About Face 3

The Essentials of Interaction Design

An international bestseller,
now completely revised and updated

Alan Cooper,
Robert Reimann, and David Cronin

About Face 3

The Essentials of Interaction Design

Alan Cooper, Robert Reimann, and Dave Cronin

Wiley Publishing, Inc.

About Face 3

About Face 3

The Essentials of Interaction Design

Alan Cooper, Robert Reimann, and Dave Cronin

Wiley Publishing, Inc.

About Face 3: The Essentials of Interaction Design
Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada
ISBN: 978-0-470-08411-3
Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data:
Cooper, Alan, 1952-
About face 3 : the essentials of interaction design / Alan Cooper, Robert Reimann, and Dave Cronin.
p. cm.
Includes bibliographical references.
ISBN 978-0-470-08411-3 (pbk.)
1. User interfaces (Computer systems) 2. Human-computer interaction. I. Reimann, Robert. II. Cronin, Dave, 1972- III. Title. IV. Title: About face three.
QA76.9.U83C6596 2007
005.4'38--dc22
2007004977

For Sue, my best friend through all the adventures of life.

For Maxwell Aaron Reimann.

For Gretchen.

And for Cooperistas past, present, and future;
and for those visionary IxD practitioners who
have helped create a new design profession.

About the Authors

Alan Cooper is a pioneering software inventor, programmer, designer, and theorist. He is credited with having produced "probably the first serious business software for microcomputers" and is well known as the "Father of Visual Basic." For the last 15 years his software design consulting company, Cooper, has helped many companies invent new products and improve the behavior of their technology. At Cooper, Alan led the development of a new methodology for creating successful software that he calls the Goal-Directed process. Part of that effort was the invention of personas, a practice that has been widely adopted since he first published the technique in his second book, *The Inmates are Running the Asylum*, in 1998. Cooper is also a well known writer, speaker, and enthusiast for humanizing technology.

Robert Reimann has spent the past 15 years pushing the boundaries of digital products as a designer, writer, lecturer, and consultant. He has led dozens of interaction design projects in domains including e-commerce, portals, desktop productivity, authoring environments, medical and scientific instrumentation, wireless, and handheld devices for startups and Fortune 500 clients alike. As director of design R&D at Cooper, Reimann led the development and refinement of many of the Goal-Directed Design methods described in *About Face*. In 2005, Reimann became the first President of IxDA, the Interaction Design Association (`www.ixda.org`), a global nonprofit professional organization for Interaction Designers. He is currently manager of user experience at Bose Corporation.

Dave Cronin is the director of interaction design at Cooper, where he's helped design products to serve the needs of people such as surgeons, museum visitors, marketers, investment portfolio managers, online shoppers, hospital staff, car drivers, dentists, financial analysts, manufacturing planners, the elderly, and the infirm. At Cooper, he has also contributed substantially to the ongoing process of developing and refining the Goal-Directed Design methods described in this book.

Credits

Executive Editor
Chris Webb

Development Editors
Sara Shlaer
Sydney Jones

Production Editor
Eric Charbonneau

Copy Editor
Foxxe Editorial Services

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator
Erin Smith

Graphics and Production Specialists
Sean Decker, Brooke Graczyk, Stephanie D. Jumper, Jennifer Mayberry, Barbara Moore, Ronald Terry

Quality Control Technician
Christy Pingleton

Book Designers
Rebecca Bortman and Nick Myers

Illustrators
Rebecca Bortman and Nick Myers

Proofreading and Indexing
Aptara

Anniversary Logo Design
Richard Pacifico

Cover Design
Rebecca Bortman and Nick Myers

Contents

5

Modeling Users: Personas and Goals

Having gone out into the wide world to understand your users' lives, motivations, and environs, a big question arises: How do you use this research data to come up with a design that will result in a successful product? You have notebooks full of conversations and observations, and it is likely that each person you spoke to was slightly different from the others. It is difficult to imagine digging through hundreds of pages of notes every time you have to make a design decision, and even if you had the time to do this, it isn't entirely obvious how these notes should inform your thinking.

We solve this problem by applying the powerful concept of a **model**. Models are used in the natural and social sciences to represent complex phenomena with a useful abstraction. Much as economists create models to describe the behavior of markets, and physicists create models to describe the behavior of particles, we have found that using our research to create descriptive models of our users is a uniquely powerful tool for interaction design. We call these user models **personas**.

Personas provide us with a precise way of thinking and communicating about how users behave, how they think, what they wish to accomplish, and why. Personas are not real people, but they are based on the behaviors and motivations of real people we have observed and represent them throughout the design process. They are

composite archetypes based on behavioral data gathered from the many actual users encountered in ethnographic interviews. Personas are based upon *behavior patterns we observe* during the course of the Research phase, which we then formalize in the Modeling phase. By using personas, we can develop an understanding of our users' goals in specific contexts — a critical tool for using user research to inform and justify our designs.

Personas, like many powerful tools, are simple in concept but must be applied with considerable sophistication. It is not enough to whip up a couple of user profiles based upon stereotypes and generalizations, nor is it particularly useful to attach a stock photograph to a job title and call it a "persona." For personas to be effective tools for design, considerable rigor and finesse must be applied to the process of identifying the significant and meaningful patterns in user behavior and turning these into archetypes that represent a broad cross-section of users.

While there are other useful models that can serve as tools for the interaction designer, such as workflow models and physical models, we've found that personas are the strongest, and it is possible to incorporate the best from other modeling techniques into a persona. This chapter focuses on personas and their goals. Other models are considered briefly at the end of the chapter.

Why Model?

Models are used extensively in design, development, and the sciences. They are powerful tools for representing complex structures and relationships for the purpose of better understanding, discussing, or visualizing them. Without models, we are left to make sense of unstructured, raw data, without the benefit of any organizing principle. Good models emphasize the salient features of the structures and relationships they represent and de-emphasize the less significant details.

Because we are designing for users, it is important that we can understand and visualize the salient aspects of their relationships with each other, with their social and physical environments, and of course, with the products we hope to design.

Just as physicists have created models of the atom based on observed data and intuitive synthesis of the patterns in their data, so must designers create models of users based on observed behaviors and intuitive synthesis of the patterns in the data. Only after we formalize such patterns can we hope to systematically construct patterns of interaction that smoothly match the behavior patterns, mental models, and goals of users. Personas provide this formalization.

Personas

To create a product that must satisfy a diverse audience of users, logic might tell you to make it as broad in its functionality as possible to accommodate the most people. *This logic, however, is flawed.* The best way to successfully accommodate a variety of users is to design for *specific types of individuals with specific needs*.

When you broadly and arbitrarily extend a product's functionality to include many constituencies, you increase the cognitive load and navigational overhead for all users. Facilities that may please some users will likely interfere with the satisfaction of others (see Figure 5-1).

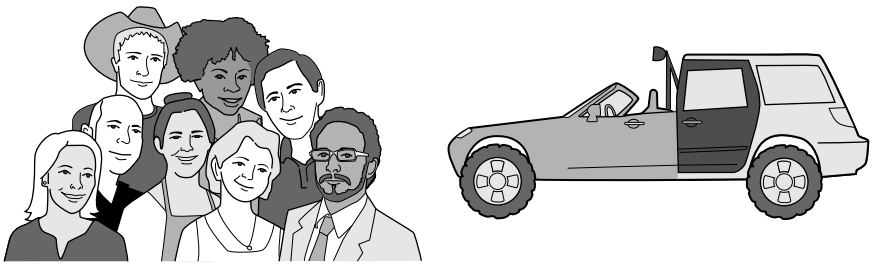

Figure 5-1 A simplified example of how personas are useful. If you try to design an automobile that pleases every possible driver, you end up with a car with every possible feature, but that pleases nobody. Software today is too often designed to please too many users, resulting in low user satisfaction. Figure 5-2 provides an alternative approach.

The key to this approach is first to choose the right individuals to design for — those users whose needs best represent the needs of a larger set of key constituents (see Figure 5-2) — and then to prioritize these individuals so that the needs of the most important users are met without compromising our ability to meet the needs of secondary users. Personas provide a powerful tool for communicating about different types of users and their needs, then deciding which users are the most important to target in the design of form and behavior.

Since they were introduced as a tool for user modeling in *The Inmates are Running The Asylum*,[1] personas have gained great popularity in the user experience community, but they have also been the subject of some misunderstandings. We'd like to clarify and explain in more depth some of the concepts and the rationale behind personas.

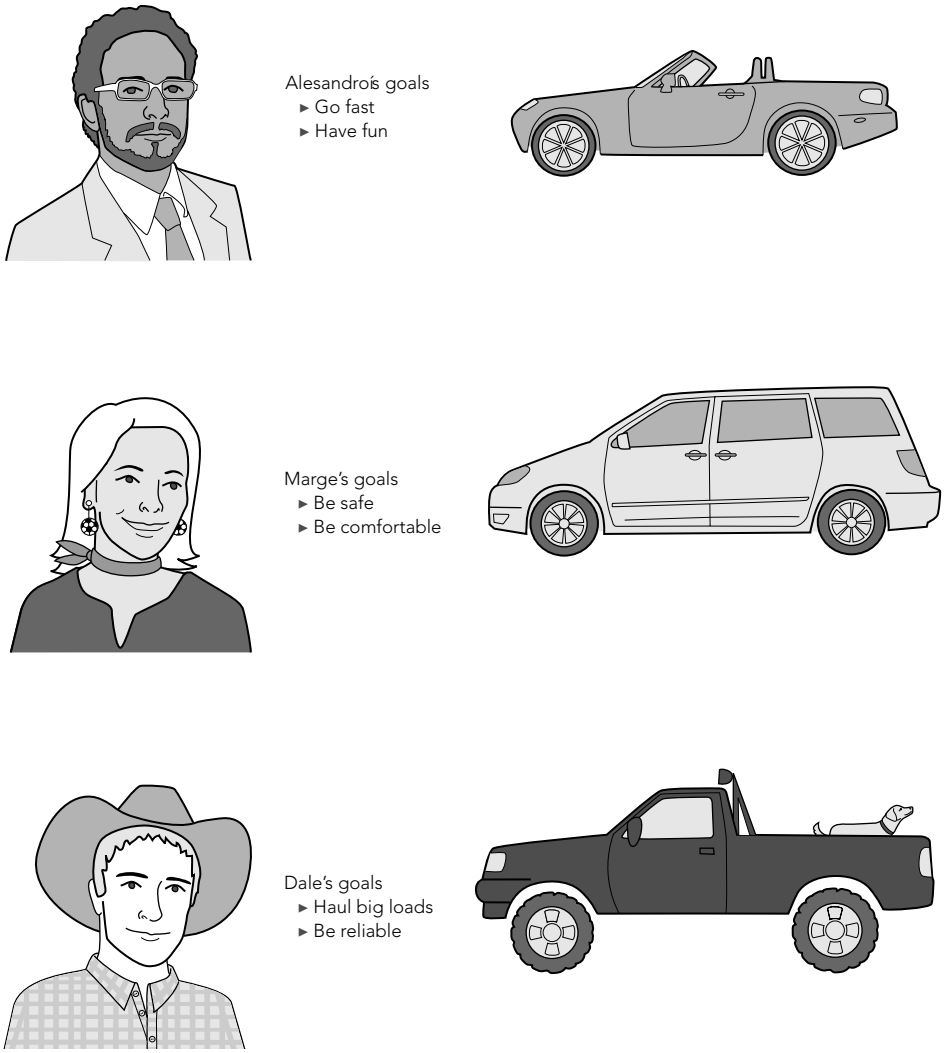

Figure 5-2 A simplified example of how personas are useful. By designing different cars for different people with different specific goals, we are able to create designs that other people with similar needs to our target drivers also find satisfying. The same holds true for the design of digital products and software.

Strengths of personas as a design tool

The persona is a powerful, multipurpose design tool that helps overcome several problems that currently plague the development of digital products. Personas help designers:

- **Determine** what a product should do and how it should behave. Persona goals and tasks provide the foundation for the design effort.
- **Communicate** with stakeholders, developers, and other designers. Personas provide a common language for discussing design decisions and also help keep the design centered on users at every step in the process.
- **Build consensus and commitment** to the design. With a common language comes a common understanding. Personas reduce the need for elaborate diagrammatic models; it's easier to understand the many nuances of user behavior through the narrative structures that personas employ. Put simply, because personas resemble real people, they're easier to relate to than feature lists and flowcharts.
- **Measure** the design's effectiveness. Design choices can be tested on a persona in the same way that they can be shown to a real user during the formative process. Although this doesn't replace the need to test with real users, it provides a powerful reality-check tool for designers trying to solve design problems. This allows design iteration to occur rapidly and inexpensively at the whiteboard, and it results in a far stronger design baseline when the time comes to test with actual people.
- **Contribute** to other product-related efforts such as marketing and sales plans. The authors have seen their clients repurpose personas across their organization, informing marketing campaigns, organizational structure, and other strategic planning activities. Business units outside of product development desire sophisticated knowledge of a product's users and typically view personas with great interest.

Personas also can resolve three design issues that arise during product development:

- The elastic user
- Self-referential design
- Edge cases

We discuss each of these briefly in the following sections.

The elastic user

Although satisfying the users of our products is our goal, the term *user* causes trouble when applied to specific design problems and contexts. Its imprecision makes it dangerous as a design tool — every person on a product team has his own conceptions of who the user is and what the user needs. When it comes time to make product decisions, this "user" becomes *elastic*, conveniently bending and stretching to fit the opinions and presuppositions of whoever's talking.

If the product development team finds it convenient to use a confusing tree control containing nested, hierarchical folders to provide access to information, they might

define the user as a computer-literate "power user." Other times, when it is more convenient to step through a difficult process with a wizard, they define the user as an unsophisticated first-time user. Designing for the elastic user gives a product-development team license to build what it pleases, while still apparently serving "the user." Of course, our goal should be to design products that appropriately meet the needs of *real* users. Real users — and the personas representing them — are not elastic, but rather have specific requirements based on their goals, capabilities, and contexts.

Even focusing on user roles or job titles rather than specific archetypes can introduce unproductive elasticity to the focus of design activities. For example, in designing clinical products, it might be tempting to lump together all nurses as having similar needs. However, if you have any experience in a hospital, you know that trauma nurses, pediatric intensive-care nurses, and operating room nurses are quite different from each other, each with their own attitudes, aptitudes, needs, and motivations. A lack of precision about the user can lead to a lack of clarity about how the product should behave.

Self-referential design

Self-referential design occurs when designers or developers project their own goals, motivations, skills, and mental models onto a product's design. Many "cool" product designs fall into this category. The audience doesn't extend beyond people like the designer, which is fine for a narrow range of products and completely inappropriate for most others. Similarly, programmers apply self-referential design when they create implementation-model products. *They* understand perfectly how the data is structured and how software works and are comfortable with such products. Few nonprogrammers would concur.

Edge cases

Another syndrome that personas help prevent is designing for edge cases — those situations that might possibly happen, but usually won't for the target personas. Typically, edge cases must be designed and programmed for, but they should never be the design *focus*. Personas provide a reality check for the design. We can ask, "Will Julie want to perform this operation very often? Will she ever?" With this knowledge, we can prioritize functions with great clarity.

Personas are based on research

Personas, like any models, must be based on real-world observation. As discussed in the preceding chapter, the primary source of data used to synthesize personas should be in-context interviews borrowing from ethnographic techniques, contextual

inquiry, or other similar dialogues with and observation of actual and potential users. The quality of the data gathered following the process (outlined in Chapter 4) directly impacts the efficacy of personas in clarifying and directing design activities. Other data that can support and supplement the creation of personas include (in rough order of effectiveness):

- Interviews with users outside of their use contexts
- Information about users supplied by stakeholders and subject matter experts (SMEs)
- Market research data such as focus groups and surveys
- Market-segmentation models
- Data gathered from literature reviews and previous studies

However, none of this supplemental data can take the place of direct user interviews and observation. Almost every aspect of a well-developed persona can be traced back to a user statement or behavior.

Personas are represented as individual people

Personas are user models that are represented as specific, individual human beings. They are not actual people but are synthesized directly from observations of real people. One of the key elements that allow personas to be successful as user models is that they are *personifications*.[2] This is appropriate and effective because of the unique aspects of personas as user models: They engage the *empathy* of the design and development towards the human target of the design.

Empathy is critical for the designers, who will be making their decisions for design frameworks and details based on both the cognitive *and* emotional dimensions of the persona, as typified by the persona's goals. (We will discuss the important connections between goals, behaviors, and personas later in this chapter.) However, the power of empathy should not be quickly discounted for other team members. Not only do personas help make our design solutions better at serving real user needs, but they also make these solutions more compelling to stakeholders. When personas have been carefully and appropriately crafted, stakeholders and engineers begin to think about them as if they are real human beings and become much more interested in creating a product that will give this person a satisfying experience.

We're all aware of the power of fictional characters in books, movies, and television programs to engage viewers. Jonathan Grudin and John Pruitt have discussed how this can relate to interaction design.[3] They note, as well, the power of **method**

acting as a tool that actors use to understand and portray realistic characters. In fact, the process of creating personas from user observation, and then imagining and developing scenarios from the perspective of these personas, is, in many ways, analogous to method acting. (We've even heard our Goal-Directed use of personas referred to as the Stanislavsky Method of interaction design.)

Personas represent groups of users

Although personas are depicted as specific individuals, because they function as archetypes, they *represent* a class or type of user of a *specific* interactive product. A persona encapsulates a distinct set of **behavior patterns** regarding the use of a particular product (or analogous activities if a product does not yet exist), which are identified through the analysis of interview data, and supported by supplemental quantitative data as appropriate. These patterns, along with specific motivations or goals, define our personas. Personas are also sometimes referred to as **composite user archetypes** because personas are in a sense composites assembled by grouping related usage patterns observed across individuals in similar roles during the Research phase.[4]

Personas and reuse

Organizations with more than one product often want to reuse the same personas. However, to be effective, personas must be context specific — they should be focused on the behaviors and goals related to the specific domain of a particular product. Personas, because they are constructed from specific observations of users interacting in specific contexts, cannot easily be reused across products even when those products form a closely linked suite.[5]

For a set of personas to be an effective design tool for multiple products, the personas must be based upon research concerning the usage contexts for all of these products. In addition to broadening the scope of the research, an even larger challenge is to identify manageable and coherent sets of behavior patterns across all of the contexts. Clearly, it is a fallacy to believe that just because two users exhibit similar behaviors in regard to one product, that those two users would behave similarly with respect to a different product. Thus, as focus expands to encompass more and more products, it becomes increasingly difficult to create a concise and coherent set of personas that represents the diversity of real-world users. We've found that, in most cases, personas should be researched and developed individually for different products.

Archetypes versus stereotypes

Don't confuse persona archetypes with **stereotypes**. Stereotypes are, in most respects, the antithesis of well-developed personas. Stereotypes represent designer

or researcher biases and assumptions, rather than factual data. Personas developed by drawing on inadequate research (or synthesized with insufficient empathy and sensitivity to interview subjects) run the risk of degrading to stereotypical caricatures. Personas must be developed and treated with dignity and respect for the people whom they represent. If the designer doesn't respect his personas, nobody else will either.

Personas also bring issues of social and political consciousness to the forefront.[6] Because personas provide a precise design target and also serve as a communication tool to the development team, the designer much choose particular demographic characteristics with care. Ideally, persona demographics should be a composite reflection of what researchers have observed in the interview population, modulated by broad market research. Personas should be *typical* and believable, but not stereotypical. If the data is not conclusive or the characteristic is not important to the design or its acceptance, we prefer to err on the side of gender, ethnic, age, and geographic diversity.

Personas explore ranges of behavior

The target market for a product describes demographics as well as lifestyles and sometimes job roles. What it does not describe are the ranges of different behaviors exhibited by members of that target market regarding the product and related situations. Ranges are distinct from *averages*: Personas do not seek to establish an average user, but rather to express *exemplary* or definitive behaviors within these identified ranges.

Because products must accommodate *ranges* of user behavior, attitudes and aptitudes, designers must identify a **persona set** associated with any given product. Multiple personas carve up ranges of behavior into discrete clusters. Different personas represent different correlated behavior patterns. These correlations are arrived at through analyzing research data. This process of identifying behaviors is discussed in greater detail later in this chapter.

Personas must have motivations

All humans have motivations that drive their behaviors; some are obvious, and many are subtle. It is critical that personas capture these motivations in the form of goals. The goals we enumerate for our personas (discussed at length later in this chapter) are shorthand notation for motivations that not only point at specific usage patterns but also provide a reason why those behaviors exist. Understanding

why a user performs certain tasks gives designers great power to improve or even eliminate those tasks yet still accomplish the same goals.

Personas can also represent nonusers

While the users and potential users of a product should always be an interaction designer's primary concern, it is sometimes useful to represent the needs and goals of people who do not use the product but nevertheless must be considered in the design process. For example, it is commonly the case with enterprise software (and children's toys) that the person who purchases the product is not the same person who uses it. In these cases, it may be useful to create one or more **customer personas**, distinct from the set of user personas. Of course, these should also be based upon behavior patterns observed through ethnographic research, just as user personas are.

Similarly, for many medical products, patients do not directly interact with the user interface, but they have motivations and objectives that may be very different than the clinician using the product. Creating a **served persona** to represent patients' needs can be useful in these cases. We discuss served and customer personas in greater depth later in this chapter.

Personas and other user models

There a number of other user models commonly employed in the design of interactive products, including user roles, user profiles, and market segments. These are similar to personas in that they seek to describe users and their relationship to a product. However, personas and the methods by which they are created and employed as a design tool differ significantly from these in several key aspects.

User roles

A user role or role model, as defined by Larry Constantine, is an *abstraction*, a defined relationship between a class of users and their problems, including needs, interests, expectations, and patterns of behavior.[7] As abstractions (generally taking the form of a list of attributes), they are not imagined as people, and do not typically attempt to convey broader human motivations and contexts.

Holtzblatt and Beyer's use of roles in consolidated flow, cultural, physical, and sequence models is similar in that it attempts to abstract various attributes and relationships abstracted from the people possessing them.[8]

We find these methods limiting for several reasons:

- It is more difficult to clearly communicate human behaviors and relationships in the abstract, isolated from people who possess them. The human power of empathy cannot easily be brought to bear on abstract classes of people.
- Both methods focus on *tasks* almost exclusively and neglect the use of goals as an organizing principle for design thinking and synthesis.
- Holtzblatt and Beyer's consolidated models, although useful and encyclopedic in scope, are difficult to bring together as a coherent tool for developing, communicating, and measuring design decisions.

Personas address each of these problems. Well-developed personas describe the same type of behaviors and relationships that user roles do, but express them in terms of goals and examples in narrative. This makes it possible for designers and stakeholders to understand the implications of design decisions in human terms. Describing a persona's goals provides context and structure for tasks, incorporating how culture and workflow influence behavior.

In addition, focusing on user roles rather than on more complex behavior patterns can oversimplify important distinctions and similarities between users. It is possible to create a persona that represents the needs of several user roles (for example, in designing a mobile phone, a traveling salesperson might also represent the needs of a busy executive who's always on the road), and it is also possible that there are several people in the same role who think and act differently (perhaps a procurement planner in the chemical industry thinks about her job very differently from a procurement planner in the consumer electronics industry). In consumer domains, roles are next to useless. If you're designing a Web site for a car company, "car buyer" is meaningless as a design tool — different people approach the task in very different manners.

In general, personas provide a more holistic model of users and their contexts, where many other models seek to be more reductive. Personas can certainly be used in combination with these other modeling techniques, and as we'll discuss at the end of the chapter, some other models make extremely useful complements to personas.

Personas versus user profiles

Many usability practitioners use the terms **persona** and **user profile** synonymously. There is no problem with this if the profile is truly generated from ethnographic data and encapsulates the depth of information the authors have described. Unfortunately, all too often, the authors have seen user profiles that reflect Webster's definition of **profile** as a "brief biographical sketch." In other words, user profiles often consist of a name and a picture attached to a brief, mostly demographic

description, along with a short, *fictional* paragraph describing the kind of car this person drives, how many kids he has, where he lives, and what he does for a living. This kind of user profile is likely to be based on a stereotype and is not useful as a design tool. Although we give our personas names, and sometimes even cars and family members, these are employed sparingly as narrative tools to help better communicate the real underlying data. Supporting fictional detail plays only the most minor part in persona creation and is used just enough to make the persona come to life in the minds of the designers and the product team.

Personas versus market segments

Marketing professionals may be familiar with a process similar to persona development because it shares some process similarities with market definition. The main difference between market segments and design personas is that the former are based on demographics, distribution channels, and purchasing behavior, whereas the latter are based on usage behavior and motivations. The two are not the same and don't serve the same purpose. Marketing personas shed light on the sales process, whereas design personas shed light on the product definition and development process.

However, market segments play a role in persona development. They can help determine the demographic range within which to frame the persona hypothesis (see Chapter 4). Personas are segmented along ranges of usage behavior, not demographics or buying behavior, so there is seldom a one-to-one mapping of market segments to personas. Rather, market segments can act as an initial filter to limit the scope of interviews to people within target markets (see Figure 5-3). Also, we typically use the prioritization of personas as a way to make strategic product definition decisions (see the discussion of persona types later in this chapter). These decisions should incorporate market intelligence; an understanding of the relationship between user personas and market segments can be an important consideration here.

When rigorous personas aren't possible: Provisional personas

Although it is highly desirable that personas be based upon detailed qualitative data, there are some occasions when there simply is not enough time, resources, or corporate buy-in to perform the necessary fieldwork. In these cases, *provisional* personas (or, as Don Norman refers to them, "ad hoc" personas) can be useful rhetorical tools to clearly communicate assumptions about who the important users are and what they need, and to enforce rigorous thinking about serving specific user needs (even if these needs are not validated).

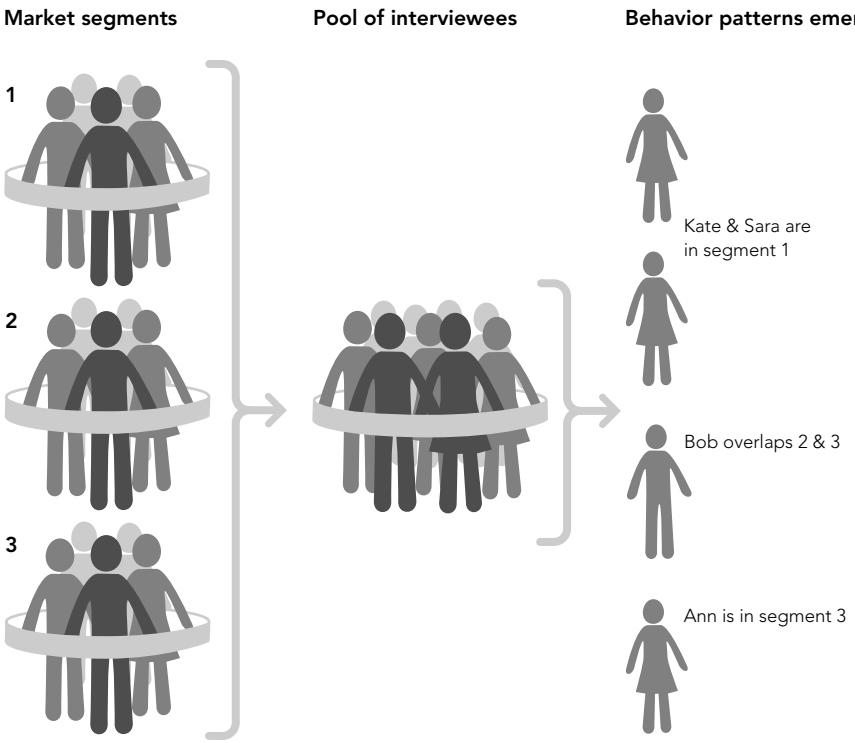

Figure 5-3 Personas versus market segments. Market segments can be used in the Research phase to limit the range of personas to target markets. However, there is seldom a one-to-one mapping between market segments and personas.

Provisional personas are structured similarly to real personas but rely on available data and designer best guesses about behaviors, motivations, and goals. They are typically based on a combination of stakeholder and subject matter expert knowledge of users (when available), as well as what is understood about users from existing market data. Provisional personas are, in fact, a more fleshed-out persona hypothesis (as described in Chapter 4).

Our experience is that, regardless of a lack of research, using provisional personas yields better results than no user models at all. Like real personas, provisional personas can help focus the product team and build consensus around product features and behaviors. There are, however, caveats: Provisional personas are called this because they should be recognized as stand-ins for personas based on definitive qualitative data. While provisional personas may help focus your design and product team, if you do not have data to back up your assumptions you may:

- Focus on the wrong design target
- Focus on the right target, but miss key behaviors that could differentiate your product

- Have a difficult time getting buy-in from individuals and groups who did not participate in their creation
- Discredit the value of personas, causing your organization to reject the use of personas in the long term

If you are using provisional personas, it's important to:

- Clearly label and explain them as such
- Represent them visually with sketches, not photos, to reinforce their provisional nature
- Try to make use of as much existing data as possible (market surveys, domain research, subject matter experts, field studies, or personas for similar products)
- Document what data was used and what assumptions were made
- Steer clear of stereotypes (more difficult to do without field data)
- Focus on behaviors and motivations, not demographics

Goals

If personas provide the context for sets of observed behaviors, **goals** are the drivers behind those behaviors. A persona without goals can still serve as a useful communication tool, but it lacks utility as a design tool. User goals serve as a lens through which designers must consider the functions of a product. The function and behavior of the product must address goals via tasks — typically, as few tasks as absolutely necessary. Remember, tasks are only a means to an end; goals are that end.

Goals motivate usage patterns

People's or personas' goals motivate them to behave the way they do. Thus, goals not only provide an answer to why and how personas desire to use a product but also can serve as a shorthand in the designer's mind for the sometimes complex behaviors in which a persona engages and, therefore, for their tasks as well.

Goals should be inferred from qualitative data

You usually can't ask a person what his goals are directly. Either he won't be able to articulate them, or he won't be accurate or even perfectly honest. People simply aren't well prepared to answer such questions accurately. Therefore, designers and researchers need to carefully reconstruct goals from observed behaviors, answers to

other questions, nonverbal cues, and clues from the environment such as the titles of books on shelves. One of the most critical tasks in the modeling of personas is identifying goals and expressing them succinctly: Each goal should be expressed as a simple sentence.

User goals and cognitive processing

Don Norman's book *Emotional Design* introduced the idea that product design should address three different levels of cognitive and emotional processing, which he has called visceral, behavioral, and reflective. Norman's ideas, based on years of cognitive research, provide an articulated structure for modeling user responses to product and brand and a rational context for many intuitions long held by professional designers.

Norman's three levels of cognitive processing are:

- **Visceral** — The most immediate level of processing, in which we react to visual and other sensory aspects of a product that we can perceive before significant interaction occurs. Visceral processing helps us make rapid decisions about what is good, bad, safe, or dangerous. This is one of the most exciting types of human behavior, and one of the most challenging to effectively support with digital products. Malcolm Gladwell explores this level of cognitive processing in his book *Blink*. For even more in-depth study of intuitive decision making, see Gary Klein's *Sources of Power* or *Hare Brain, Tortoise Mind* by Guy Claxton.
- **Behavioral** — The middle level of processing that lets us manage simple, everyday behaviors, which according to Norman, constitute the majority of human activity. Norman states — and rightly so — that historically, interaction design and usability practices have nearly exclusively addressed this level of cognitive processing. Behavioral processing can *enhance* or *inhibit* both lower-level visceral reactions and higher-level reflective responses, and conversely, both visceral and reflective processing can enhance or inhibit behavioral processing.
- **Reflective** — The least immediate level of processing, which involves conscious consideration and reflection on past experiences. Reflective processing can enhance or inhibit behavioral processing but has no direct access to visceral reactions. This level of cognitive processing is accessible only via memory, not through direct interaction or perception. The most interesting aspect of reflective processing as it relates to design is that, through reflection, we are able to integrate our experiences with designed artifacts into our broader life experiences and, over time, associate meaning and value with the artifacts themselves.

Designing for Visceral Responses

Designing for the visceral level means designing what the senses initially perceive, before any deeper involvement with a product or artifact occurs. For most of us, that means designing visual appearance and motion, though sound can also play a role — think of the distinctive Mac power-up chord. Those of us designing devices may design for tactile sensations as well.

A misconception often arises when discussing visceral-level design: that designing for visceral response is about designing *beautiful* things. Battlefield software and radiation-therapy systems are just two examples where designing for beauty may not be the proper focus. Visceral design is actually about designing for affect — that is, eliciting the appropriate psychological or emotional response for a particular context — rather than for aesthetics alone. Beauty — and the feelings of transcendence and pleasure it evokes — is really only a small part of the possible affective design palette. For example, an MP3 player and an online banking system require very different affects. We can learn a great deal about affect from architecture, the cinema and stage, and industrial design.

However, in the world of consumer products and services, attractive user interfaces *are* typically appropriate. Interestingly, usability researchers have demonstrated that users initially judge attractive interfaces to be more usable, and that this belief often persists long after a user has gained sufficient experience with an interface to have direct evidence to the contrary.[9] Perhaps the reason for this is that users, encouraged by perceived ease of use, make a greater effort to learn what may be a challenging interface and are then unwilling to consider their investment ill spent. For the scrupulous designer, this means that, when a user interface promises ease of use at the visceral level — or whatever else the visceral promise of an interaction may be — it should then be sure to deliver on that promise at the behavioral level.

Designing for Behavior

Designing for the behavioral level means designing product behaviors that complement a user's own behaviors, implicit assumptions, and mental models. Of the three levels of design Norman contemplates, behavioral design is perhaps the most familiar to interaction designers and usability professionals.

One intriguing aspect of Norman's three-level model as it relates to design is his assertion that behavioral processing, uniquely among his three levels, has direct influence upon and is influenced directly by both of the other two levels of processing. This would seem to imply that the day-to-day behavioral aspects of interaction design should be the primary focus of our design efforts, with visceral and reflective considerations playing a supporting role. Getting design of behavior

right — assuming that we also pay adequate attention to the other levels — provides our greatest opportunity for positively influencing the way users construct their experience with products.

Not following this line of reasoning can lead to the problem of users' initial impressions being out of sync with reality. Also, it is difficult to imagine designing for reflective meaning in memory without a solid purpose and set of behaviors in place for the here and now. The user experience of a product or artifact, therefore, should ideally *harmonize elements of visceral design and reflective design with a focus on behavioral design*.

Designing for Reflection

Reflective processing — and, particularly, what it means for design — is perhaps the most challenging aspect of the three levels of processing that Norman discusses. What is clear is that designing for the reflective level means designing to build long-term product relationships. What isn't clear at all is the best way to ensure success — if that's even possible — at the reflective level. Is it chance that drives success here — being in the right place at the right time — or can premeditated design play a part in making it happen?

In describing reflective design, Norman uses several high-concept designs for commodity products as examples — such as impractically configured teapots and the striking Phillipe Starck juicer that graces the cover of his book. It is easy to see how such products — whose value and purpose are, in essence, the aesthetic statements they make — could appeal strongly to people's reflective desire for uniqueness or cultural sophistication that perhaps may come from an artistic or stylish self-image.

It is more difficult to see how products that also serve a truly useful purpose need to balance the stylistic and the elegant with the functional. The Apple iPod comes very close to achieving this balance. Although its click-wheel navigation scheme is perhaps less than optimal in some respects, users' visceral reaction to the product is tremendous, due to its elegant industrial design. Its reflective potential is also significant, because of the powerful emotional connection people experience with their music. It's a winning combination that no competitor has yet been able to challenge.

Few products become iconic in people's lives in the way that, say, the Sony Walkman or the iPod has. Clearly there are some products that stand little chance of ever becoming symbolic in peoples lives — like Ethernet routers, for instance — no matter how wonderful they look or how well they behave. However, when the design of a product or service addresses users' goals and motivations — possibly going beyond the product's primary purpose, yet somehow connected to it via personal or cultural associations — the opportunity for the creation of reflective meaning is greatly enhanced.

The three types of user goals

In *Emotional Design*, Norman presents his three-level theory of cognitive processing and discusses its potential importance to design. However, Norman does not suggest a method for systematically integrating his model of cognition and affect into the practice of design or user research. In our practice, we've found that the key to doing so lies in properly delineating and modeling three specific types of user goals as part of each persona's definition.[10]

Three types of user goals correspond to Norman's visceral, behavioral, and reflective processing levels:

- Experience goals
- End goals
- Life goals

We describe each of these in detail in the following sections.

Experience goals

Experience goals are simple, universal, and personal. Paradoxically, this makes them difficult for many people to talk about, especially in the context of impersonal business. Experience goals express how someone *wants to feel* while using a product or the quality of their interaction with the product. These goals provide focus for a product's visual and aural characteristics, its interactive feel — such as animated transitions, latency, and the snap ratio (clickiness) of a physical button — and its physical design by providing insights into persona motivations that express themselves at the visceral level. For example:

- Feel smart or in control
- Have fun
- Feel cool or hip or relaxed
- Remain focused and alert

When products make users feel stupid or uncomfortable, their self-esteem drops and their effectiveness plummets, regardless of their other goals. Their level of resentment also increases. Enough of this type of treatment and users will be primed to use any chance to subvert the system. Any product that egregiously violates experience goals will ultimately fail, regardless of how well it purports to achieve other goals.

Interaction, visual, and industrial designers must translate persona experience goals into form, behavior, motion, and auditory elements that communicate the proper feel, affect, emotion, and tone. Visual language studies, as well as mood or inspiration boards, which attempt to establish visual themes based on persona attitudes and behaviors, are a useful tool for defining the tonal expectations of personas.

End goals

End goals represent the user's motivation for performing the tasks associated with using a specific product. When you pick up a cell phone or open a document with a word processor, you likely have an outcome in mind. A product or service can help accomplish such goals directly or indirectly. These goals are the focus of a product's interaction design, information architecture, and the functional aspects of industrial design. Because behavioral processing influences both visceral and reflective responses, end goals should be among the most significant factors in determining the overall product experience. End goals must be met for users to think that a product is worth their time and money.

Examples of end goals include:

- Be aware of problems before they become critical
- Stay connected with friends and family
- Clear my to-do list by 5:00 every day
- Find music that I'll love
- Get the best deal

Interaction designers must use end goals as the foundation for a product's behaviors, tasks, look, and feel. Context or day-in-the-life scenarios and cognitive walkthroughs are effective tools for exploring users' goals and mental models, which, in turn, facilitate appropriate behavioral design.

Life goals

Life goals represent personal aspirations of the user that typically go beyond the context of the product being designed. These goals represent deep drives and motivations that help explain *why* the user is trying to accomplish the end goals he seeks to accomplish. Life goals describe a persona's long-term desires, motivations, and self-image attributes, which cause the persona to connect with a product. These goals form the focus for a product's overall design, strategy, and branding. For example:

- Live the good life
- Succeed in my ambitions to . . .

- Be a connoisseur of . . .
- Be attractive, popular, or respected by my peers

Interaction designers must translate life goals into high-level system capabilities, formal design concepts, and brand strategy. Mood boards and context scenarios can be helpful in exploring different aspects of product concepts, and broad ethnographic research and cultural modeling are critical for discovering users' behavior patterns and deeper motivations. Life goals rarely figure directly into the design of specific elements or behaviors of an interface. However, they are very much worth keeping in mind. A product that the user discovers will take him closer to his life goals, and not just his end goals, will win him over more decisively than any marketing campaign. Addressing life goals of users makes the difference (assuming that other goals are also met) between a satisfied user and a fanatically loyal user.

User goals are user motivations

In summary, it's important to remember that understanding personas is more about understanding motivations and goals than it is about understanding specific tasks or demographics. Linking up persona goals with Norman's model, top-level user motivations include:

- Experience goals, which are related to visceral processing: how a user wants to *feel*
- End goals, which are related to behavior: what a user wants to *do*
- Life goals, which are related to reflection: who a user wants to *be*

Using personas, goals, and scenarios (as you'll learn in upcoming chapters) provides the key to unlocking the power of visceral, behavioral, and reflective design, and bringing these together into a harmonious whole. While some of our best designers seem to understand and act upon these aspects of design almost intuitively, consciously designing for all levels of human cognition and emotion offers tremendous potential for creating more satisfying and delightful user experiences.

Types of goals

User goals are not the only type of goals that designers need to take into account. Customer goals, business goals, and technical goals are all nonuser goals. Typically, these goals must be acknowledged and considered, but they do not form the basis for the design direction. Although these goals do need to be addressed, they must not be addressed at the expense of the user.

Customer goals

Customers, as already discussed, have different goals than users. The exact nature of these goals varies quite a bit between consumer and enterprise products. Consumer customers are often parents, relatives, or friends who often have concerns about the safety and happiness of the persons for whom they are purchasing the product. Enterprise customers are typically IT managers, and they often have concerns about security, ease of maintenance, and ease of customization. Customer personas also may have their own life, experience, and especially end goals in relation to the product if they use it in any capacity. Customer goals should never trump end goals but need to be considered within the overall design.

Business and organizational goals

Businesses and other organizations have their own requirements for products, services, and systems, which should also be modeled and considered when devising design solutions. While the goals of businesses, where users and customers work, are typically captured in user and customer personas, it is often useful to define the business goals of the organization commissioning the design and developing and selling (or otherwise distributing) the product. Clearly, these organizations are hoping to accomplish something with the product (which is why they are willing to spend money and effort on design and development),

Business goals include the following:

- Increase profit
- Increase market share
- Retain customers
- Defeat the competition
- Use resources more efficiently
- Offer more products or services

You may find yourself designing on behalf of an organization that is not necessarily a business, such as a museum, nonprofit, or school (though all organizations are increasingly run as businesses these days). These organizations also have goals that must be considered, such as:

- Educate the public
- Raise enough money to cover overhead

Technical goals

Most of the software-based products we use everyday are created with technical goals in mind. Many of these goals ease the task of software creation, which is a programmer's goal. This is why they typically take precedence at the expense of the users' goals. Technical goals include:

- Run in a variety of browsers
- Safeguard data integrity
- Increase program execution efficiency
- Use a particular development language or library
- Maintain consistency across platforms

Technical goals in particular are very important to the development staff. It is important to stress early in the education process that these goals must ultimately serve user and business goals. Technical goals are not terribly meaningful to the success of a product unless they are derived from the need to meet other more human-oriented goals. It might be a software company's *task* to use new technology, but it is rarely a *user's goal* for them to do so. In most cases, users don't care if their job is accomplished with hierarchical databases, relational databases, object-oriented databases, flat-file systems, or black magic. What we care about is getting our job done swiftly, effectively, and with a modicum of ease and dignity.

Successful products meet user goals first

"Good design" has meaning only for a person using a product for some purpose. You cannot have purposes without people. The two are inseparable. This is why personas are such an important tool in the process of designing behavior; they represent specific people with specific purposes or goals.

The most important purposes or goals to consider when designing a product are those of the individuals who actually use it, not necessarily those of its purchaser. A real person, not a corporation or even an IT manager, interacts with your product, so you must regard her personal goals as more significant than those of the corporation who employs her or the IT manager who supports her. Your users will do their best to achieve their employer's business goals, while at the same time looking after their own personal goals. A user's most important goal is always to retain her human dignity: not to feel stupid.

We can reliably say that we make the user feel stupid if we let her make big mistakes, keep her from getting an adequate amount of work done, or bore her.

Don't make the user feel stupid.

This is probably the most important interaction design guideline. In the course of this book, we examine numerous ways in which existing software makes the user feel stupid, and we explore ways to avoid that trap.

The essence of good interaction design is devising interactions that achieve the goals of the manufacturer or service provider and their partners without violating the goals of users.

Constructing Personas

As previously discussed, personas are derived from patterns observed during interviews with and observations of users and potential users (and sometimes customers) of a product. Gaps in this data are filled by supplemental research and data provided by SMEs, stakeholders, and available literature. Our goal in constructing a set of personas is to represent the diversity of observed motivations, behaviors, attitudes, aptitudes, mental models, work or activity flows, environments, and frustrations with current products or systems.

Creating believable and useful personas requires an equal measure of detailed analysis and creative synthesis. A standardized process aids both of these activities significantly. The process described in this section, developed by Robert Reimann, Kim Goodwin, and Lane Halley at Cooper, is the result of an evolution in practice over the span of hundreds of interaction design projects, and has been documented in several papers.[11] There are a number of effective methods for identifying behavior patterns in research and turning these into useful user archetypes, but we've found the transparency and rigor of this process to be an ideal way for designers new to personas to learn how to properly construct personas, and for experienced designers to stay focused on actual behavior patterns, especially in consumer domains. The principle steps are:

1. Identify behavioral variables.
2. Map interview subjects to behavioral variables.
3. Identify significant behavior patterns.
4. Synthesize characteristics and relevant goals.
5. Check for redundancy and completeness.

6. Expand description of attributes and behaviors.
7. Designate persona types.

We discuss each of these steps in detail in the following sections.

Step 1: Identify behavioral variables

After you have completed your research and performed a cursory organization of the data, list the distinct aspects of observed behavior as a set of **behavioral variables**. Demographic variables such as age or geographic location may also seem to affect behavior, but be wary of focusing on demographics because behavioral variables will be far more useful in developing effective user archetypes.

Generally, we see the most important distinction between behavior patterns emerge by focusing on the following types of variables:

- Activities — What the user does; frequency and volume
- Attitudes — How the user thinks about the product domain and technology
- Aptitudes — What education and training the user has; capability to learn
- Motivations — Why the user is engaged in the product domain
- Skills — User capabilities related to the product domain and technology

For enterprise applications, behavioral variables are often closely associated with job roles, and we suggest listing out the variables for each role separately. Although the number of variables will differ from project to project, it is typical to find 15 to 30 variables per role.

These variables may be very similar to those you identified as part of your persona hypothesis. Compare behaviors identified in the data to the assumptions made in the persona hypothesis. Were the possible roles that you identified truly distinct? Were the behavioral variables (see Chapter 4) you identified valid? Were there additional, unanticipated ones, or ones you anticipated that weren't supported by data?

List the complete set of behavioral variables observed. If your data is at variance with your assumptions, you need to add, subtract, or modify the roles and behaviors you anticipated. If the variance is significant enough, you may consider additional interviews to cover any gaps in the new behavioral ranges that you've discovered.

Step 2: Map interview subjects to behavioral variables

After you are satisfied that you have identified the set of significant behavioral variables exhibited by your interview subjects, the next step is to map each interviewee against each variable. Some of these variables will represent a continuous range of behavior (for instance, from a computer novice to a computer expert), and a few will represent multiple discrete choices (for example, uses a digital camera versus uses a film camera).

Mapping the interviewee to a precise point in the range isn't as critical as identifying the placement of interviewees in relationship to each other. In other words, it doesn't matter if an interviewee falls at precisely 45% or 50% on the scale. There's often no good way to measure this precisely; you must rely on your gut feeling based on your observations of the subject. The desired outcome of this step is to accurately represent the way multiple subjects cluster with respect to each significant variable (see Figure 5-4).

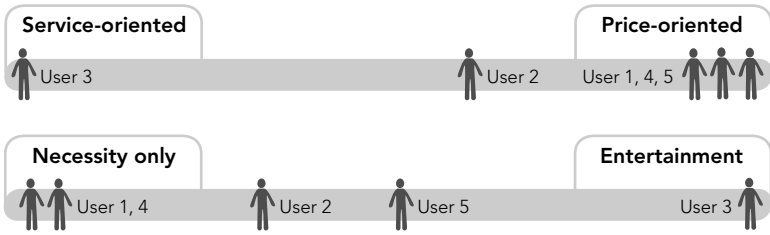

Figure 5-4 Mapping interview subjects to behavioral variables. This example is from an online store. Interview subjects are mapped across each behavioral axis. Precision of the absolute position of an individual subject on an axis is less important than its relative position to other subjects. Clusters of subjects across multiple axes indicate significant behavior patterns.

Step 3: Identify significant behavior patterns

After you have mapped your interview subjects, look for clusters of subjects that occur across multiple ranges or variables. A set of subjects who cluster in six to eight different variables will likely represent a significant **behavior pattern** that will form the basis of a persona. Some specialized roles may exhibit only one significant pattern, but typically you will find two or even three such patterns.

For a pattern to be valid there must be a logical or causative connection between the clustered behaviors, not just a spurious correlation. For example, there is clearly a logical connection if data shows that people who regularly purchase CDs also like to download MP3 files, but there is probably no logical connection if the data shows that interviewees who frequently purchase CDs online are also vegetarians.

Step 4: Synthesize characteristics and relevant goals

For each significant behavior pattern you identify, you must synthesize details from your data. Describe the potential use environment, typical workday (or other relevant context), current solutions and frustrations, and relevant relationships with others.

At this point, brief bullet points describing characteristics of the behavior are sufficient. Stick to observed behaviors as much as possible. A description or two that sharpens the personalities of your personas can help bring them to life. However, too much fictional, idiosyncratic biography is a distraction and makes your personas less credible. Remember that you are creating a design tool, not a character sketch for a novel. Only concrete data can support the design and business decisions your team will ultimately make.

One fictional detail at this stage *is* important: the personas' first and last names. The name should be evocative of the type of person the persona is, without tending toward caricature or stereotype. We use a baby name book as a reference tool in creating persona names. You can also, at this time, add in some demographic information such as age, geographic location, relative income (if appropriate), and job title. This information is primarily to help you visualize the persona better as you assemble the behavioral details. From this point on, you should refer to the persona by his or her name.

Synthesizing goals

Goals are the most critical detail to synthesize from your interviews and observations of behaviors. Goals are best derived from an analysis of the behavior patterns comprising each persona. By identifying the logical connections between each persona's behaviors, you can begin to infer the goals that lead to those behaviors. You can infer goals both by observing actions (what interview subjects in each persona cluster are trying to accomplish and why) and by analyzing subject responses to goal-oriented interview questions (see Chapter 4).

To be effective as design tools, goals must always directly relate, in some way, to the product being designed. Typically, the majority of useful goals for a persona are *end goals*. You can expect most personas to have three to five end goals associated with

them. Life goals are most useful for personas of consumer-oriented products, but they can also make sense for enterprise personas in transient job roles. Zero or one life goal is appropriate for most personas. General experience goals such as "don't feel stupid" and "don't waste time" can be taken as implicit for almost any persona. Occasionally, a specific domain may dictate the need for more specific experience goals; zero to two experience goals is appropriate for most personas.

Persona relationships

It sometimes makes sense for the set of personas for a product to be part of the same family or corporation and to have interpersonal or social relationships with each other. The typical case, however, is for individual personas to be completely unrelated to each other and often from completely different geographic locations and social groups.

When considering whether it makes sense for personas to have business or social relationships, think about:

1. Whether you observed any behavioral variations in your interview subjects related to variations in company size, industry, or family/social dynamic. (In this case, you'll want to make sure that your persona set represents this diversity by being situated in at least a couple of different businesses or social settings.)
2. If it is critical to illustrate workflow or social interactions between coworkers or members of a family or social group.

If you create personas that work for the same company or have social relationships with each other, you might run into difficulties if you need to express a significant goal that doesn't belong with the preestablished relationship. While a single social relationship between your set of personas is easier to define than several different, unrelated social relationships between individual personas and minor players outside the persona set, it can be much better to put the initial effort into development of diverse personas than to risk the temptation of bending more diverse scenarios to fit a single social dynamic.

Step 5: Check for completeness and redundancy

At this point, your personas should be starting to come to life. You should check your mappings and personas' characteristics and goals to see if there are any important gaps that need filling. This again may point to the need to perform additional research directed at finding particular behaviors missing from your behavioral axes. You might also want to check your notes to see if there are any political personas that you need to add to satisfy stakeholder assumptions or requests.

If you find that two personas seem to vary only by demographics, you may choose to eliminate one of the redundant personas or tweak the characteristics of your personas to make them more distinct. Each persona must vary from all others in at least one significant behavior. If you've done a good job of mapping, this shouldn't be an issue.

By making sure that your persona set is complete and that each persona is meaningfully distinct, you ensure that your personas sufficiently represent the diversity of behaviors and needs in the real world, and that you have as compact a design target as possible, which reduces work when you begin designing interactions.

Step 6: Expand description of attributes and behaviors

Your list of bullet point characteristics and goals arrived at in Step 4 points to the essence of complex behaviors, but leaves much implied. Third-person narrative is far more powerful at conveying the persona's attitudes, needs, and problems to other team members. It also deepens the designer/authors' connection to the personas and their motivations.

A typical persona description should be a synthesis of the most important details observed during research, relevant to this persona. This becomes a very effective communication tool. Ideally, the majority of your user research findings should be contained in your persona description. This will be the manner in which your research directly informs design activities (as you will see in the upcoming chapters).

This narrative should be no longer than one or two pages of prose. The persona narrative does not need to contain every observed detail because, ideally, the designers also performed the research, and most people outside the design team do not require more detail than this.

The narrative must, by nature, contain some fictional situations, but as previously discussed, it is not a short story. The best narrative quickly introduces the persona in terms of his job or lifestyle, and briefly sketches a day in his life, including peeves, concerns, and interests that have direct bearing on the product. Details should be an expansion of your list of characteristics, with additional data derived from your observations and interviews. The narrative should express what the persona is looking for in the product by way of a conclusion.

Be careful about the precision of detail in your descriptions. The detail should not exceed the depth of your research. In scientific disciplines, if you record a measurement of 35.421 meters, this implies that your measurements are accurate to .001 meters. A detailed persona description implies a similar level of observation in your research.

When you start developing your narrative, choose photographs of your personas. Photographs make them feel more real as you create the narrative and engage others on the team when you are finished. You should take great care in choosing a photograph. The best photos capture demographic information, hint at the environment (a persona for a nurse should be wearing a nurse's uniform and be in a clinical setting, perhaps with a patient), and capture the persona's general attitude (a photo for a clerk overwhelmed by paperwork might look harried). The authors keep several searchable databanks of stock photography available for finding the right persona pictures.

We have also found it useful to create photographic collages for each persona to convey more emotional and experiential forces that drive the persona (see Figure 5-5). Numerous small images juxtaposed have the potential to convey things that are difficult to describe in words. There are also times that we find it useful to create models of the personas' environments (for example, in the form of a floorplan). Again, this helps to make these environmental considerations more tangible.

When creating such communication aides, it's important to remember that personas are design and decision-making tools, not an end in themselves. While there can be a lot of power in creating a holistic image of a persona, too much embellishment and theatre can run the risk of making personas seem a fluffy waste of time. This can ultimately reduce their usefulness as user models.

Figure 5-5 Collages such as this, combined with carefully written narratives, are an effective way to convey the emotional and experiential aspects of a persona.

Step 7: Designate persona types

By now, your personas should feel very much like a set of real people whom you know. The final step in persona construction finishes the process of turning your qualitative research into a powerful set of design tools.

Design requires a target — the audience upon whom the design is focused. Typically, the more specific the target, the better. Trying to create a design solution that simultaneously serves the needs of even three or four personas can be quite an overwhelming task.

What we then must do is *prioritize* our personas to determine which should be the primary design target. The goal is to find a single persona from the set whose needs and goals can be completely and happily satisfied by a single interface without disenfranchising any of the other personas. We accomplish this through a process of designating **persona types**. There are six types of persona, and they are typically designated in roughly the order listed here:

- Primary
- Secondary
- Supplemental
- Customer
- Served
- Negative

We discuss each of these persona types and their significance from a design perspective in the following sections.

Primary personas

Primary personas represent the primary target for the design of an interface. There can be only one primary persona per *interface* for a product, but it is possible for some products (especially enterprise products) to have multiple distinct interfaces, each targeted at a distinct primary persona. For example, a health-care information system might have separate clinical and financial interfaces, each targeted at a different persona. It should be noted that we use the term *interface* in an abstract sense here. In some cases, two separate interfaces might be two separate applications that act on the same data; in other cases, the two interfaces might simply be two different sets of functionality served to two different users based upon their role or customization.

A primary persona will not be satisfied by a design targeted at any other persona in the set. However, if the primary persona is the target, all other personas will not, at least, be dissatisfied. (As you'll see below, we will then figure out how to satisfy these other personas without disturbing the primary.)

Focus the design for each interface on a single primary persona.

Choosing the primary persona is a process of elimination: Each persona must be tested by comparing the goals of that persona against goals of the others. If no clear primary persona is evident, it could mean one of two things: Either the product needs multiple interfaces, each with a suitable primary persona (often the case for enterprise and technical products), or the product is trying to accomplish too much. If a consumer product has multiple primary personas, the scope of the product may be too broad.

Secondary personas

A **secondary persona** is mostly satisfied with the primary persona's interface but has specific additional needs that can be accommodated without upsetting the product's ability to serve the primary persona. We do not always have a secondary persona, and more than three or four secondary personas can be a sign that the proposed product's scope may be too large and unfocused. As you work through solutions, your approach should be to first design for the primary, and then adjust the design to accommodate the secondary.

Supplemental personas

User personas that are not primary or secondary are **supplemental personas**. Their needs are completely represented by a combination of primary and secondary personas and are completely satisfied by the solution we devise for one of our primaries. There can be any number of supplemental personas associated with an interface. Often political personas — the ones added to the cast to address stakeholder assumptions — become supplemental personas.

Customer personas

Customer personas address the needs of customers, not end users, as discussed earlier in this chapter. Typically, customer personas are treated like secondary personas. However, in some enterprise environments, some customer personas may be primary personas for their own administrative interface.

Served personas

Served personas are somewhat different from the persona types already discussed. They are not users of the product at all; however, they are *directly affected by the use of the product*. A patient being treated by a radiation therapy machine is not a user of the machine's interface, but she is very much *served* by a good interface. Served personas provide a way to track second-order social and physical ramifications of products. These are treated like secondary personas.

Negative personas

Negative personas are used to communicate to stakeholders and product team members that there are specific types of users that the product is *not* being built to serve. Like served personas, they aren't users of the product. Their use is purely rhetorical: to help communicate to other members of the team that a persona should definitely *not* be the design target for the product. Good candidates for negative personas are often technology-savvy early adopter personas for consumer products and IT specialists for business-user enterprise products.

Other Models

Personas are extremely useful tools, but they are certainly not the only tool to help model users and their environment. Holtzblatt and Beyer's *Contextual Design* provides a wealth of information on the models briefly discussed here.

Workflow models

Workflow or **sequence models** are useful for capturing information flow and decision-making processes inside organizations and are usually expressed as flow charts or directed graphs that capture several phenomena:

- The goal or desired outcome of a process
- The frequency and importance of the process and each action
- What initiates or prompts the execution of the process and each action
- Dependencies — what must be in place to perform the process and each action, as well as what is dependent on the completion of the process and each action
- People who are involved and their roles and responsibilities
- Specific actions that are performed
- Decisions that are made

- Information that is used to support decisions
- What goes wrong — errors and exception cases
- How errors and exceptions are corrected

A well-developed persona should capture individual workflows, but workflow models are still necessary for capturing interpersonal and organizational workflows. Interaction design based primarily on workflow often fails in the same way as "implementation model" software whose interaction is based primarily on its internal technical structure. Because workflow is to business what structure is to programming, workflow-based design typically yields a kind of "business implementation model" that captures all of the functionality but little of the humanity.

Artifact models

Artifact models represent, as the name suggests, different artifacts that users employ in their tasks and workflows. Often these artifacts are online or paper forms. Artifact models typically capture commonalities and significant differences between similar artifacts for the purpose of extracting and replicating best practices in the eventual design. Artifact models can be useful later in the design process, with the caveat that direct translation of paper systems to digital systems, without a careful analysis of goals and application of design principles (especially those found in Part II of this book), usually leads to usability issues.

Physical models

Physical models, like artifact models, endeavor to capture elements of the user's environment. Physical models focus on capturing the layout of physical objects that comprise the user's workspace, which can provide insight into frequency of use issues and physical barriers to productivity. Good persona descriptions will incorporate some of this information, but it may be helpful in complex physical environments (such as hospital floors and assembly lines) to create discrete, detailed physical models (maps or floorplans) of the user environment.

Personas and other models make sense out of otherwise overwhelming and confusing user data. Now that you are empowered with sophisticated models as design tools, the next chapter will show you how to employ these tools to translate user goals and needs into workable design solutions.

Notes

1. Cooper, 1999
2. Constantine and Lockwood, 2002
3. Grudin and Pruitt, 2002
4. Mikkelson, N., and Lee, W. O., 2000
5. Grudin and Pruitt, 2002
6. Grudin and Pruitt, 2002
7. Constantine and Lockwood, 1999
8. Beyer and Holtzblatt, 1998
9. Dillon, 2001
10. Goodwin, 2001
11. Goodwin, 2002, 2002a

6

The Foundations of Design: Scenarios and Requirements

In the two previous chapters, we talked about how to gather qualitative information about users and create models using that information. Through careful analysis of user research and synthesis of personas and other user models, we create a clear picture of our users and their respective goals. This brings us, then, to the crux of the whole method: how we use this understanding of people to create design solutions that satisfy and inspire users, while simultaneously addressing business goals and technical constraints.

This chapter describes the first part of a process for bridging the research-design gap. It employs personas as the main characters in a set of techniques that rapidly arrive at design solutions in an iterative, repeatable, and testable fashion. This process has four major activities: developing stories or *scenarios* as a means of imagining ideal user interactions, using those scenarios to define *requirements*, using these requirements in turn to define the fundamental *interaction framework* for the product, and filling in the framework with ever-increasing amounts of design detail. The glue that holds the processes together is *narrative*: using personas to create stories that point to design.

Scenarios: Narrative as a Design Tool

Narrative, or storytelling, is one of the oldest human activities. Much has been written about the power of narrative to *communicate* ideas. However, narrative is also one of our most powerful creative methods. From a very young age, we are accustomed to using stories to think about possibilities, and this is an incredibly effective way to *imagine* a new and better future for our users. Imagining a story about a person using our product leverages our creativity to a greater power than when we just imagine a better form factor or configuration of screen elements. Further, because of the intrinsically social aspect of narrative, it is a very effective and compelling way to share good ideas among team members and stakeholders. Ultimately, experiences designed around narrative tend to be more comprehensible and engaging for users because they are structured around a story.

Evidence of the effectiveness of narrative as a design tool is all around us. The famous Disney Imagineers would be lost without the modern-day myths they use as the foundation for the experiences they build. Much has been written about this idea: Brenda Laurel explored the concept of structuring interaction around dramatic principles in her 1991 book *Computers as Theater*, where she urges us to ". . . focus on designing the action. The design of objects, environments, and characters is all subsidiary to this central goal."[1] John Rheinfrank and Shelley Evenson also talk about the power of "stories of the future" for developing conceptually complex interactive systems,[2] and John Carroll has created a substantial body of work about scenario-based design, which we discuss later in this chapter.

Narrative also lends itself to effective visual depictions of interactive products. Because interaction design is first and foremost the design of behavior that occurs over time, a narrative structure, combined with the support of fast and flexible visualization tools (such as the humble whiteboard), is perfectly suited for motivating, envisioning, representing, and validating interaction concepts.

Interaction design narratives are quite similar to the comic-book-like sequences called storyboards that are used in the motion picture industry. They share two significant characteristics: plot and brevity. Just as storyboards breathe life into a movie script, design solutions should be created and rendered to follow a plot — a story. Putting too much detail into the storyboards simply wastes time and money and has a tendency to tie us to suboptimal ideas simply because drawing them consumes significant resources.

In the initial requirements definition phase we are free to focus only on the "plot points," allowing us to be fluid as we explore design concepts. Because they are enough to convey the action and the potential experience, many millions of Hollywood dollars

have been invested on the basis of simple pencil sketches or line drawings. By focusing on the narrative, we are able to quickly and flexibly arrive at a high-level design solution without getting bogged-down by the inertia and expense inherent to high-production-value renderings (though such renderings are certainly appropriate once a working design framework is in place).

Scenarios in design

In the 1990s, substantial work was done by the HCI (Human-Computer Interaction) community around the idea of use-oriented software design. From this work came the concept of the **scenario**, commonly used to describe a method of *design problem solving by concretization*: making use of a specific story to both construct and illustrate design solutions. These concepts are discussed by John Carroll, in his book, *Making Use*:

> *Scenarios are paradoxically concrete but rough, tangible but flexible . . . they implicitly encourage "what-if?" thinking among all parties. They permit the articulation of design possibilities without undermining innovation . . . Scenarios compel attention to the use that will be made of the design product. They can describe situations at many levels of detail, for many different purposes, helping to coordinate various aspects of the design project.*[3]

Carroll's use of **scenario-based design** focuses on describing how *users accomplish tasks*. It consists of an environmental *setting* and includes *agents* or *actors* that are abstracted stand-ins for users, with role-based names such as Accountant or Programmer.

Although Carroll certainly understands the power and importance of scenarios in the design process, we've found two shortcomings with scenarios as Carroll approaches them:

- Carroll's concept of the actor as an abstracted, role-oriented model is not sufficiently concrete to provide understanding of or empathy with users. It is impossible to design appropriate behaviors for a system without understanding the users of the system in specific detail.
- Carroll's scenarios jump too quickly to the elaboration of tasks without considering the user's goals and motivations that drive and filter these tasks. Although Carroll does briefly discuss goals, he refers only to *goals of the scenario*. These goals are circularly defined as the completion of specific tasks. In our experience, user goals must be considered before user tasks can be identified and prioritized. Without addressing the motivation of human behavior, high-level product definition can be difficult and misguided.

The missing ingredient in Carroll's scenario-based design methods is the use of personas. A persona provides a tangible representation of the user to act as a believable agent in the setting of a scenario. In addition to reflecting current behavior patterns and motivations, personas enable the exploration of how user motivations should inflect and prioritize tasks in the future. Because personas model *goals* and not simply tasks, the scope of the problems addressed by scenarios can be broadened to include those related to product definition. They help answer the questions, "What should this product *do*?" and "How should this product look and behave?"

Using personas in scenarios

Persona-based scenarios are concise narrative descriptions of one or more personas using a product to achieve specific goals. They allow us to start our designs from a story describing an ideal experience from the persona's perspective, focusing on people, and how they think and behave, rather than on technology or business goals.

Scenarios can capture the *nonverbal dialogue*[4] between the user and a product, environment, or system over time, as well as the structure and behavior of interactive functions. Goals serve as a filter for tasks and as guides for structuring the display of information and controls during the iterative process of constructing the scenarios.

Scenario content and context are derived from information gathered during the Research phase and analyzed during the Modeling phase. Designers role-play personas as the characters in these scenarios,[5] similar to actors performing improvisation. This process leads to real-time synthesis of structure and behavior — typically, at a whiteboard — and later informs the detailed look-and-feel. Finally, personas and scenarios are used to test the validity of design ideas and assumptions throughout the process.

Different types of scenarios

The Goal-Directed Design method employs three types of persona-based scenarios at different points in the process, each with a successively more interface-specific focus. The first — the **context scenario** — is used to explore, at a high level, how the product can best serve the needs of the personas. (We used to call these "day-in-the-life scenarios," but found that term excessively broad.) The context scenarios are created before any design is performed and are written from the perspective of the persona, focused on human activities, perceptions, and desires. It is in the development of this kind of scenario that the designer has the most leverage to imagine an

ideal user experience. More detail about the creation of this type of scenario can be found later in this chapter, under Step 4 in the Requirements Definition process.

Once the design team has defined the product's functional and data elements, and developed a Design Framework (as described in Chapter 7), a context scenario is revised to become a **key path scenario** by more specifically describing user interactions with the product and by introducing the vocabulary of the design. These scenarios focus on the most significant user interactions, always maintaining attention on how a persona uses the product to achieve their goals. Key path scenarios are iteratively refined along with the design as more and more detail is developed.

Throughout this process, the design team uses **validation scenarios** to test the design solution in a variety of situations. These scenarios tend to be less detailed and typically take the form of a number of "what if . . ." questions about the proposed solutions. More detail about development and use of key path and validation scenarios can be found in Chapter 7.

Persona-based scenarios versus use cases

Scenarios and use cases are both methods of describing a user's interaction with a system. However, they serve very different functions. Goal-Directed scenarios are an iterative means of defining the *behavior* of a product from the standpoint of specific users (personas). This includes not only the functionality of the system, but the priority of functions and the way those functions are expressed in terms of what the user sees and how she interacts with the system.

Use cases, on the other hand, are a technique based on exhaustive descriptions of functional requirements of the system, often of a transactional nature, focusing on low-level user action and accompanying system response.[6] The precise *behavior* of the system — precisely *how* the system responds — is not typically part of a conventional or *concrete* use case; many assumptions about the form and behavior of the system to be designed remain implicit.[7] Use cases permit a complete cataloguing of user tasks for different classes of users but say little or nothing about how these tasks are presented to the user or how they should be prioritized in the interface. In our experience, the biggest shortcoming of traditional use cases as a basis for interaction design is their tendency to treat all possible user interactions as equally likely and important. This is indicative of their origin in software engineering rather than interaction design. They may be useful in identifying edge cases and for determining that a product is functionally complete, but they should be deployed only in the later stages of design validation.

Requirements: The "What" of Interaction Design

The Requirements Definition phase determines the *what* of the design: what information and capabilities our personas require to accomplish their goals. It is absolutely critical to define and agree upon the *what* before we move on to the next question: *how* the product looks, behaves, operates, and feels. Conflating these two questions can be one of the biggest pitfalls in the design of an interactive product. Many designers are tempted to jump right into active design and render possible solutions. Regardless of how creative and skillful you are, we urge you not to do this. It runs the risk of turning into a never-ending circle of iteration; proposing a solution without clearly defining and agreeing upon the problem leaves you without a clear method of evaluating the fitness of the design. In lieu of such a method, you, your stakeholders, and your clients are likely to resort to taste and gut instinct, which have a notoriously low success ratio with something as complex as an interactive product.

Define *what* the product will do before you design *how* the product will do it.

It's important to note that our concept of a "requirement" here is much different from the way the term is commonly misused in the industry. In many product-development organizations, "requirement" has come to be synonymous with "feature" or "function." While there is clearly a relationship between requirements and functions (which we leverage as a key part of our design process, as you will see in the next chapter), we suggest that you think of requirements as synonymous with *needs*. Put another way, at this point, you want to rigorously define the human and business needs that your product must satisfy.

Another critical reason not to conflate requirements with features is that in figuring out the best way to meet a particular human need, an interaction designer has an extraordinary amount of leverage to create a powerful and compelling product. Think, for example, about designing a data analytics tool to help an executive better understand the state of his business. If you jump right to the *how* without understanding the *what*, you might assume that the output of the tool should be reports. It would be easy to come to this conclusion; if you went out and performed user research, you probably would have noticed that reports are a very widespread

and accepted solution. However, if you imagine some scenarios and analyze your users' actual requirements, you might realize that your executive actually needs a way to recognize exceptional situations before opportunities are missed or problems arise, as well a way to understand emerging trends in the data. From here, it isn't difficult to see that static, flat reports are hardly the best way to meet these needs. With such a solution, your executive has to do the hard work of scrutinizing several of these reports to find the significant data underlying such exceptions and trends. Much better solutions might include data-driven exception reporting or real-time trend monitors.

A final reason to separate problem and solution is that such an approach provides the maximum flexibility in the changing face of technological constraints and opportunities. By clearly defining the user need, designers can then work with technologists to find the best solutions, without compromising the product's ability to help people achieve their goals. Working in this manner, the product definition is not at risk when the implementation runs into problems, and it becomes possible to plan long-term technology development so that it can provide increasingly sophisticated ways of meeting user needs.

As we've mentioned briefly, these requirements come from several sources. Personas' previous experiences and mental models often result in some baseline expectations of the product. We derive the bulk of the user requirements from analyzing ideal usage scenarios, and understand business and technical requirements from our stakeholder interviews. Our Goal-Directed process for defining product requirements is described below.

Requirements Definition Using Personas and Scenarios

As discussed briefly in Chapter 1, the translation from robust models to design solutions really consists of two major phases: **Requirements Definition** answers the broad questions about what a product is and what it should do, and **Framework Definition** answers questions about how a product behaves and how it is structured to meet user goals. In this section, we'll discuss Requirements Definition in detail, followed by a discussion of the Framework Definition in Chapter 7. The methods described are based upon the persona-based scenario methodology developed at Cooper by Robert Reimann, Kim Goodwin, Dave Cronin, Wayne Greenwood, and Lane Halley.

The Requirements Definition process comprises the following five steps (which are described in detail in the remainder of this chapter):

1. Creating problem and vision statements
2. Brainstorming
3. Identifying persona expectations
4. Constructing context scenarios
5. Identifying requirements

Although these steps proceed in roughly chronological order, they represent an iterative process. Designers can expect to cycle through Steps 3 through 5 several times until the requirements are stable. This is a necessary part of the process and shouldn't be short-circuited. A detailed description of each of these steps follows.

Step 1: Creating problem and vision statements

Before beginning the process of ideation, it's important for designers to have a clear mandate for moving forward. While the Goal-Directed method aims to comprehensively define the product through personas, scenarios, and requirements, it is often useful at this point to define what direction these scenarios and requirements should be headed in. At this point in the process, we already have a sense of which users we're targeting and what their goals are, but lacking a clear product mandate, there is still room for considerable confusion. Problem and vision statements provide just such a mandate and are extremely helpful in building consensus among stakeholders before the design process moves forward.

At a high level, the **problem statement** defines the purpose of the design initiative.[8] A design problem statement should concisely reflect a situation that needs changing, for both the personas *and* for the business providing the product to the personas. Often a cause-and-effect relationship exists between business concerns and persona concerns. For example:

> *Company X's customer satisfaction ratings are low and market share has diminished by 10% over the past year because users don't have adequate tools to perform X, Y, and Z tasks that would help them meet their goal of G.*

The connection of business issues to usability issues is critical to drive stakeholders' buy-in to design efforts and to frame the design effort in terms of both user and business goals.

The **vision statement** is an inversion of the problem statement that serves as a high-level design objective or mandate. In the vision statement, you lead with the user's needs, and you transition from those to how business goals are met by the design vision:

> *The new design of Product X will help users achieve G by giving them the ability to perform X, Y, and Z with greater [accuracy, efficiency, and so on], and without problems A, B, C that they currently experience. This will dramatically improve Company X's customer satisfaction ratings and lead to increased market share.*

The content of both the problem and vision statements should come directly from research and user models. User goals and needs should derive from the primary and secondary personas, and business goals should be extracted from stakeholder interviews.

Problem and vision statements are useful both when you are redesigning an existing product and for new technology products or products being designed for unexplored market niches, when formulating user goals and frustrations into problem and vision statements helps to establish team consensus and attention on the priorities of design activity to follow.

Step 2: Brainstorming

At the early stage of Requirements Definition, brainstorming assumes a somewhat ironic purpose. At this point in the project, we have been researching and modeling users and the domain for days or even months, and it is almost impossible to avoid having developed some preconceptions about what the solution looks like. However, we'd ideally like to create context scenarios without these prejudgments, and instead really focus on how our personas would likely want to engage with the product. The reason we brainstorm at this point in the process is to get these ideas out of our heads so that we can record them and thereby "let them go" for the time being.

The primary purpose here is to eliminate as much preconception as possible, allowing designers to be open-minded and flexible as they use their imagination to construct scenarios, and use their analytic minds to derive requirements from these scenarios. A side benefit of brainstorming at this point in the process is to switch your brain into "solution mode." Much of the work performed in the Research and Modeling phases is analytical in nature, and it takes a different mindset to come up with inventive designs.

Brainstorming should be unconstrained and uncritical — put all the wacky ideas you've been considering (plus some you haven't) out on the table and then be prepared to record them and file them away for safekeeping until much later in the process. It's not necessarily likely any of them will be useful in the end, but there might be the germ of something wonderful that will fit into the design framework you later create. Karen Holtzblatt and Hugh Beyer describe a facilitated method for brainstorming that can be useful for getting a brainstorming session started, especially if your team includes nondesigners.[9]

Don't spend too much time on the brainstorming step; a few hours should be more than sufficient for you and your teammates to get all those crazy ideas out of your systems. If you find your ideas getting repetitious, or the popcorn stops popping, that's a good time to stop.

Step 3: Identifying persona expectations

As we discussed in Chapter 2, a person's **mental model** is their own internal representation of reality — the way they think about or explain something to themselves. Mental models are deeply ingrained and are often the result of a lifetime of experience. People's expectations about a product and the way it works are highly informed by their mental model.

Returning to our discussion in Chapter 2, it's absolutely critical that the **represented model** of the interface — how the design behaves and presents itself — should match the user's mental model as closely as possible, rather than reflecting the implementation model of how the product is actually constructed internally.

In order to accomplish this, we must formally record these expectations. They will be an important source of requirements. For each primary and secondary persona, you must identify:

- Attitudes, experiences, aspirations, and other social, cultural, environmental, and cognitive factors that influence the persona's expectations
- General expectations and desires the persona may have about the experience of using the product
- Behaviors the persona will expect or desire from the product
- How that persona thinks about basic elements or units of data (for example, in an e-mail application, the basic elements of data might be messages and people)

Your persona descriptions may contain enough information to answer these questions directly; however, your research data will remain a rich resource. Use it to

analyze how interview subjects define and describe objects and actions that are part of their usage patterns, along with the language and grammar they use. Some things to look for include:

- What do the subjects mention first?
- Which action words (verbs) do they use?
- Which intermediate steps, tasks, or objects in a process *don't* they mention? (Hint: These might not be terribly important to the way they think about things.)

Step 4: Constructing context scenarios

While all scenarios are stories about people and their activities, context scenarios are the most storylike of the three types we employ. The focus is on the persona's activities, as well as her motivations and mental model. **Context scenarios** describe the broad context in which usage patterns are exhibited and include environmental and organizational (in the case of enterprise systems) considerations. [10]

As we discussed above, *this is where design begins.* As you develop context scenarios, you should be focusing on how the product you are designing can best help your personas achieve their goals. Context scenarios establish the primary touch points that each primary and secondary persona has with the system (and possibly with other personas) over the course of a day or some other meaningful length of time.

Context scenarios should be broad and relatively shallow in scope. They should not describe product or interaction detail but rather should focus on high-level actions from the user's perspective. It is important to map out the big picture first so that we can systematically identify user requirements. Only then will we be able to design appropriate interactions and interfaces.

Context scenarios address questions such as the following:

- In what setting(s) will the product be used?
- Will it be used for extended amounts of time?
- Is the persona frequently interrupted?
- Are there multiple users on a single workstation or device?
- With what other products will it be used?
- What primary activities does the persona need to perform to meet her goals?
- What is the expected end result of using the product?
- How much complexity is permissible, based on persona skill and frequency of use?

Context scenarios should *not* represent system behaviors as they currently are. These scenarios represent the brave new world of Goal-Directed products, so, especially in the initial phases, focus on the goals. Don't yet worry about exactly *how* things will get accomplished — you should initially treat the design as a bit of a magic black box.

In most cases, more than one context scenario is necessary. This is true especially when there are multiple primary personas, but sometimes even a single primary persona may have two or more distinct contexts of use.

Context scenarios are also entirely *textual*. We are not yet discussing form, only the behaviors of the user and the system. This discussion is best accomplished as a textual narrative.

An example context scenario

The following is an example of a first iteration of a context scenario for a primary persona for a personal digital assistant (PDA) type phone, including both the device and its service. Our persona is Vivien Strong, a real-estate agent in Indianapolis, whose goals are to balance work and home life, close the deal, and make each client feel like he is her *only* client.

Vivien's context scenario:

1. While getting ready in the morning, Vivien uses her phone to check her e-mail. It has a large enough screen and quick connection time so that it's more convenient than booting up a computer as she rushes to make her daughter, Alice, a sandwich for school.
2. Vivien sees an e-mail from her newest client, Frank, who wants to see a house this afternoon. The device has his contact info, so now she can call him with a simple action right from the e-mail.
3. While on the phone with Frank, Vivien switches to speakerphone so she can look at the screen while talking. She looks at her appointments to see when she's free. When she creates a new appointment, the phone automatically makes it an appointment with Frank, because it knows with whom she is talking. She quickly enters the address of the property into the appointment as she finishes her conversation.
4. After sending Alice off to school, Vivien heads into the real-estate office to gather some papers for another appointment. Her phone has already updated her Outlook appointments, so the rest of the office knows where she'll be in the afternoon.
5. The day goes by quickly, and she's running a bit late. As she heads towards the property she'll be showing Frank, the phone alerts her that her appointment is in

15 minutes. When she flips open the phone, it shows not only the appointment, but a list of all documents related to Frank, including e-mails, memos, phone messages, and call logs to Frank's number. Vivien presses the call button, and the phone automatically connects to Frank because it knows her appointment with him is soon. She lets him know she'll be there in 20 minutes.

6. Vivien knows the address of the property but is a bit unsure exactly where it is. She pulls over and taps the address she put into the appointment. The phone downloads directions along with a thumbnail map showing her location relative to the destination.
7. Vivien gets to the property on time and starts showing it to Frank. She hears the phone ring from her purse. Normally while she is in an appointment, the phone will automatically transfer directly to voicemail, but Alice has a code she can press to get through. The phone knows it's Alice calling, and uses a distinctive ring tone.
8. Vivien takes the call — Alice missed the bus and needs a pickup. Vivien calls her husband to see if he can do it. She gets his voicemail; he must be out of service range. She tells him she's with a client and asks if he can get Alice. Five minutes later the phone makes a brief tone Vivien recognizes as her husband's; she sees he's sent her an instant message: "I'll get Alice; good luck on the deal!"

Notice how the scenario remains at a fairly high level, without getting too specific about interfaces or technologies. It's important to create scenarios that are within the realm of technical possibility, but at this stage the details of reality aren't yet important. We want to leave the door open for truly novel solutions, and it's always possible to scale back; we are ultimately trying to describe an *optimal*, yet still feasible, experience. Also notice how the activities in the scenario tie back to Vivien's goals and try to strip out as many tasks as possible.

Pretending it's magic

A powerful tool in the early stages of developing scenarios is to *pretend the interface is magic*. If your persona has goals and the product has magical powers to meet them, how simple could the interaction be? This kind of thinking is useful to help designers look outside the box. Magical solutions obviously won't suffice, but figuring out creative ways to technically accomplish interactions that are as close to magical solutions as possible (from the personas' perspective) is the essence of great interaction design. Products that meet goals with the minimum of hassle and intrusion seem almost magical to users. Some of the interactions in the preceding scenario may seem a bit magical, but all are possible with technology available today. It's the goal-directed behavior, not the technology alone, that provides the magic.

In early stages of design, pretend the interface is magic.

Step 5: Identifying requirements

After you are satisfied with an initial draft of your context scenario, you can analyze it to extract the personas' needs or requirements. These **requirements** can be thought of as consisting of *objects*, *actions*, and *contexts*.[11] And remember, as we discuss above, we prefer not to think of requirements as identical to features or tasks. Thus, a need from the scenario above might be:

Call (action) a person (object) directly from an appointment (context).

If you are comfortable extracting needs in this format, it works quite well; otherwise, you may find it helpful to separate them into data, functional, and contextual requirements, as described in the following sections.

Data requirements

Personas' data needs are the objects and information that must be represented in the system. Using the semantics described above, it is often useful to think of data requirements as the objects and adjectives related to those objects. Common examples include accounts, people, documents, messages, songs, images, as well as attributes of those such as status, dates, size, creator, subject, and so on.

Functional requirements

Functional needs are the operations or actions that need to be performed on the objects of the system and which are typically translated into interface controls. These can be thought of as the *actions* of the product. Functional needs also define places or containers where objects or information in the interface must be displayed. (These are clearly not actions in and of themselves but are usually suggested by actions.)

Other requirements

After you've gone through the exercise of pretending it's magic, it's important to get a firm idea of the realistic requirements of the business and technology you are designing for (although we hope that designers have some influence over technology choices when it directly affects user goals).

- **Business requirements** can include development timelines, regulations, pricing structures, and business models.

- **Brand and experience requirements** reflect attributes of the experience you would like users and customers to associate with your product, company, or organization.
- **Technical requirements** can include weight, size, form factor, display, power constraints, and software platform choices.
- **Customer and partner requirements** can include ease of installation, maintenance, configuration, support costs, and licensing agreements.

Having performed these steps, you should now have a rough, creative overview of how the product is going to address user goals in the form of context scenarios, and a reductive list of needs and requirements extracted from your research, user models, and the scenarios. Now you are ready to delve deeper into the details of your product's behaviors, and begin to consider how the product and its functions will be represented. You are ready to define the framework of the interaction.

Notes

1. Laurel, *Computers as Theater,* 134
2. Rheinfrank and Evenson, 1996
3. Carroll, 2001
4. Buxton, 1990
5. Verplank, et al, 1993
6. Wirfs-Brock, 1993
7. Constantine and Lockwood, 1999
8. Newman and Lamming, 1995
9. Holtzblatt and Beyer, 1998
10. Kuutti, 1995
11. Shneiderman, 1998