality of the Internet Sales System will be improved in comparison to starting from scratch. Figure 8-B portrays the Order and Customer packages after Alec used the patterns.

Next, Brian added invariants, pre-conditions, and post-conditions to the classes and their methods. For example, Figure 8-C portrays the back of the CRC card for the CD class. He decided to add only the invariant information to the CRC cards and not the class diagram to keep the class diagram as simple and as easy to understand as possible. Notice the additional set of multiplicity, domain, and referential integrity invariants added to the attributes and relationships. Furthermore, Brian created contracts for each method. For example,



Back:

Attributes:

CD Number	(1..1)	(unsigned long)
CD Name	(1..1)	(String)
Pub Date	(1..1)	(Date)
Artist Name	(1..1)	(String)
Artist Number	(1..1)	(unsigned long)
Vendor	(1..1)	(Vendor)
Vendor ID	(1..1)	(unsigned long) {Vendor ID = Vendor.GetVendorID()}

Relationships:

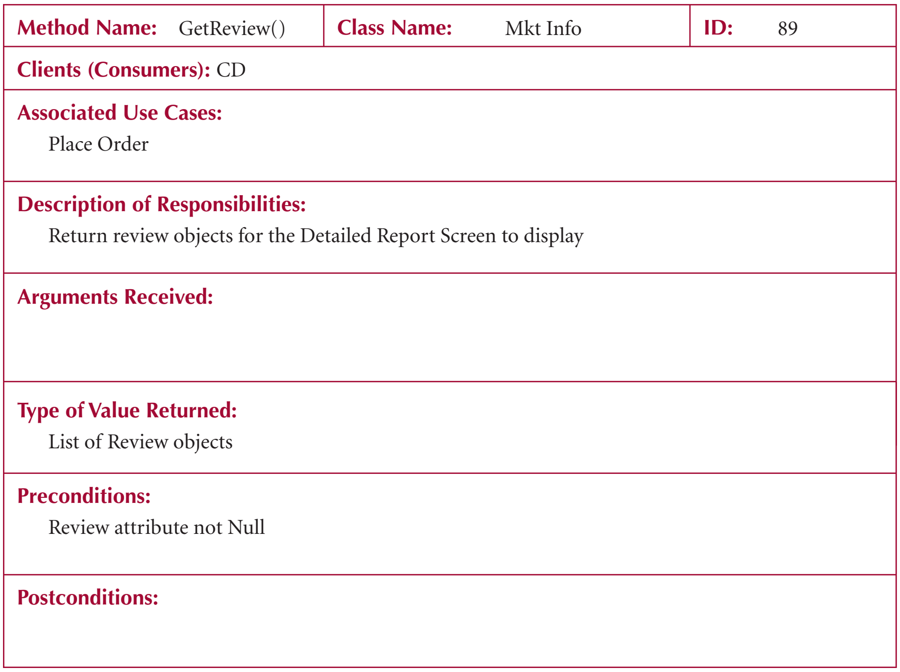
Generalization (a-kind-of):

Aggregation (has-parts):

Other Associations: Order Item {1..1) SC Item (1..1} CD List {0..*} Vendor {1..1}
Mkt Info {0..1}

FIGURE 8-C Back of CD CRC Card

Figure 8-D portrays the contract for the GetReview() method associated with the Mkt Info class. Notice that there is a pre-condition for this method to succeed—Review attribute not Null. Given the overall simplicity of the contracts with the classes in the CD package, Brian decided not to use OCL like constraints (see Figure 8-18). He hopes when the team brings everything back together, that the use of English-like constraints will be sufficient.



Method Name: GetReview() **Class Name:** Mkt Info **ID:** 89

Clients (Consumers): CD

Associated Use Cases:
Place Order

Description of Responsibilities:
Return review objects for the Detailed Report Screen to display

Arguments Received:

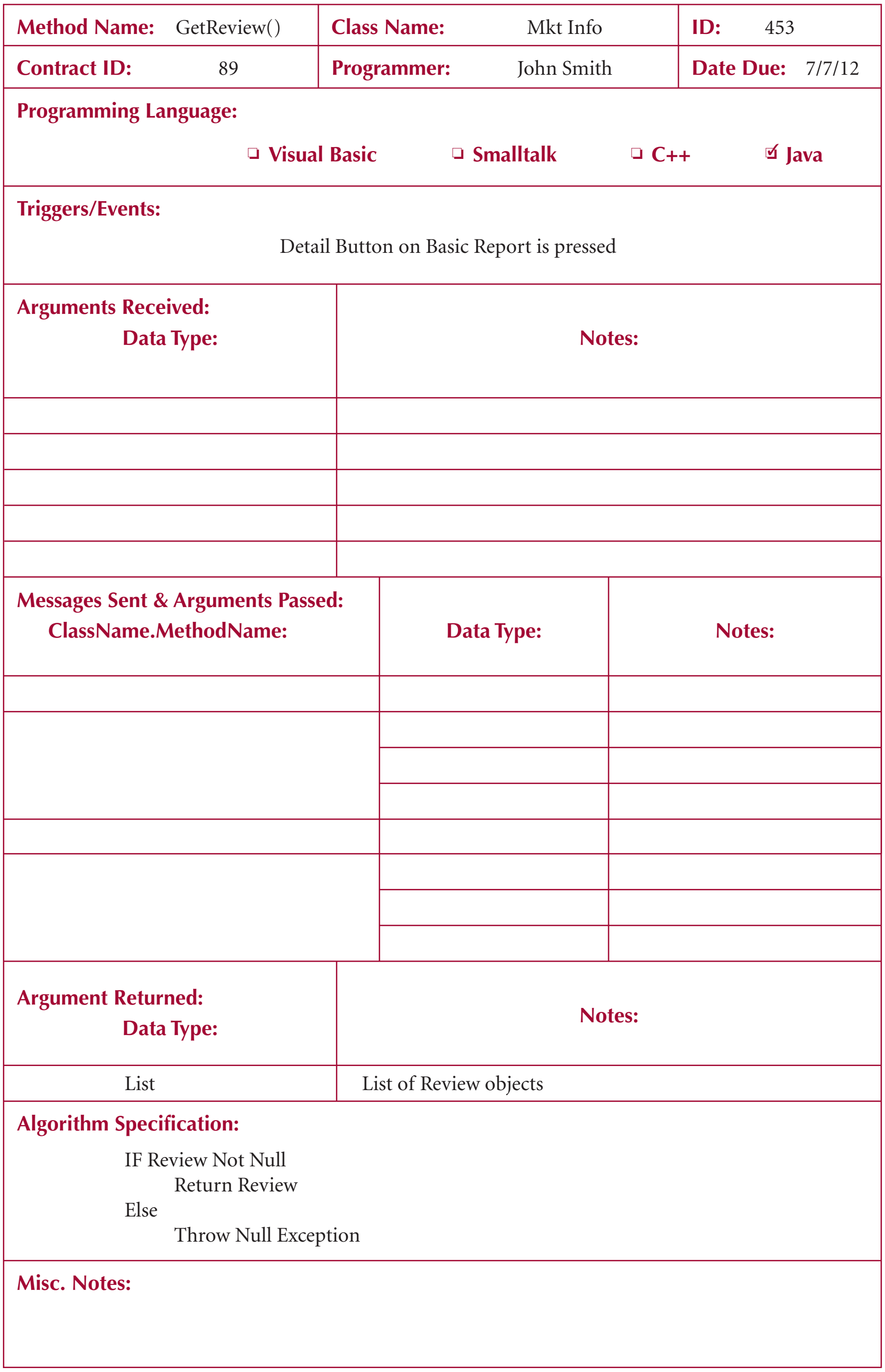
Type of Value Returned:
List of Review objects

Preconditions:
Review attribute not Null

Postconditions:

FIGURE 8-D Get Review Method Contract

Upon completing the CRC cards and contracts, Brian moved on to specifying the detailed design for each method. For example, the method specification for the GetReview() method is given in Figure 8-E. Brian developed this specification by reviewing the Place Order use case (see Figure 4-G), the sequence diagram (see Figure 6-B), and the contract (see Figure 8-D). Notice that Brian is enforcing the pre-condition on the contract by testing



Method Name: GetReview()	**Class Name:** Mkt Info	**ID:** 453
Contract ID: 89	**Programmer:** John Smith	**Date Due:** 7/7/12

Programming Language:

□ **Visual Basic** □ **Smalltalk** □ **C++** ☑ **Java**

Triggers/Events:

Detail Button on Basic Report is pressed

Arguments Received:

Data Type:	**Notes:**

Messages Sent & Arguments Passed:

ClassName.MethodName:	**Data Type:**	**Notes:**

Argument Returned:

Data Type:	**Notes:**
List	List of Review objects

Algorithm Specification:

```
IF Review Not Null
        Return Review
Else
        Throw Null Exception
```

Misc. Notes:

FIGURE 8-E Create Review Method Specification

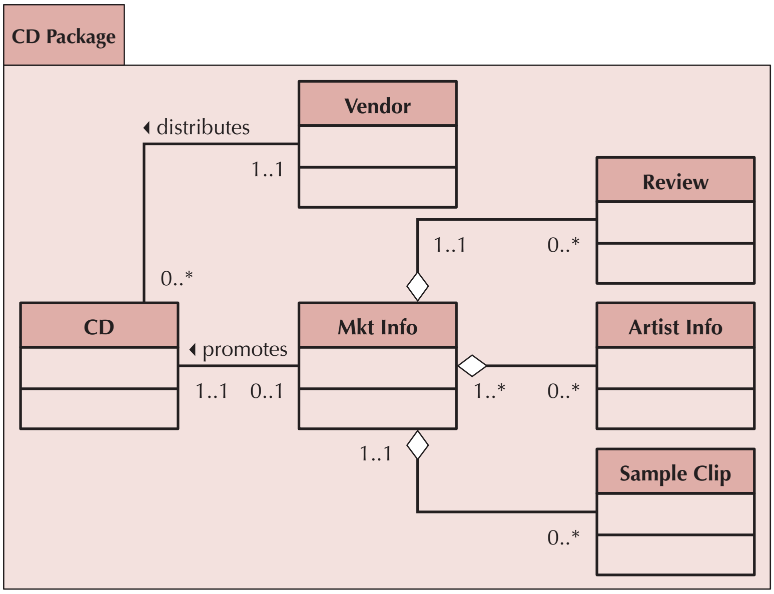


FIGURE 8-F Revised Package Diagram for the CD Package on the PD Layer of CD Selections Internet Sales System

to see whether the Review attribute that contains the list of reviews contains a value or not. Since the method is to be implemented in Java, he has specified that an exception is to be thrown if there are no reviews. Finally, Brian updated the class diagram for the CD package (see Figure 8-F). After looking at the diagram, Brian realized that there could be additional changes necessary depending on how the team decided to implement the data management layer (see Chapter 9). However, he decided that without any additional information regarding the data management layer, the current representation would have to suffice.

CHAPTER 9: DATA MANAGEMENT LAYER DESIGN

In the previous installments of the CD Selections case, we saw how Alec, Margaret, and the development team had worked through developing models and designs of the problem domain classes. Now that the design of the problem domain layer is somewhat stable, the team has moved into developing the models and designs of the solution domain (data management, human–computer interaction, and physical architecture) classes. In this installment, we follow the team members that have been assigned to the development of the data management layer classes for the Web-based system being developed for CD Selections.

To begin with, Margaret met with Alec to make sure that he realize that the CD Selections Internet Sales System needed to both present CD information effectively to users and to capture order data. Alec recognized that these goals were dependent upon a good design of the data management layer for the new application. He approached the design of the data management layer first by asking Brian to take charge of this task. Second, Alec made sure that Brian followed the following four steps:

- select the object-persistence format,
- map the problem domain classes to the selected format,
- optimize the selected format for processing efficiency, and
- design the data access and manipulation classes.

Brian assured Alec that he would do so and keep him abreast of the data management layer design as it progressed. Based on a quick review of the requirements, Brian requested to add two database experts to help with the design of the data management layer: John and Susan. After a little deliberation, Alec decided it would be worth the additional expense to add them to the team.

Select Object-Persistence Format

The first thing that Brian did was to call a meeting of his data management layer team to discuss two issues that would drive the object-persistence format selection: what kind of objects would be in the system and how they would be used. Using a whiteboard, they listed their ideas (see Figure 9-A). The project team agreed that the bulk of the data in the system

Data	Type	Use	Suggested Format
Customer information	Simple (mostly text)	Transactions	Relational
Order Information	Simple (text and numbers)	Transactions	Relational
Marketing Information	Both simple and complex (eventually the system will contain audio clips, video, etc.)	Transactions	Object add-on?
Information that will be exchanged with the Distribution System	Simple text, formatted specifically for importing into the Distribution System	Transactions	Transaction file
Temporary Information	The Web component will likely need to hold information for temporary periods of time. (e.g., the shopping card will store order information before the order is actually placed)	Transactions	Transaction file

FIGURE 9-A
Types of Data in Internet Sales System

would be the text and numbers that are exchanged with Web users regarding customers and orders. A relational database would be able to handle the data effectively, and the technology would be well received at CD Selections because relational technology is already in place throughout the organization.

However, they realized that relational technology was not optimized to handle complex data, such as the images, sound, and video that the marketing facet of the system ultimately will require. Alec asked Brian to look into relational databases that offered object add-on products (i.e., an RDBMS that could become an ORDBMS). It might be possible for the team to invest in a RDBMS foundation and then upgrade to an ORDBMS version of the same product. However, in the meantime, Alec decided to store sample clips using a random file. This way, they could still deliver the system as envisioned while keeping the technology requirements reasonable.

The team also noted that it must design two transaction files to handle the interface with the distribution system and the Web shopping cart program. The Internet Sales System will regularly download order information to the distribution system using a transaction file containing all the required information for that system. Also, the team must design the file that stores temporary order information on the Web server as customers shop through the Web site. The file would contain the fields that ultimately would be transferred to an order object. Of course, Alec realized that other data needs might arise over time, but he felt confident that the major data issues were identified (e.g., like the capability to handle complex data) and that the design of the data management layer would be based on the proper storage technologies.

Map Problem Domain Objects to Object-persistence format

Based on the decision to use an RDBMS and a random file to store the problem domain objects, Brian created an object-persistence design. To begin, Brian first reviewed the current class and package diagrams for the evolving Internet Sales System (see Figures 7-A, 7-B, 8-A, 8-B, and 8-F). Focusing on these figures, Brian and his team began applying the appropriate mapping rules (see Figure 9-9). Based on Rule 1, Brian identified 12 problem domain classes that needed to have there objects stored; as such, Brian created 11 RDBMS tables and 1 file to represent these objects. These included Credit Card Center table, Customer table, Person table, Organization table, Order table, Order Item table, CD table, Vendor table, Mkt Info table, Review table, Artist Info table, and Sample Clip File. He also created a set of tentative primary keys for each of the tables and the file. Based on the fact that the objects in the Shopping Package (see Figures 7-A, 7-B, and 8-A), Shopping Cart, SC Item, Search Req, and CD List, are only temporary, Brian decided that there was no real need to address them at this point in the design.

Using Rule 4, Brian identified the need for the CD table and the Mkt Info table to both have each others primary key stored as a foreign key to the other table. Upon further reflection, Brian reasoned that because an actual instance of marketing information was only going to be associated with a single instance of CD and vice versa, he could have merged the two tables together. However, the team had earlier decided to keep them separate, so he decided to simply use the primary key of the CD table as the primary key of the Mkt Info table.

While reviewing the current set of attributes for each of the tables, John suggested that the development had left out the idea of a CD containing a set of tracks. As such, they added a multi-valued attribute, tracks, to the CD problem domain class. However, Brian then pointed out that when they apply Rule 5 to the CD class, they really needed to factor the tracks attribute out as a separate table. Furthermore, as Brian, John, and

Susan discussed the track attribute further, it was decided to include it as a problem domain class also.

Next, the data management layer team applied Rule 6 to the evolving object-persistence design. In doing this, Susan pointed out that the checks relationship between the Customer and Credit Card Center problem domain classes was a multivalued association. Furthermore, Brian then pointed out the same was true for the aggregation relationship between the Artist Info and Mkt Info problem domain classes. Therefore, these relationships needed their own table in the relational database. Not to be upstaged by Brian and Susan, John immediately pointed out that Rule 7 was applicable to 8 associations: Customer places Order, Order includes Order Item, Order Item contains CD, Vendor distributes CD, Mkt Info promotes CD, CD contains Tracks, and the three aggregation associations with the Mkt Info class (Review, Artist Info, and Sample Clip). As such, quite a few primary keys had to be copied into the relevant tables as foreign keys (e.g., the primary key of the Customer table had to be copied to the Order table). Can you identify the others?

Finally, Susan suggested the solution to the inheritance problem because RDBMSs do not support inheritance. She pointed out that when applying Rule 8a to the Customer superclass and the Person and Organization subclasses, the primary key of the Customer table also had to be copied to the tables representing the subclasses. Furthermore, she pointed out that an exclusive-or (XOR) condition existed between the two subclasses.

Based on all of the suggestions and hard work accomplished by the data management layer team, Brian was able to create a design of the object persistence for the Internet Sales System (see Figure 9-B).

Optimize Object Persistence and Estimate its Size

Upon completing the object persistence design, Brian requested a meeting with the development team to walkthrough the design.[6] After the walkthrough, Alec asked Brian to stay behind to discuss the data management layer model. Now that the team had a good idea of the type of object-persistence formats that would be used, they were ready for the third step: optimizing the design for performance efficiency. Since Brian was the analyst in charge of the data management layer, Alec wanted to discuss with him whether the model was optimized for storage efficiency. He also needed this done before the team discussed access speed issues. Brian assured Alec that the current object persistence model was in third normal form. He was confident of this because the project team followed the modeling guidelines that lead to a well-formed model.

Brian then asked about the file formats for the two transaction files identified in the earlier meeting. Alec suggested that he normalize the files to better understand the various tables that would be involved in the import procedure. Figure 9-C shows the initial file layout for the Distribution System import file as well as the steps that were taken as Brian applied each normalization rule.

The next step for the design of the data management layer was to optimize the design for access speed. Alec met with the data management layer design team and talked about the techniques that were available to speed up access. Together they listed all of the data

[6] It seems that Brian finally got the importance of verifying and validating everything.

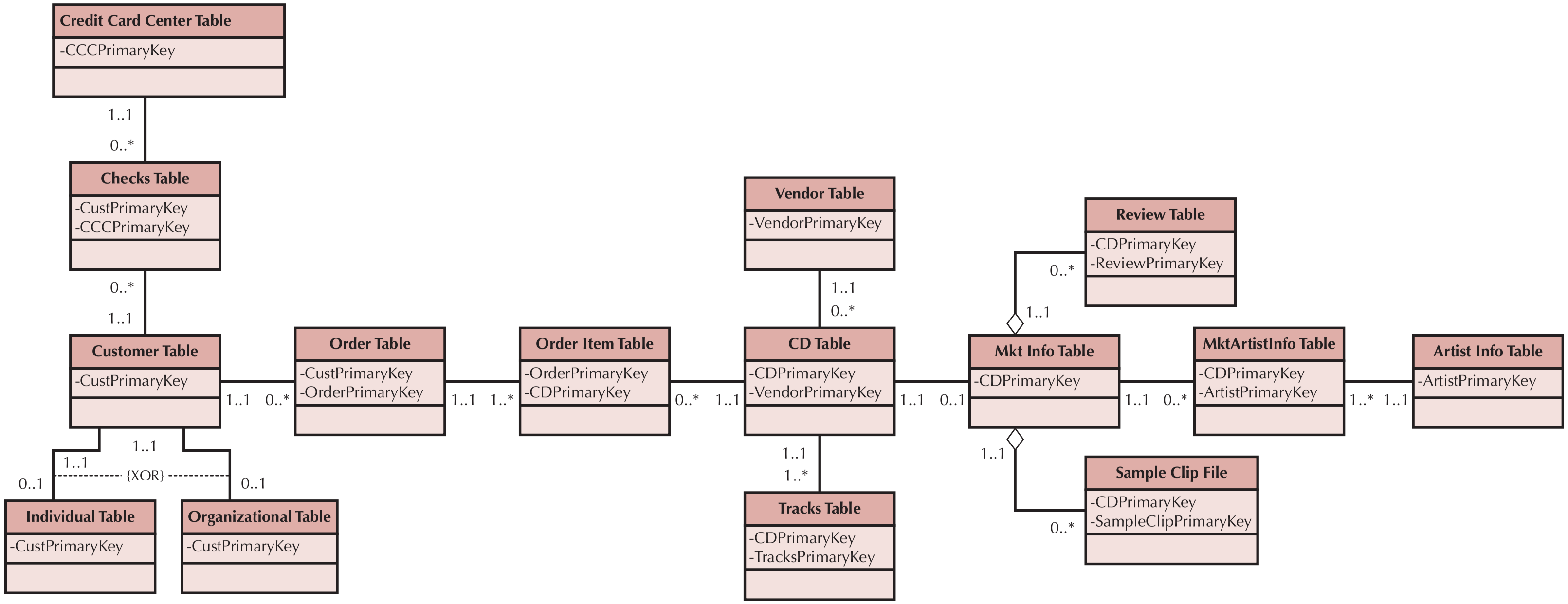


FIGURE 9-B Internet Sales System Object Persistence Design

0NF:

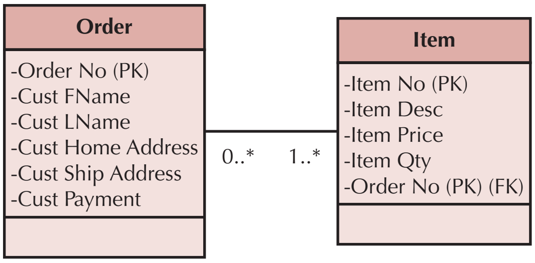
File Layout Required for Distribution System

Order Number	Cust Fname	Cust Lname	Cust HomeAdd	Cust ShipAdd	Cust Pay	Item No*	Item Desc*	Item Price*	Item Qty*
A (7)	A (20)	A (20)	A (150)	A (150)	9999.99	A (7)	A (20)	9999.99	9999

* Item No, Item Desc, Item Price, and Item Qty repeats four (4) times.

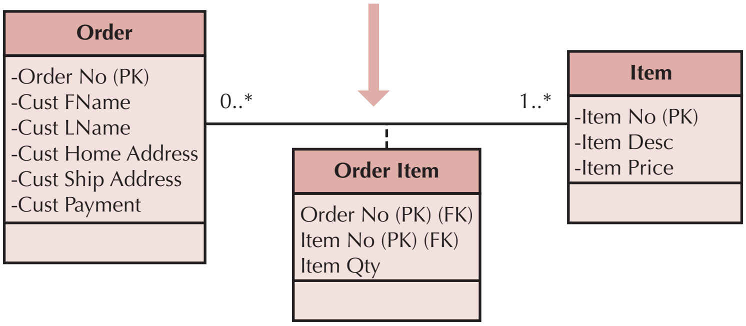
Remove repeating groups of items and place them in a separate Item entity

1NF:



There is one partial dependencies in the above data model since Item qty is dependent on the whole primary key while Item Desc and Item Price is dependent only on Item No.

2NF:



Remove transitive dependencies. Customer home-address is dependent upon Cust-fname and Cust-lname (and these do not serve as the identifier for the Order entity); therefore, a Customer entity is added to contain customer information. Order then contains only order information and necessary foreign keys.

3NF:

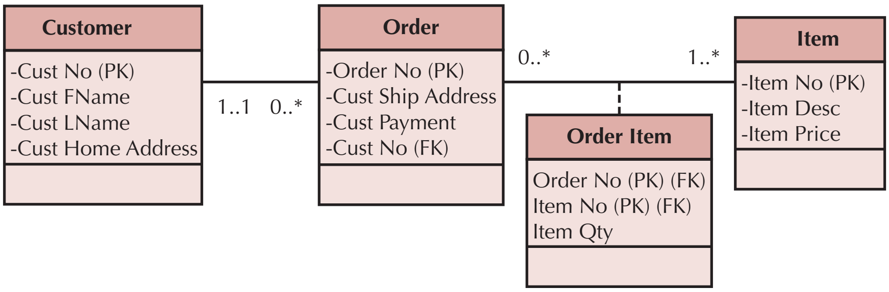


FIGURE 9-C
Distribution System
Import File
Normalization Process

Target	Comments	Suggestions to Improve Data Access Speed
All tables	Basic table manipulation	• Investigate if records should be clustered physically by primary key • Create indexes for primary keys • Create indexes for foreign key fields
All tables	Sorts and Grouping	• Create indexes for fields that are frequently sorted or grouped
CD information	Users will need to search CD information by title, artist, and category	• Create indexes for CD title, artist, and category
Order Information	Operators should be able to locate information about a particular customer's order	• Create an index in the Order table for orders by customer name
Entire Physical Model	Investigate denormalization opportunities for all fields that are not updated very often	• Investigate one-to-one relationships • Investigate look-up tables • Investigate one-to-many relationships

FIGURE 9-D
Internet Sales System Performance

that will be supported by the Internet Sales System and discussed how all of the data would be used. Based on these discussions, they developed the strategy to identify the specific techniques to put in place (see Figure 9-D).

Ultimately, clustering strategies, indexes, and denormalization decisions were applied to the physical data model, and a volumetrics report was run from the CASE tool to estimate the initial and projected size of the database. The report suggested that an initial storage capacity of about 450 megabytes would be needed for the expected one-year life of the first version of the system. Additional storage capacity would be needed for the second version that would include sound files for samples of the songs, but for the moment not much storage would be needed.

Since Anne was in charge of managing the server hardware acquisition, Alec gave the estimates to her so that the she could make sure that the technology could handle the expected volume of data for the Internet Sales System. She then would send the estimates to the DBMS vendor during the implementation of the software so that the DBMS could be configured properly.

Data Access and Manipulation Class Design

The final step in designing the data management layer was to develop the design of the data access and manipulation classes that would act as translators between the object persistence and the problem domain classes. Since the CD package (see Figures 7-A, 7-B, 8-A, and 8-F) was the most important package, Alec asked Brian to complete the design of the data management layer for the CD package and report back to Alec when he was finished. Upon reviewing the concrete problem domain classes in the CD package, Brian realized that he needed to have seven data access and manipulation classes; one for each concrete problem domain class. These classes would be fully dependent on their related problem domain classes. Next, Brian mapped the data access and manipulation classes down to the RDBMS tables and the random file associated with storing the objects (see Figure 9-B). In this case, there were seven RDBMS tables and one random file. Again, the data access and manipulation classes are dependent on the object-persistence format. Brian created Figure 9-E to depict the data management layer and problem domain layer for the CD package of the Internet sales system.

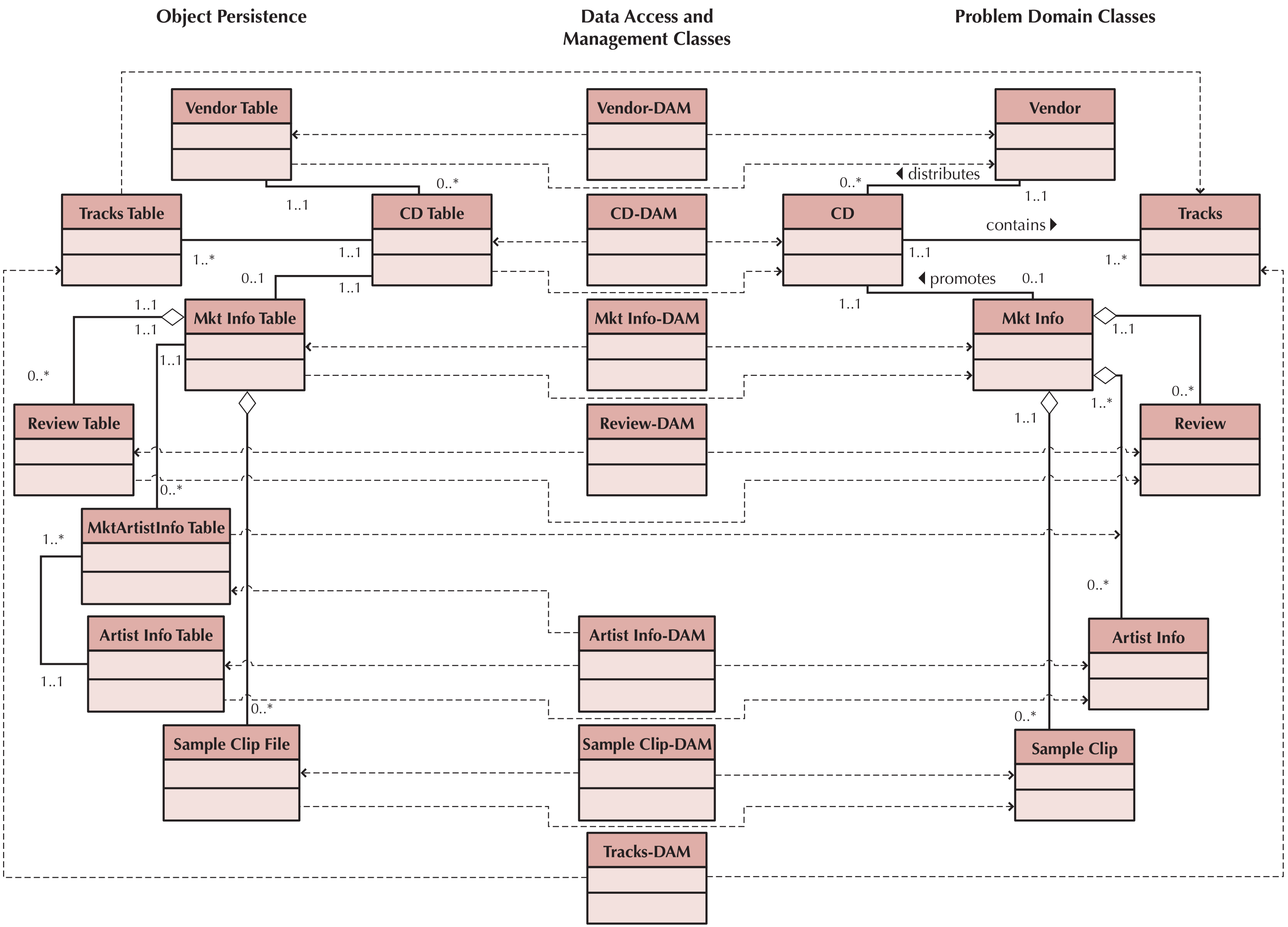


FIGURE 9-E Data Management Layer and Problem Domain Layer Design for the CD Package of the Internet Sales System

When Brian was going over the data management layer design with Alec (see Figure 9-E), Alec noticed that the Artist Info-DAM class was dependent upon both the Artist Info Table and the MktArtistInfo Table. Brian explained that this was necessary since an instance of the Artist Info problem domain class can only exist if a related instance of the Mkt Info problem domain class exists (see the multiplicities of 1..*). To guarantee this, the Artist Info-DAM class must depend on the existence of both the Artist Info Table and the table that represents the required relationship: MktArtistInfo Table. However, since the relationship from the Mkt Info problem domain class to the Artist Info class was optional (see multiplicities of 0..*), the Mkt Info-DAM class was not dependent on the MktArtistInfo Table. If it had been a required relationship also, then the Mkt Info-Dam class would have also been dependent on the relationship based table. Based on Brian's design and explanation, Alec felt that the design of the data management layer was completed.

CHAPTER 10: HUMAN-COMPUTER INTERACTION LAYER DESIGN

Previously, Alec had the development team focusing on developing the analysis models of the problem domain. In the previous chapter's installment, Alec had split part of the team and had assigned them to work on the data management layer and to develop its design. In this installment, we follow the development team members that have been assigned to the human–computer interaction layer. Based on what Margaret has learned about mobile computing, social media, and globalization, she really wants to be able to deploy across multiple platforms in such a way that CD Selections will be able to reach a global market. However, Alec isn't quite sure that trying to deploy over multiple incompatible platforms is a good idea.

To begin with, the team reviewed the functional models for the CD Selections Internet Sales System. The use case diagram showed that there were three different high-level use cases in (see Figures 4-A and 4-D): Maintain CD Information, Place Order, and Maintain CD Marketing Information. There are also six additional use cases, Search/Browse CDs, Checkout, Create New Customer, Place In Store Hold, Place Special Order, and Fill Mail Order, associated with the Place Order use case.

Based on the perceived complexity of developing the current system without the deployment on multiple incompatible platforms, Alec was able to convince Margaret that even though he agreed with her as to the importance of both mobile and social computing platforms, that it would be better if they delayed the design and deployment for those platforms until a later version of the system is developed. Furthermore, given that the current system's focus was more about bringing in more customers to the bricks and mortar stores, they also agreed to delay the internationalization of the web site until a later version.

To keep the complexity of the current example under control, in this section, we focus only on the Place Order, Browse/Search CDs, and Checkout use cases.

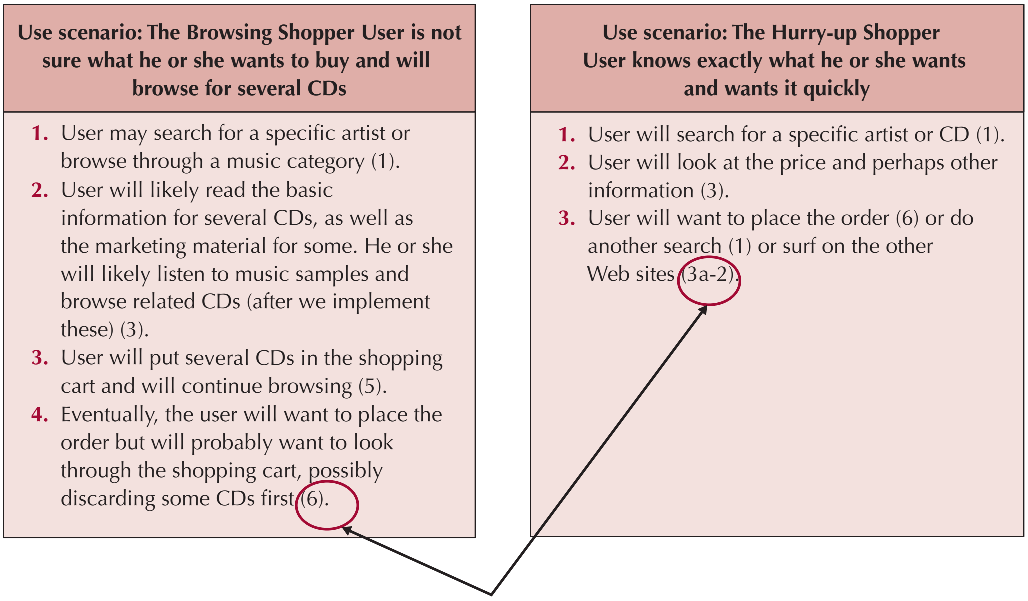
Use Scenario Development

The first step in the interface design process was to develop the key use scenarios for the Internet Sales System. Since Alec assigned the human computer interaction layer design to himself, he began by examining the essential use cases (see Figures 4-B through 4-G) and thinking about the types of users and how they would interact with the system. To begin with, Alec identified two use scenarios: the browsing shopper and the hurry-up shopper (see Figure 10-A).[7] Alec also thought of several other use scenarios for the Web site in general, but omitted them since they were not relevant to the Internet sales portion. Likewise, he thought of several use scenarios that did not lead to sales (e.g., fans looking for information about their favorite artists and albums), and omitted them as well.

Interface Structure Design

Next, Alec created a window navigation diagram (WND) for the Web system. He began with the Place Order, Browse/Search CDs, and Checkout essential use cases to ensure that

[7] Of course, it may be necessary to modify the original essential use cases in light of these new subtypes of customer. Furthermore, the structural and behavioral models may have to be modified. Remember that object-oriented systems analysis and design is incremental and iterative, as such, additional requirements can be uncovered at any time.



Use scenario: The Browsing Shopper User is not sure what he or she wants to buy and will browse for several CDs

1. User may search for a specific artist or browse through a music category (1).
2. User will likely read the basic information for several CDs, as well as the marketing material for some. He or she will likely listen to music samples and browse related CDs (after we implement these) (3).
3. User will put several CDs in the shopping cart and will continue browsing (5).
4. Eventually, the user will want to place the order but will probably want to look through the shopping cart, possibly discarding some CDs first (6).

Use scenario: The Hurry-up Shopper User knows exactly what he or she wants and wants it quickly

1. User will search for a specific artist or CD (1).
2. User will look at the price and perhaps other information (3).
3. User will want to place the order (6) or do another search (1) or surf on the other Web sites (3a-2).

The numbers in parentheses refer to specific events in the essential use case.

FIGURE 10-A Use Scenarios for the Browsing and Hurry-Up Customers

all functionality defined for the system was included in the WND. Figure 10-B shows the WND for the Web portion of the Internet Sales System. The system will start with a home page that contains the main menu for the sales system. Based on the essential use cases, Alec identified four basic operations that he felt made sense to support on the main menu: search the CD catalog, search by music category, review the contents of the shopping cart, and to actually place the order. Each of these was modeled as a hyperlink on the home page.

Alec then decided to model the full search option as a pop-up search menu that allowed the customer to choose to search the CD catalog based on artist, title, or composer. He further decided that a textbox would be required to allow the customer to type in the name of the artist, title, or composer depending on the type of search requested. Finally, he chose to use a button to submit the request to the system. After the "submit" button is pressed, the system produces a report that was composed of hyperlinks to the individual information on each CD. A CD report containing the basic information is generated by clicking on the hyperlink associated with the CD. On the basic report, Alec added buttons for choosing to find out additional information on the CD and to add the CD to the shopping cart. If the "Detail" Button is pressed, a detailed report containing the marketing information on the CD is produced. Finally, Alec decided to include a button on this report to add the CD to the shopping cart.

The second basic operation supported on the home page was to allow the user to search the CD catalog by category of music. Like the previous operation, Alec chose to model the category search with a pop up search menu. In this case, once the customer chose the category, the system would produce the report with the hyperlinks to the individual information on each CD. From that point on, the navigation would be identical to the previous searches.

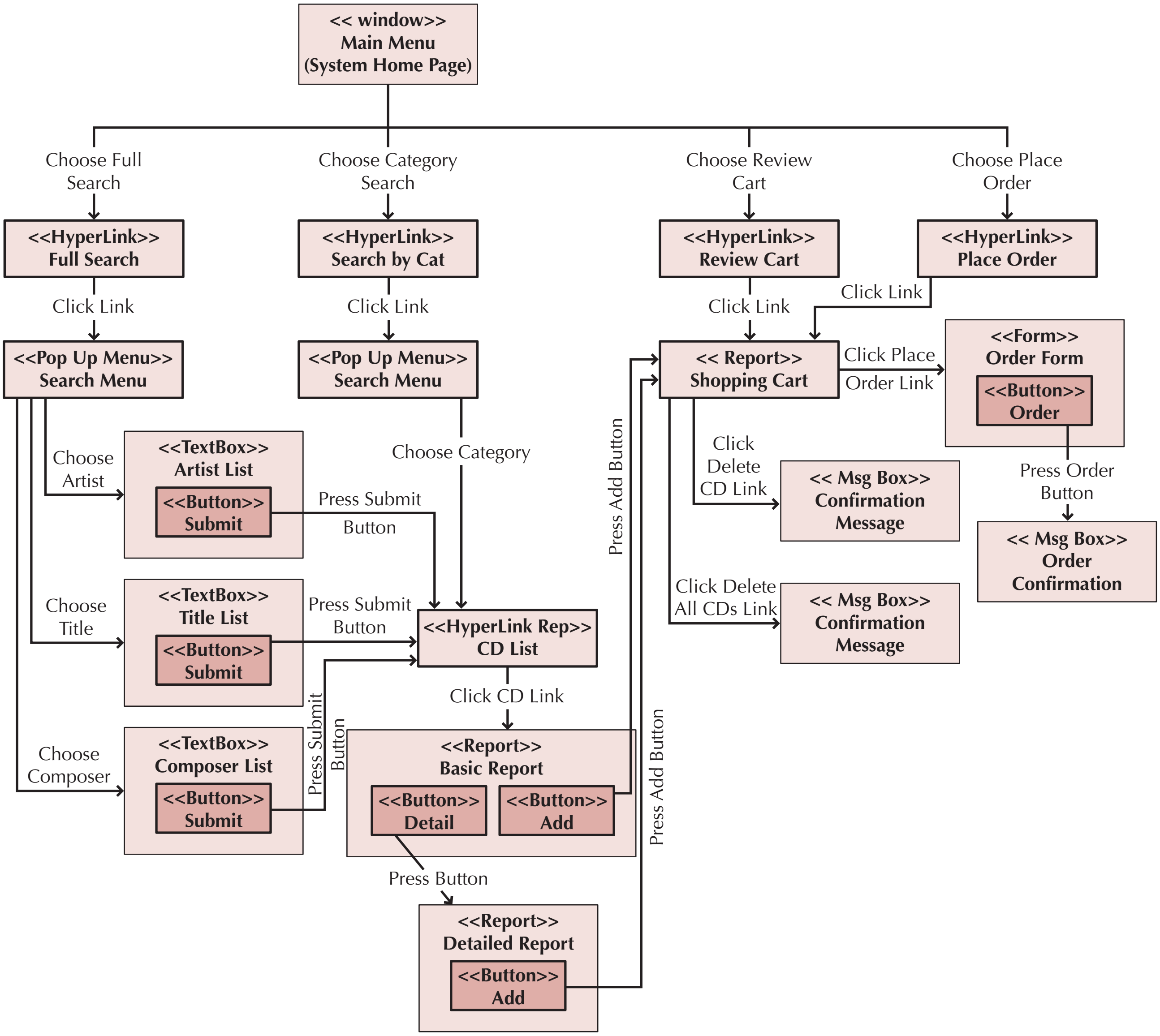


FIGURE 10-B CD-Selections Initial WND for the Web portion of the Internet Sales System

The third operation supported was to review the contents of the shopping cart. In this case, Alec decided to model the shopping cart as a report that contained three types of hyperlinks; one for removing an individual CD from the shopping cart, one for removing all CDs from the shopping cart, and one for placing the order. The removal hyperlinks would remove the individual CD (or all of the CDs) from the shopping cart if the user would confirm the operation. The place order link would send the customer to an order form. Once the customer filled out the order form, the customer would press the order button. The system would then respond with an order confirmation message box.

The fourth operation supported on the home page was to allow the customer to place an order directly. Upon review, Alec decided that the place order and review cart operations were essentially identical. As such, he decided to force the user to have the Place Order and Review Shopping Cart operations go through the same process.

Alec also envisioned that by using frames, the user would be able to return to the home page from any screen. Documenting these would give the WND too many lines, so Alec simply put a note describing it with the WND.

The Revised WND

Alec then examined the use scenarios to see how well the initial WND enabled different types of users to work through the system. He started with the Browsing Shopping use scenario and followed it through the WND, imagining what would appear on each screen and pretending to navigate through the system. He found the WND to work well, but he noticed a couple of minor issues related to the shopping cart. First, he decided that it would make sense to allow the customer to retrieve the information related to the CDs contained in the shopping cart. As such, he changed the stereotype of the user interface component from Report to HyperLink Rep and added a hyperlink from the Shopping Cart to the Basic Report created by the different search requests. Second, he noticed that the Shopping Cart was using hyperlinks to link to the Removal and Place Order processes. However, in all the other elements of the WND, he was using buttons to model the equivalent ideas. As such, he decided to change the Shopping Cart component to model these connections as buttons. Of course, this forced him to modify the transitions as well.

Alec next explored the Hurry-up Shopper use scenario. In this case, the WND did not work as well. Moving from the home page, to the search page, to the list of matching CDs, to the CD page with price and other information takes three mouse clicks. This falls within the three clicks rule, but for someone in a hurry, this may be too many. Alec decided to add a "quick-search" option to the home page that would enable the user to enter one search criteria (e.g., just artist name or title, rather than a more detailed search as would be possible on the search page) that would with one click take the user to the one CD that matched the criteria or to a list of CDs if there were more than one. This would enable an impatient user to get to the CD of interest in one or two clicks.

Once the CD is displayed on the screen, the Hurry-up Shopper use scenario would suggest that the user would immediately purchase the CD, do a new search, or abandon the Web site and surf elsewhere. This suggested two important changes. First, there had to be an easy way to go to the place order screen. As the WND stands (see Figure 10-B), the user must add the item to the shopping cart and then click on the link on the HTML frame to get to the place order screen. While the ability of users to notice the place order link in the frame would await the interface evaluation stage, Alec suspected, based on past experience, that a significant number of users would not see it. Therefore, he decided to add a buttons to the Basic Report screen and the Detailed Report screen called "Buy" (See Figure 10-C).

Second, since the Hurry-up Shopper might want to search for another CD instead of buying the CD, Alec decided to include the quick-search item from the home page on the frame. This would make all searches immediately available from anywhere in the system. This would mean that all functionality on the home page would now be carried on the frame. Alec updated the note on the bottom attached to the WND to reflect the change.

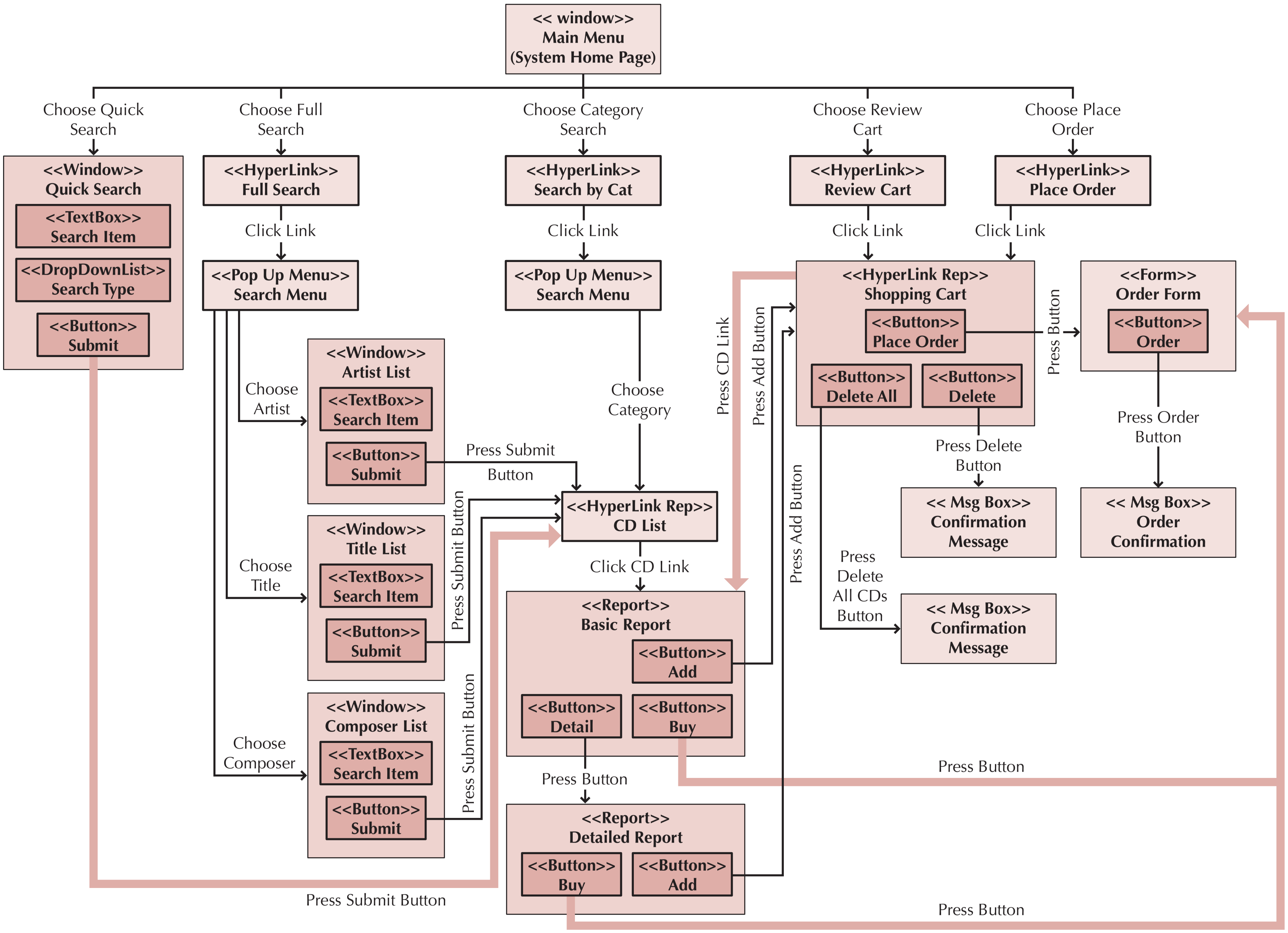


FIGURE 10-C CD-Selections revised WND for the Web portion of the Internet Sales System (changes are highlighted)

Finally, upon review of the WND, Alec decided to remodel the Artist List, Title List, and Composer Lists as window stereotypes instead of textbox stereotypes. He then added a Search Item textbox to each of these elements. Figure 10-C shows the revised WND for the Web portion of the Place Order use case. All changes are highlighted.

Interface Standards Design

Once the WND was complete, Alec moved on to develop the interface standards for the system. The interface metaphor was straightforward: a CD Selections music store. The key interface objects and actions were equally straightforward, as was the use of the CD Selections logo icon (see Figure 10-D).

Interface Template Design

For the interface template, Alec decided on a simple, clean design that had a modern background pattern, with the CD-Selections logo in the upper-left corner. The template had two navigation areas: one menu across the top for navigation within the entire Web site (e.g., overall Web site home page, store locations) and one menu down the left edge for navigation within the Internet sales system. The left edge menu contained the links to the top-level operations (see WND in Figure 10-C), as well as the "quick search" option. The center area of the screen is used for displaying forms and reports when the appropriate operation is chosen (see Figure 10-E).

At this point, Alec decided to seek some quick feedback on the interface structure and standards before investing time in prototyping the interface designs. Therefore, he met with Margaret Mooney, the project sponsor, and Chris Campbell, the consultant, to discuss the emerging design. Making changes at this point would be much simpler than after doing the prototype. Margaret and Chris had a few suggestions, so after the meeting Alec made the changes and moved into the design prototyping step.

Design Prototyping

Alec decided to develop a hypertext mark-up language (HTML) prototype of the system. The Internet sales system was new territory for CD Selections and a strategic investment in

Interface Metaphor: A CD Selections Music Store

Interface Objects

- **CD:** All items, whether CD, tape, or DVD, unless it is important to distinguish among them
- **Artist:** Person or group who records the CD
- **Title:** Title or name of CD
- **Composer:** Person or group who wrote the music for the CD (primarily used for classical music)
- **Music Category:** Type of music. Current categories include: Rock, Jazz, Classical, Country, Alternative, Soundtracks, Rap, Folk, Gospel.
- **CD List:** List of CD(s) that match the specified criteria
- **Shopping Cart:** Place to store CDs until they are requested

Interface Actions

- **Search for:** Display a CD list that matches specified criteria
- **Browse:** Display a CD list sorted in order by some criteria
- **Order:** Authorize special order or place hold

Interface Icons

- **CD Selections Logo:** Will be used on all screens

FIGURE 10-D
CD Selections Interface Standards

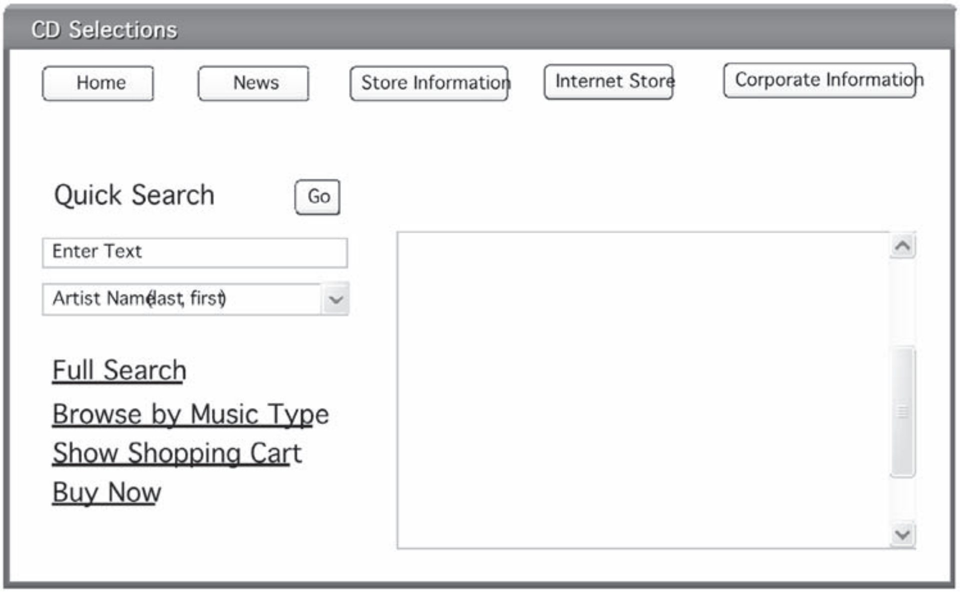


FIGURE 10-E CD Selections Interface Template for the Web Portion of the Internet Sales System

a new business model, so it was important to make sure that no key issues were overlooked. The HTML prototype would provide the most detailed information and enable interactive evaluation of the interface.

In designing the prototype, Alec started with the home screen and gradually worked his way through all the screens. The process was very iterative and he made many changes to the screens as he worked. Once he had an initial prototype designed, he posted it on CD Selections intranet and solicited comments from several friends with lots of Web experience. He revised it based on the comments he received. Figure 10-F presents some screens from the prototype.

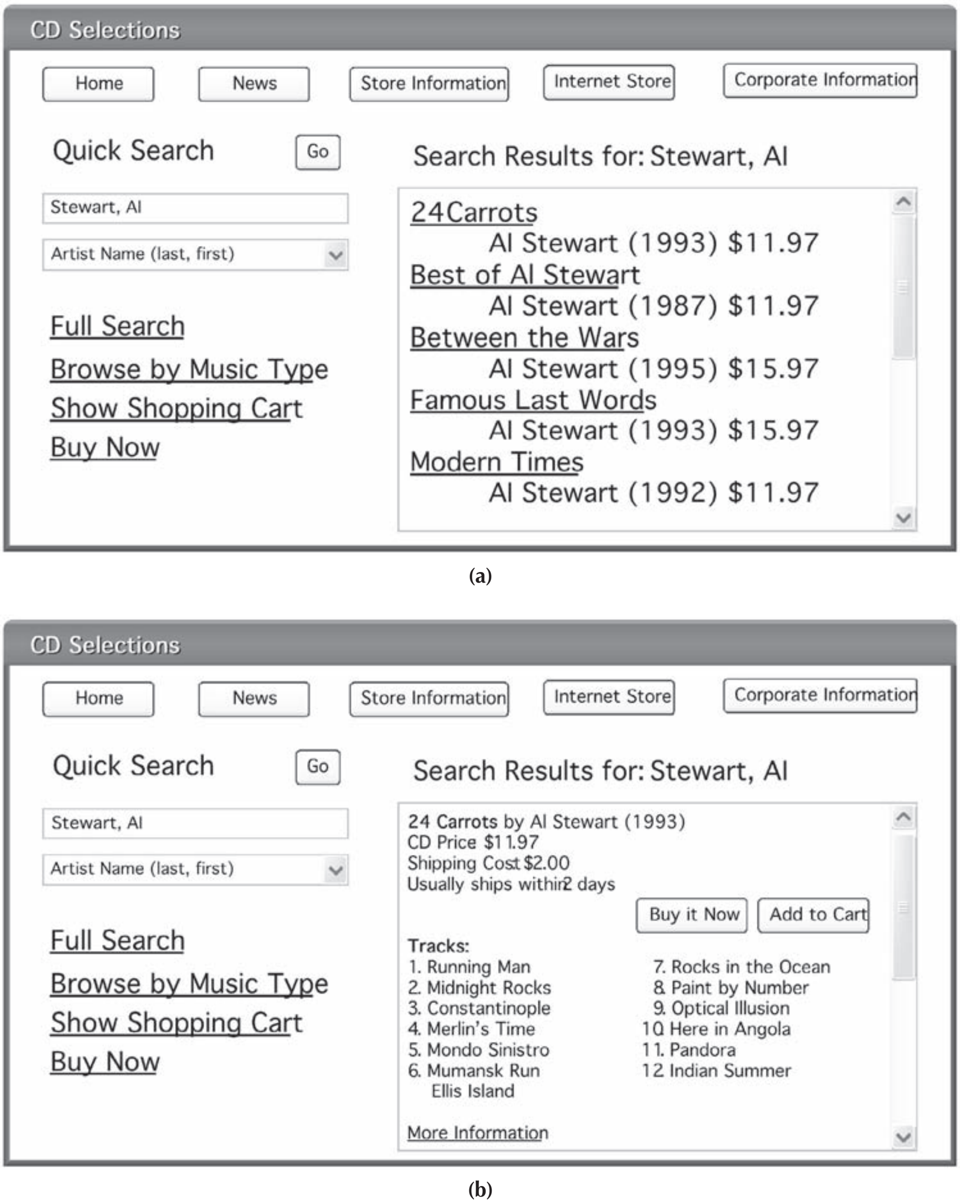
Interface Evaluation

The next step was interface evaluation. Alec decided on a two-phase evaluation. The first evaluation was to be an interactive evaluation conducted by Margaret, her marketing managers, selected staff members, selected store managers, and Chris. They worked hands-on with the prototype and identified several ways to improve it. Alec modified the HTML prototype to reflect the changes suggested by the group and asked Margaret and Chris to review it again.

The second evaluation was another interactive evaluation, this time by a series of two focus groups of potential customers—one with little Internet experience, the other with extensive Internet experience. Once again, several minor changes were identified. Alec again modified the HTML prototype and asked Margaret and Chris to review it again. Once they were satisfied, the interface design was complete.

Navigation Design Documentation

The last step that Alec completed was to document the navigation design through the use of real use cases. To accomplish this, Alec gathered together the essential use case (see Figure 4-G), the use scenarios (see Figure 10-A), the window navigation diagram



Stewart, Al

Artist Name (last, first)

Search Results for: Stewart, Al

24 Carrots
Al Stewart (1993) $11.97
Best of Al Stewart
Al Stewart (1987) $11.97
Between the Wars
Al Stewart (1995) $15.97
Famous Last Words
Al Stewart (1993) $15.97
Modern Times
Al Stewart (1992) $11.97

(a)

Stewart, Al

Artist Name (last, first)

Search Results for: Stewart, Al

24 Carrots by Al Stewart (1993)
CD Price $11.97
Shipping Cost $2.00
Usually ships within2 days

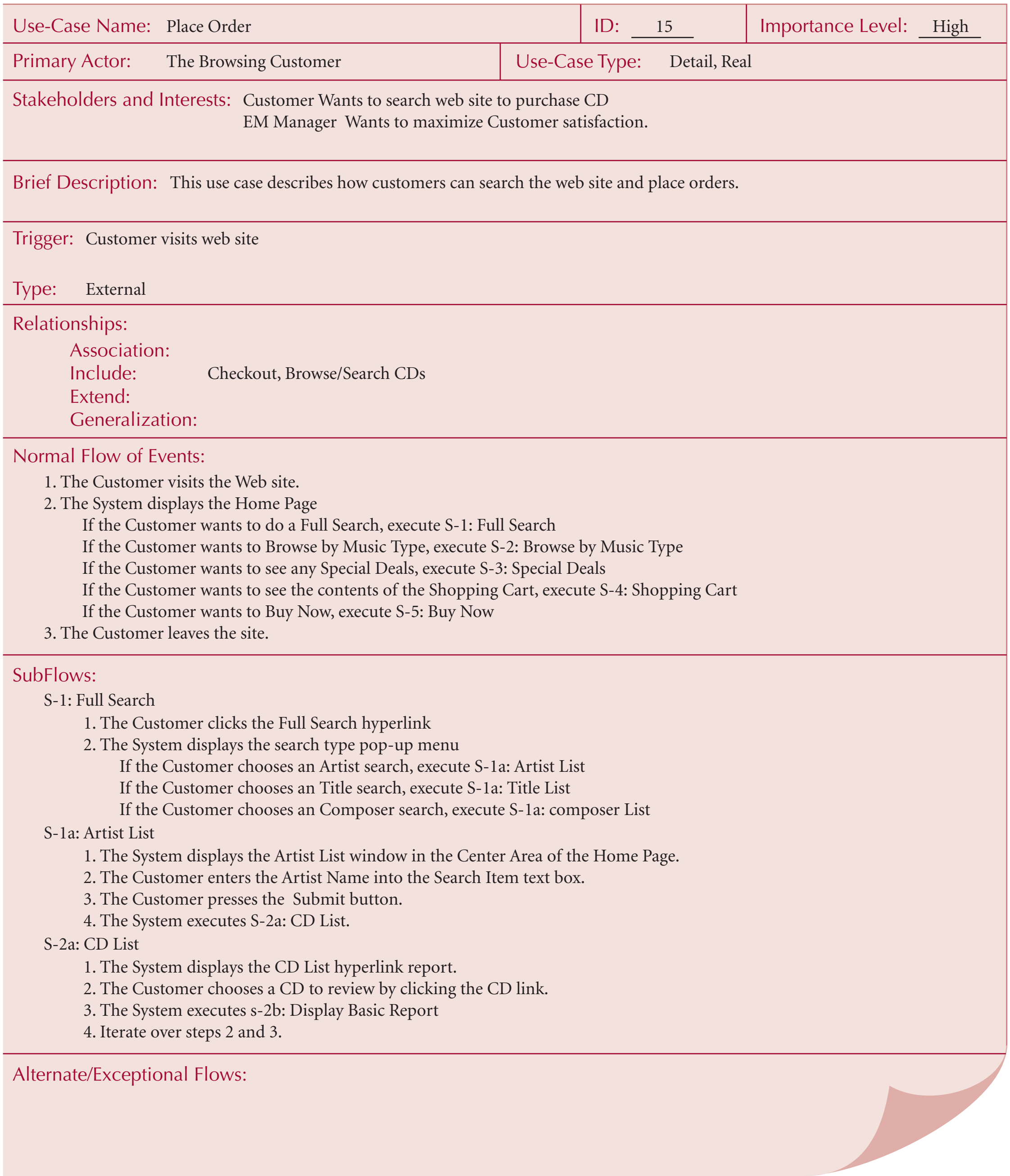
Tracks:

1. Running Man
2. Midnight Rocks
3. Constantinople
4. Merlin's Time
5. Mondo Sinistro
6. Mumansk Run
   Ellis Island
7. Rocks in the Ocean
8. Paint by Number
9. Optical Illusion
10. Here in Angola
11. Pandora
12. Indian Summer

(b)

FIGURE 10-F
Sample Interfaces from the CD Selections Design Prototype

(see Figure 10-C), and the user interface prototype (see Figures 10-E and 10-F). First, he copied the contents of the essential use case to the real use case. He changed the type from detail, essential to detail, real and the primary actor was specialized to browsing customer instead of simply customer. Second, he wrote the specific set of steps and responses that described the interaction between the browsing customer and system. Figure 10-G shows a partial listing of the steps in the Normal Flow of Events and SubFlows sections of the real use case. Last, he repeated the steps for the hurry-up customer. Based on the detailed, real use cases, Alec realized that changes really needed to be cascaded back to the use case diagram and the detailed, essential use case descriptions. Again, this is typical of the iterative and incremental nature of object-oriented system development.



Use-Case Name: Place Order | ID: 15 | Importance Level: High

Primary Actor: The Browsing Customer | Use-Case Type: Detail, Real

Stakeholders and Interests: Customer Wants to search web site to purchase CD
EM Manager Wants to maximize Customer satisfaction.

Brief Description: This use case describes how customers can search the web site and place orders.

Trigger: Customer visits web site

Type: External

Relationships:
Association:
Include: Checkout, Browse/Search CDs
Extend:
Generalization:

Normal Flow of Events:
1. The Customer visits the Web site.
2. The System displays the Home Page
   If the Customer wants to do a Full Search, execute S-1: Full Search
   If the Customer wants to Browse by Music Type, execute S-2: Browse by Music Type
   If the Customer wants to see any Special Deals, execute S-3: Special Deals
   If the Customer wants to see the contents of the Shopping Cart, execute S-4: Shopping Cart
   If the Customer wants to Buy Now, execute S-5: Buy Now
3. The Customer leaves the site.

SubFlows:
S-1: Full Search
   1. The Customer clicks the Full Search hyperlink
   2. The System displays the search type pop-up menu
      If the Customer chooses an Artist search, execute S-1a: Artist List
      If the Customer chooses an Title search, execute S-1a: Title List
      If the Customer chooses an Composer search, execute S-1a: composer List
S-1a: Artist List
   1. The System displays the Artist List window in the Center Area of the Home Page.
   2. The Customer enters the Artist Name into the Search Item text box.
   3. The Customer presses the Submit button.
   4. The System executes S-2a: CD List.
S-2a: CD List
   1. The System displays the CD List hyperlink report.
   2. The Customer chooses a CD to review by clicking the CD link.
   3. The System executes s-2b: Display Basic Report
   4. Iterate over steps 2 and 3.

Alternate/Exceptional Flows:

FIGURE 10-G The Browsing Customer Real Use Case (Partial Listing Only)

CHAPTER 11: PHYSICAL ARCHITECTURE LAYER DESIGN

As with the previous two chapters, this installment of the CD Selections case we see that Alec has spun off part of his team to focus on designing the physical architecture layer. However, given the dependence among the human–computer interaction, data management, and physical architecture layers, this group finds that they must be in relatively constant contact with the other groups. Otherwise, deploying the problem domain, human–computer interaction, and data management layers over the architecture could prove to be difficult. Consequently, Alec has decided to be focus his coordination efforts among the different layer groups by heading up the physical architecture layer's group. He saw this as a way to better understand the implications of deploying the system over multiple, and possibly incompatible, platforms (see the previous installment).

Alec realized that the hardware, software, and networks that would support the new application would need to be integrated into the current infrastructure at CD Selections. Consequently, he had asked Anne to review the high-level nonfunctional requirements developed in the analysis phase (see Figure 3-A) and by conducting a JAD session and a series of interviews with managers in the marketing department and three store managers to refine the nonfunctional requirements into more detail. Figure 11-A shows some of the results. The clear business need for a Web-based architecture required a thin client–server architecture for the Internet sales portion of the system.

CD Selections had a formal architecture group responsible for managing CD Selections architecture and its hardware and software infrastructure. Therefore, Anne set up a meeting with the development team and the architecture group. During the meeting, she confirmed that CD Selections was still moving toward a target client–server architecture, although the central mainframe still existed as the primary server for many server-based applications.

They discussed the Internet system and decided that it should be built using a three-tier thin client–server architecture. Everyone believed that it was hard to know at this point exactly how much traffic this Web site would get and how much power the system would require, but a client–server architecture would allow CD Selections to easily scale up the system as needed.

By the end of the meeting, it was agreed that a three-tiered client–server architecture was the best configuration for the Internet portion of the Internet sales system (i.e., the Place Order process in Figures 4-D and 4-G). Customers would use their personal computers running a Web browser as the client. A database server would store the Internet system's databases; whereas, an application server would have Web server software and the application software to run the system.

A separate two-tier client server system will maintain the CD and CD Marketing Material information (i.e., see Figure 4-D). This system will have an application for the personal computers of the staff working in the Internet sales group that communicates directly with the database server and enables staff to update the information. The database server will have a separate program to enable it to exchange data with CD Selections' distribution system on the company mainframe. Furthermore, the in-store system was currently built using a two-tier client–server architecture, so the portion of the system responsible for the in-store holds would conform to that architecture.

Next, Anne created a network model to show the major components of the Internet sales system (see Figure 11-B). The Internet sales system is on a separate network segment separated from the CD Selections' main network by a firewall that separates the network

1. Operational Requirements		
Technical Environment	1.1	The system will work over the Web environment with Internet Explorer and Real Audio.
	1.2	Customers will only need Internet Explorer and RA on their desktops.
System Integration	1.3	The Internet sales system will read information from the main CD information database, which contains basic information about the CD (e.g., title, artist, id number, price, quantity in inventory). The Internet order system will not write information to the main CD information database.
	1.4	The Internet sales system will transmit orders for new CDs in the special order system, and will rely on the special order system to complete the special orders generated.
	1.5	The Internet sales system will read and write to the main inventory database.
	1.6	A new module for the In-store system will be written to manage the "holds" generated by the Internet system. The requirements for this new module will be documented as part of the Internet sales system because they are necessary for the Internet sales system to function.
	1.7	A new module will be written to handle the mail order sales. The requirements for this new module will be documented as part of the Internet sales system because they are necessary for the Internet sales system to function.
Portability	1.8	The system will need to remain current with evolving Web standards, especially those pertaining to music formats.
Maintainability	1.9	No special maintainability requirements are anticipated.
2. Performance Requirements		
Speed	2.1	Response times must be less than 7 seconds.
	2.2	The inventory database must be updated in real time.
	2.3	In-store holds must be sent to the store within 5 minutes.
Capacity	2.4	There will be a maximum of 20–50 simultaneous users at peak use times.
	2.5	The system will support streaming audio to up to forty simultaneous users.
	2.6	The system will send up to 5K of data to each store daily.
	2.7	The in-store hold database will require 10–20K of disk space per store.
Availability and Reliability	2.8	The system should be available 24/7.
	2.9	The system shall have 99 percent uptime performance.
3. Security Requirements		
System Value	3.1	No special system value requirements are anticipated.
Access Control	3.2	Only store managers will be able to override In-Store Holds.
Encryption/Authentication	3.3	No special encryption/authentication requirements are anticipated.
Virus Control	3.4	No special virus control requirements are anticipated.
4. Cultural and Political Requirements		
Multilingual	4.1	No special multilingual requirements are anticipated.
Customization	4.2	No special customization requirements are anticipated.
Unstated Norms	4.3	No special unstated norms requirements are anticipated.
Legal	4.4	No special legal requirements are anticipated.

FIGURE 11-A Selected Nonfunctional Requirements for the CD Selections Internet Sales System

from the Internet while granting access to the Web and database servers. The Internet sales system has two parts. A firewall is used to connect the Web/Application server to the Internet, while another firewall further protects the Internet sales group's client computers and database server from the Internet. In order to improve response time, a direct connection is made from the Web/Application server to the database server because these will

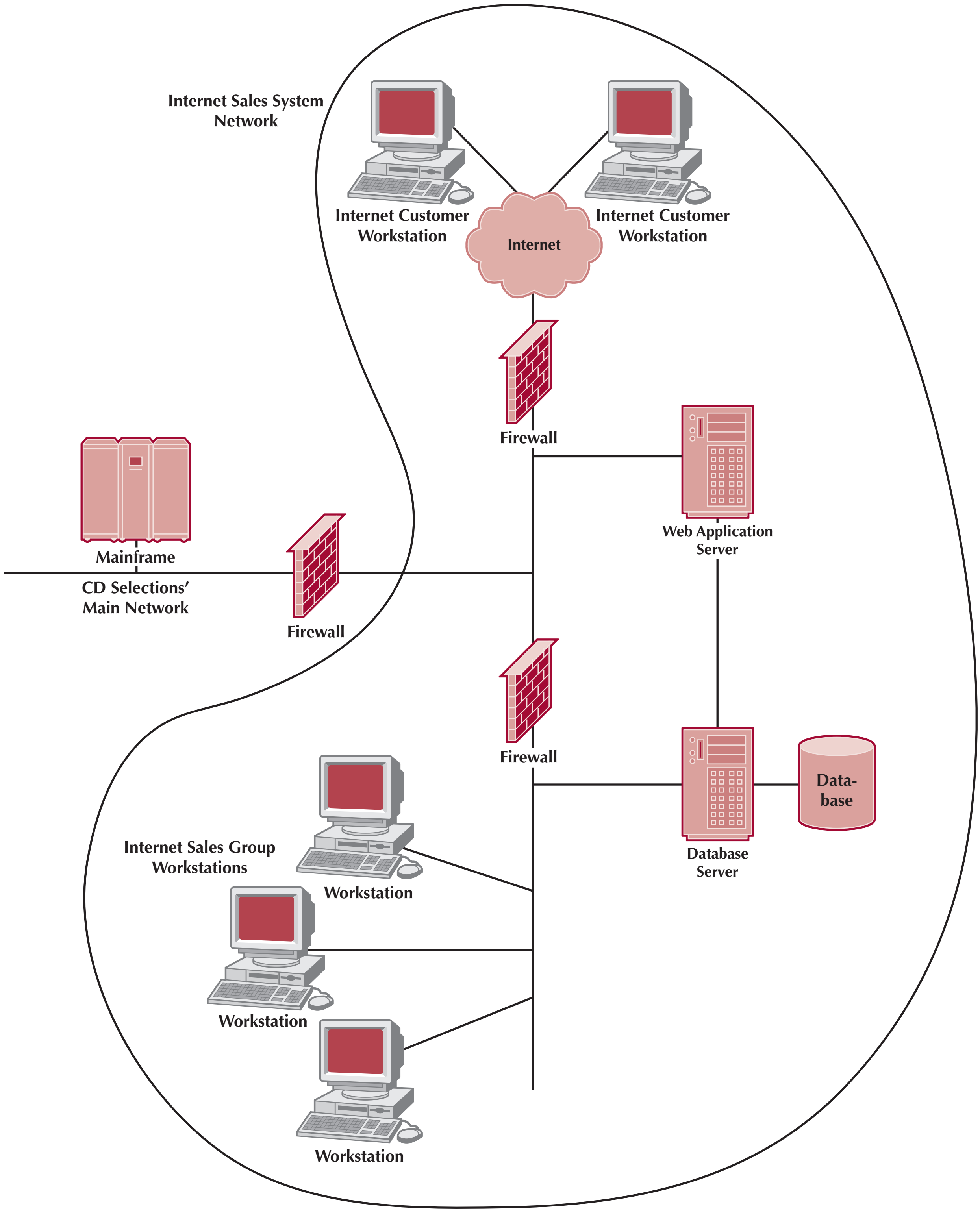

Internet Sales System Network
Internet Customer Workstation
Internet Customer Workstation
Internet
Firewall
Web Application Server
Mainframe
CD Selections' Main Network
Firewall
Firewall
Database Server
Data-base
Internet Sales Group Workstations
Workstation
Workstation
Workstation

FIGURE 11-B Deployment Diagram of Network Model for the CD Selections Internet Sales System

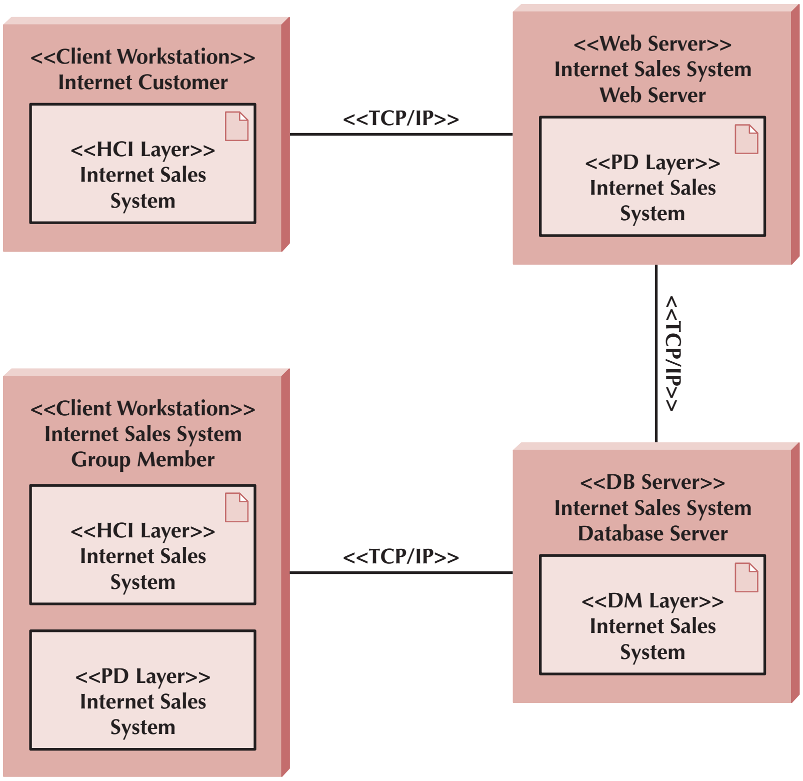


FIGURE 11-C Deployment Diagram of Layers for the CD Selections Internet Sales System

exchange a lot of data. Based on these decisions, she also created a deployment diagram that shows how the problem domain, human computer interaction, and data management layers would be deployed over the physical architecture layer (see Figure 11-C).

Given that the Web interface could reach a geographically dispersed group, the development team realized that it needed to plan for 24/7 system support. As such, Anne scheduled a meeting to talk with the CD Selections systems operations group and discussed how they might be able to support the Internet system outside of standard working hours.

After examining the network model, the architecture group and the development team decided that the only components that needed to be acquired for the project were a database server, a Web server, and five new client computers for the marketing group, who will maintain the CD marketing materials. They developed a hardware and software specification for these components and handed them off to the purchasing department to start the acquisition process.

CHAPTER 12: CONSTRUCTION

Because the material in this chapter actually takes place throughout the systems development process, this installment of the CD Selections case simply revisits some of the earlier installments and shows where this material has either already been described or where the development team should have performed these tasks.

Managing Programming

To keep the project on schedule, Anne requested that she be allowed to assign three programmers from the CD Selections programming staff to develop the three major parts of the Internet Sales System. The first was the Web interface, both the client side (browser) and the server side. The second was the client-server-based management system (managing the CD information and marketing materials data bases). The third was the interfaces between the Internet Sales System and CD Selections' existing distribution system and the credit card center. Programming went smoothly and despite a few minor problems went according to plan.

Testing

While Anne and the programmers were working, Alec began developing the test plans. The test plans for the three components were similar, but slightly more intensive for the Web interface component (see Figure 12-A). Unit testing would use black box testing based on the CRC cards, class diagrams, and contracts for all components. For example, the Order class was described in the chapter (see Figures 12-4, 12-6, and 12-7) and the CD class has been described in earlier installments (see Figures 8-C and 8-D).

Integration testing for the Web interface and system management component would be subjected to all user interface and use case tests to ensure the interface works properly. The system interface component would undergo system interface tests to ensure that the system performed calculations properly and was capable of exchanging data with the CD Selections' other systems and the credit card center.

Systems tests are by definition tests of the entire system—all components together. However, not all parts of the system would receive the same level of testing. Requirements tests would be conducted on all parts of the system to ensure that all requirements were met. Security was a critical issue, so the security of all aspects of the system would be tested. Security tests would be developed by CD Selections' infrastructure team, and once the system passed those tests, an external security consulting firm would be hired to attempt to break-in to the system.

Performance was an important issue for the parts of the system used by the customer (the Web interface and the system interfaces to the credit card and inventory systems) but not as important for the management component that would be used by staff, not customers. The customer-facing components would undergo rigorous performance testing to see how many transactions (whether searching or purchasing) they could handle before

Test Stage	Web Interface	System Management	System Interfaces
Unit tests	Black-box tests	Black-box tests	Black-box tests
Integration tests	User interface tests; use-case tests	User interface tests; use-case tests	System interface tests
System tests	Requirements tests; security tests; performance tests; usability tests	Requirements tests; security tests	Requirements tests; security tests; performance tests
Acceptance tests	Alpha test; beta test	Alpha test; beta test	Alpha test; beta test

FIGURE 12-A
Test Plan for CD Selections

they were unable to provide a response time of two seconds or less. Alec also developed an upgrade plan so that as demand on the system increased, there was a clear plan for when and how to increase the processing capability of the system.

Finally, formal usability tests would be conducted on the Web interface portion of the system with six potential users (both novice and expert Internet users).

Acceptance tests would be conducted in two stages, alpha and beta. Alpha tests would be done during the training of CD Selections' staff. The Internet Sales manager would work together with Alec to develop a series of tests and training exercises to train the Internet Sales group staff on how to use the system. They would then load the real CD data into the system and begin adding marketing materials. These same staff and other CD Selections staff members would also pretend to be customers and test the Web interface.

Beta testing would be done by "going live" with the Web site but only announcing its existence to CD Selections employees. As an incentive to try the Web site (rather than buying from the store in which they worked) employees would be offered triple their normal employee discount for all products ordered from the Web site. The site would also have a prominent button on every screen that would enable employees to e-mail comments to the project team, and the announcement would encourage employees to report problems, suggestions, and compliments to the project team. After one month, assuming all went well, the beta test would be completed, and the Internet Sales site linked to main Web site and advertised to the general public.

Developing User Documentation

While Anne and Alec were busy with the programming and test plans, Brian began the process of developing all necessary documentation. There were three types of documentation (reference documents, procedures manuals, and tutorials) that needed to be produced for the Web interface and the management component. Since the number of CD Selections staff using the system management component would be small, Brian suggested that they only produce the reference documentation (an online help system). After talking with Alec and Anne, the team felt that an intensive training program and a one month beta test period would be sufficient without tutorials and formal procedures manuals. Likewise, they felt that the process of ordering CDs and the Web interface itself was simple enough to not require a tutorial on the Web—a help system would be sufficient (and a procedures manual didn't make sense).

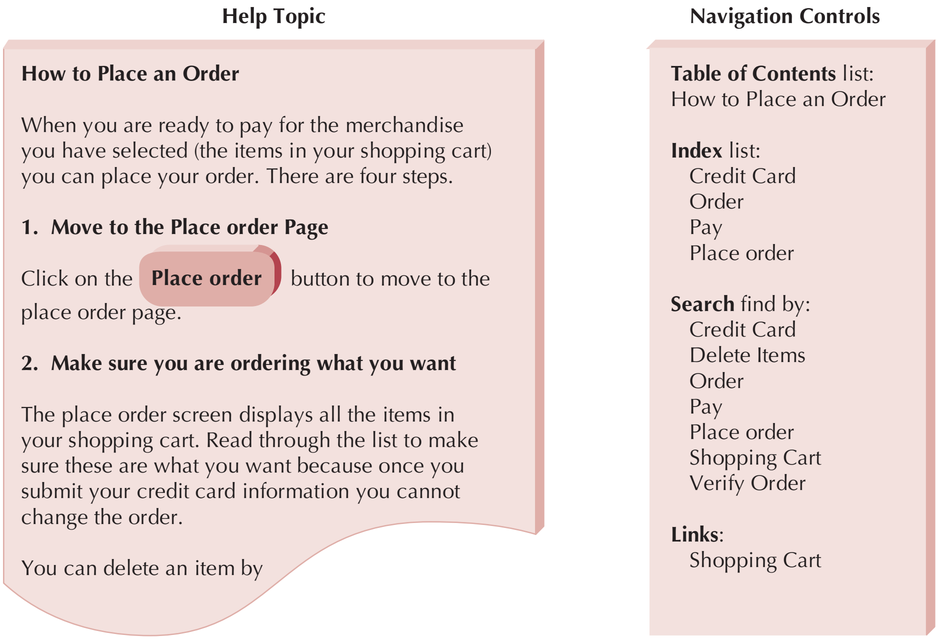
Brian decided that the reference documents for both the Web interface and system management components would contain help topics for user tasks, commands, and definitions. He also decided that the documentation component would contain four types of navigation controls: a table of contents, an index, a find, and links to definitions. He did not feel that the system was complex enough to benefit from a search agent.

After these decisions were discussed with the development team, Brian delegated the development of the reference documents to a technical writer assigned to the project team. Figure 12-B shows examples of a few of topics that the writer developed. The tasks and

Tasks	Commands	Terms
Find an album	Find	Album
Add an album to my shopping cart	Browse	Artist
Placing an order	Quick search	Music type
How to buy	Full search	Special deals
What's in my shopping cart?		Cart
		Shopping cart

FIGURE 12-B
Sample Help Topics for CD Selections

FIGURE 12-C Sample Documentation Topic for CD Selections



Help Topic

How to Place an Order

When you are ready to pay for the merchandise you have selected (the items in your shopping cart) you can place your order. There are four steps.

1. Move to the Place order Page

Click on the **Place order** button to move to the place order page.

2. Make sure you are ordering what you want

The place order screen displays all the items in your shopping cart. Read through the list to make sure these are what you want because once you submit your credit card information you cannot change the order.

You can delete an item by

Navigation Controls

Table of Contents list:
How to Place an Order

Index list:
- Credit Card
- Order
- Pay
- Place order

Search find by:
- Credit Card
- Delete Items
- Order
- Pay
- Place order
- Shopping Cart
- Verify Order

Links:
- Shopping Cart

commands were taken directly from the interface design. The list of definitions were developed once the tasks and commands were developed based on the writer's experience in understanding what terms might be confusing to the user.

Once the topic list was developed, the technical writer then began writing the topics themselves and the navigation controls to access. Figure 12-C shows an example of one topic taken from the task list: how to place an order. This topic presents a brief description of what it is and then leads the user through the step by step process needed to complete the task. The topic also lists the navigation controls that will be used to find the topic, in terms of the table of contents entries, the index entries, and search entries. It also lists what words in the topic itself will have links to other topics (e.g., shopping cart).

CHAPTER 13: INSTALLATION AND OPERATIONS

In this installment of the CD Selections case, we see how the new system is transitioned from the development team and put into production by the user community. To ensure a smooth transition, Alec and Margaret oversaw the necessary user training, including employees from CD Selections help desk department, and the creation of the necessary, relevant documentation. Looking back over the development of the system, Alec and Margaret evaluate the processes used and the individual development team members to identify lessons learned throughout the process. Finally, they set up a process to maintain the system.

The installation of the Internet Sales System at CD Selections was somewhat simpler than the installation of most systems because the system was entirely new; there was no As-Is system for the new system to replace. Also, there were not a large number of staff members who needed to be trained on the operation of the new system.

Conversion

Conversion went smoothly. First, the new hardware was purchased and installed. Then the software was installed on the Web server and on the client computers to be used by the staff of the Internet sales group. There was no data conversion per se, although the system started receiving data downloaded from the distribution system every day as it would during normal operations.

Alec decided on a direct conversion (because there was no As-Is system) in the one location (because there was only one location) of all system modules. The conversion, if you could call it that, went smoothly through the alpha and beta tests and the system was declared technically ready for operation.

Change Management

There were few change management issues because there were no existing staff members who had to change. All new staff was hired, most by internal transfer from other groups within CD Selections. The most likely stakeholders to be concerned by the change would be managers and employees in the traditional retail stores who might see the Internet Sales System as a threat to their stores. Alec therefore developed an information campaign (distributed through the employee newsletter and internal Web site) that discussed the reasons for the change and explained that the Internet Sales System was seen as a complement to the existing stores, not as a competitor. The system was instead targeted at Web-based competitors, such as Amazon.com.

The new management policies were developed, along with a training plan that encompassed both the manual work procedures and computerized procedures. Alec decided to use classroom training for the Internet sales system personnel because there was a small number of them and it was simpler and more cost effective to train them all together in one classroom session.

Post-Implementation Activities

Support of the system was turned over to the CD Selections operations group, who had hired four additional support staff members with expertise in networking and Web-based systems. System maintenance began almost immediately, with Alec designated as the project manager responsible for maintenance of this version of the system plus the development of the next version. Alec began the planning to develop the next version of the system.

Project team review uncovered several key lessons learned, mostly involving Web-based programming and the difficulties in linking to existing Structured Query Language

(SQL) databases. The project was delivered on budget (see Figure 2-C), with the exception that more was spent on programming than was anticipated.

A preliminary system review was conducted after two months of operations. Sales were $40,000 for the first month and $60,000 for the second, showing a gradual increase (remember that the goal for the first year of operations was $1,000,000). Operating expenses averaged $60,000 per month, a bit higher than the projected average, owing to startup costs and the initial marketing campaign. Nonetheless, Margaret Mooney, vice president of marketing and the project sponsor, was quite pleased. She approved the feasibility study for the follow-on project to develop the second version of the Internet Sales System. If you recall, there were quite a few features in which Margaret was interested. For example, deployment on mobile platforms, the use of social media, and the internationalization of the system was delayed until the next version of the system. Consequently, Alec got right to work to help Margaret create the system request so that the system development process could start all over again.