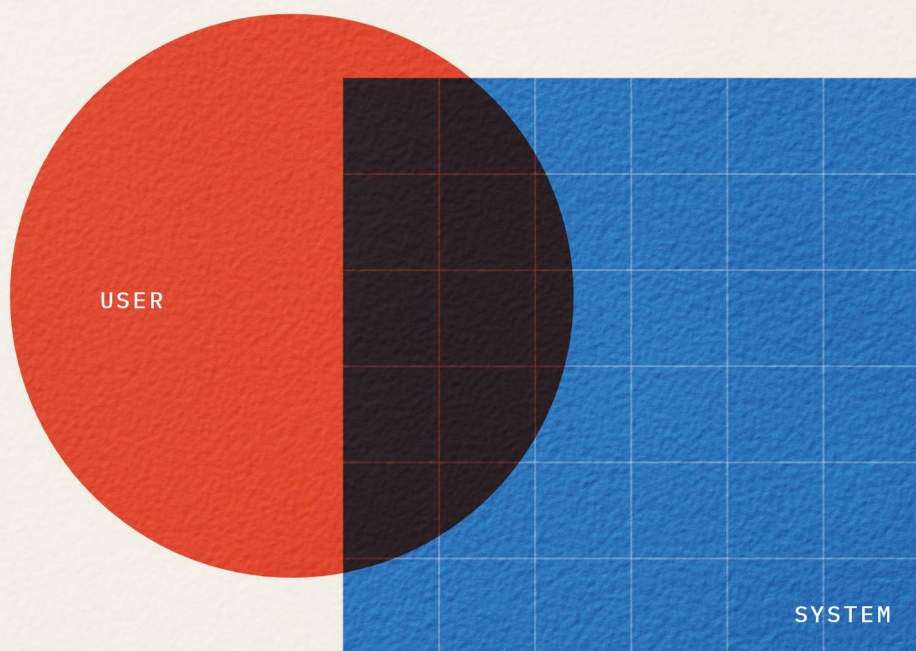


IZAIAS CAVALCANTI

From Cognitive Load to

Cognitive Distance



*Reframing Usability and Accessibility
in Information Systems*

From Cognitive Load to Cognitive Distance

From Cognitive Load to

Cognitive Distance

*Reframing Usability and Accessibility
in Information Systems*

IZAIAS CAVALCANTI

From Cognitive Load to Cognitive Distance: Reframing Usability and Accessibility in Information Systems

Copyright © 2026 Izaias Cavalcanti. All rights reserved.

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written permission of the author, except for brief quotations in reviews and scholarly works.

First edition, 2026.

Free digital edition, published before the print release.

<https://book.izaiax.xyz>

The prototype screens quoted in Chapter 6 are from the author's Everyday Services Study. All other product and company names mentioned in this book are used for illustration only and may be trademarks of their respective owners. Their mention does not imply endorsement.

Contents

Preface	ix
1 The Distance Problem	1
1.1 The Strange Persistence of Confusing Systems	1
1.2 When Systems Think Differently Than People	2
1.3 Usability Success That Still Feels Wrong	2
1.4 The Limits of Simplification	2
1.5 The Emergence of Cognitive Distance	3
1.6 Cognitive Load Versus Cognitive Distance	4
1.7 Where the Idea Comes From, and What It Is Not . . .	6
1.8 Why Cognitive Distance Matters Now	8
2 How Systems Drift Away From Human Thinking	9
2.1 The Designer’s Mental Model	11
2.2 Organizational Forces That Increase Distance	12
2.3 The Hidden Influence of Technology	13
2.4 Recognizing the Pattern	14
3 Mental Models and the Architecture of Understanding	15
3.1 How People Build Mental Models	16
3.2 When Mental Models Break Down	17
3.3 Mental Models in Complex Systems	18
3.4 The Designer’s Challenge	19
4 Usability Revisited: When Easy Does Not Mean Understandable	21

4.1	The Rise of Usability as a Discipline	22
4.2	When Usability Metrics Hide Deeper Problems	23
4.3	The Limits of Simplicity	24
4.4	A New Question for Usability	25
5	Accessibility and the Hidden Dimension of Cognitive Barriers	27
5.1	The Evolution of Digital Accessibility	28
5.2	Accessibility Beyond Sensory Barriers	29
5.3	Accessibility in Public Systems	30
5.4	Externalized Processing: When Systems Hand Their Work to Users	31
5.5	Rethinking Accessibility Through Cognitive Distance	34
6	Measuring the Invisible: Detecting Cognitive Distance in Real Systems	37
6.1	Why Traditional Usability Metrics Fall Short	37
6.2	Observing Mental Models	38
6.3	Behavioral Signals of Cognitive Distance	39
6.4	Quantifying Conceptual Alignment	40
6.5	The Cognitive Distance Index	41
6.6	Testing the Idea: The Everyday Services Study	43
6.7	Diagnosing Cognitive Distance in Practice	46
7	Design Assumptions and the Architecture of Misunderstanding	47
7.1	The Assumption of Rational Navigation	48
7.2	The Assumption of Shared Vocabulary	49
7.3	The Assumption of Linear Workflows	50
7.4	The Assumption of Static Understanding	50
7.5	Recognizing the Pattern	51
8	Designing for Alignment: Reducing Cognitive Distance	53
8.1	Designing Around Human Goals	54
8.2	Language That Matches Human Thinking	55
8.3	Designing Transparent Systems	56
8.4	Internalize Explanation	57

8.5	Encouraging Exploration Without Confusion	58
8.6	Designing Systems That Stay Understandable	59
9	Artificial Intelligence and the Expanding Distance	61
9.1	From Deterministic to Probabilistic Systems	62
9.2	Anthropomorphism and the Illusion of Understanding	63
9.3	Opacity and the Challenge of Explanation	64
9.4	Designing AI Systems That Humans Can Understand	67
9.5	The Future of Human–AI Interaction	68
10	Systems, Institutions, and the Distance Between People and Digital Power	69
10.1	Digital Bureaucracy	70
10.2	The Experience of Institutional Opacity	72
10.3	Power, Knowledge, and System Design	73
10.4	Toward Human-Centered Digital Institutions	74
11	A Practical Framework for Evaluating Cognitive Distance	77
11.1	From Usability Testing to Conceptual Evaluation . . .	78
11.2	The Cognitive Distance Diagnostic Model	79
11.3	Methods Designers Can Apply Immediately	81
11.4	Integrating Cognitive Distance Into Development Cycles	82
11.5	Organizational Adoption	83
12	Toward a New Paradigm in Human-Computer Interaction	85
12.1	Expanding the Theoretical Landscape	86
12.2	Implications for Design Education	87
12.3	Implications for Research	88
12.4	A Future of More Understandable Systems	89
13	Closing the Distance	91
13.1	What We Have Learned	92
13.2	A Practical Framework for Closing Cognitive Distance	92
13.3	Implications Beyond Design	95
13.4	The Role of the Designer and Researcher	95
13.5	The Distance Ahead	96
	References	97

About the Author

103

Preface

Why do so many digital systems work, yet still leave people confused, dependent or excluded? That question shaped this book.

For years, design teams have used cognitive load theory to explain failure (Kirschner 2002; Paas and Merriënboer 1994; Sweller 1988, 2011). When a system asks people to hold too much in working memory, performance drops. The response has been clear: reduce steps, simplify screens and remove visual clutter. Those changes help, but they do not solve the whole problem.

People still complete tasks and then say they are not sure what happened. Patients still call support lines for hospital portals that pass usability checks. Public websites meet accessibility rules and still feel hard to navigate. New AI products look simple and still create uncertainty. That recurring pattern points to a deeper issue. The challenge is not only effort; it is alignment.

This book introduces the idea of *cognitive distance*: the mismatch between the conceptual structure a system requires and the mental model a person actually builds while using it. When the distance is small, people feel oriented, confident and in control. When it is large, interaction becomes fragile, learning does not stick, and confidence falls, even when every task is completed on time.

The chapters draw on Human-Computer Interaction, cognitive psychology, public administration and organizational research, and on everyday cases from healthcare, government services, banking, enterprise software and AI systems (Card et al. 1983; Norman 1988; Seffah and Metzker 2004; Zhang et al. 2025). They also draw on my own research program on cognitive distance, including a measurement

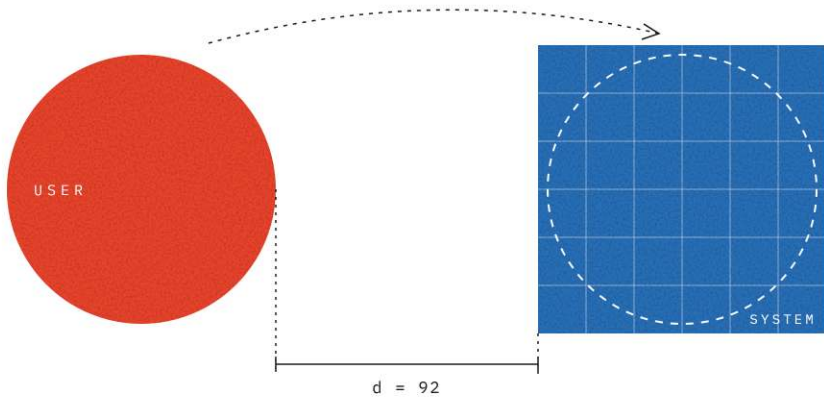
instrument, the Cognitive Distance Index, and an experiment that is testing the idea under controlled conditions. Together they show why high completion rates can hide deep structural problems, why compliance does not always produce clarity, and why better alignment produces better outcomes.

This is not a rejection of cognitive load theory. It is an expansion that makes design practice more complete.

If you are a designer, researcher, product leader or policymaker who feels that current usability language is missing something essential, this book offers a practical framework for diagnosing and reducing cognitive distance. Systems now shape daily life, public services and economic opportunity. In that context, alignment between systems and people is not a luxury; it is a responsibility.

The distance is real, and it can be designed.

Izaías Cavalcanti



CHAPTER 1

The Distance Problem

1.1 The Strange Persistence of Confusing Systems

For more than four decades, Human-Computer Interaction has tried to answer a simple question: why do people struggle with digital systems (Card et al. 1983; Norman 1988; Shneiderman 1983)? At first, the answer looked technical. Computers were new, interfaces were primitive, and users needed time.

Then everything improved. We moved from command lines to graphical interfaces. Processing power exploded. Design became a profession, and usability became a discipline. Still, confusion stayed.

People hesitate before they click, repeat actions to make sure a task really completed, and feel one mistake away from breaking something important. This happens in daily, high-stakes situations. Patients struggle with medical portals. Employees depend on internal help

desks for tools they use every week. Citizens abandon government forms midway through critical processes. AI tools produce fast outputs while users remain unsure how the results were generated. These are not rare edge cases; they are ordinary moments in modern digital life.

1.2 When Systems Think Differently Than People

Every system carries a model of the world. Designers define categories, labels and flows. Engineers define structures, states and constraints. Product teams define success paths. Users arrive with their own models, built from daily life, prior tools, work habits and cultural expectations.

When those models align, interaction feels natural. Users can predict outcomes and recover from mistakes with confidence. When they do not, interaction becomes translation work. Users stop focusing on their goals and start guessing what the system expects. Even a clean interface can feel mentally noisy when people must decode hidden logic at every step. This is where cognitive distance begins.

1.3 Usability Success That Still Feels Wrong

Traditional usability metrics matter. Time on task, completion rates and error rates all provide useful signals (Seffah and Metzker 2004). But these measures can hide a deeper problem: a user can complete a task and still not understand the system.

Many people succeed by memorizing sequences instead of building real understanding. The moment a label changes or a step moves, their performance collapses. The interface looked successful, but the relationship between user and system remained fragile. That fragility is expensive. It drives support costs, retraining cycles and persistent frustration (Ceaparu et al. 2004).

1.4 The Limits of Simplification

When teams see friction, they often simplify. Simplification is valuable, but on its own it is not enough. A minimal surface can still hide a confusing structure. A system may show fewer options while

preserving a model that does not match how people think, and in that case visual simplicity masks conceptual complexity.

The contrast is especially visible in AI products. A single input field looks easy, but behind it sits a complex system with unclear boundaries. Users must infer what to ask, what the model can do, and why the quality of its output changes. Low visible effort does not guarantee low cognitive distance.

1.5 The Emergence of Cognitive Distance

As usability practice matured, a repeating pattern became easier to spot. In interviews and think-aloud sessions, people said things like “I can use it, but I am never fully sure,” “I just follow the steps,” or “I don’t know why it works.” These are not statements about laziness or lack of skill. They describe a structural mismatch between the logic a system runs on and the reasoning a person brings to it.

This book calls that mismatch *cognitive distance*, and it uses the term in a precise sense. Figure 1.1 draws the idea.

Cognitive distance is the degree of mismatch between the conceptual structure that a system requires for correct interpretation and the mental model that a user actually constructs during interaction.

Low cognitive distance exists when people can predict what a system will do before they act, explain what happened afterwards in their own words, and carry that understanding into the next, slightly different situation. High cognitive distance exists when people complete actions without a coherent account of what produced the outcome, even when they act quickly and without any sense of strain.

Three properties distinguish the construct from the ideas it sits beside, and they recur throughout the book.

It is relational. Cognitive distance is not a property of the system alone or of the user alone. It describes the fit between a system’s structure and a particular population’s capacity to interpret it. The same result screen can sit close to a bank employee who knows

settlement terminology and far from a first-time customer who does not. Any measurement of distance therefore has to say whose distance is being measured.

It is cumulative. Mental effort rises and falls from moment to moment within a task. Distance accumulates across interactions and across design decisions. Each choice that favors internal system logic over user-facing interpretability adds a small increment, and each missed opportunity to explain an outcome adds another. This is why an evaluation conducted at a single moment can underestimate the long-term cost of an opaque system, and why Chapter 5 describes the result as a form of debt.

It can be high when effort is low. This is the defining property, and the main reason the concept is not redundant with existing frameworks. An interaction can be brief, clean and effortless while leaving the user with no idea what happened. The user presses *Confirm* and sees a result in under five seconds. The screen contains few words, no errors occur, and performance metrics report success. Cognitive distance is at its maximum.

Once this lens is applied, many familiar failures become easier to explain.

1.6 Cognitive Load Versus Cognitive Distance

Cognitive load and cognitive distance are related, but they are not the same. Cognitive load asks how much mental effort a task requires right now (Sweller 1988, 2011). Cognitive distance asks how far a system is from the way its users frame the task. Put another way, load concerns the cost of performing actions a person understands; distance concerns the gap that exists before the person even knows what is being asked.

Because the two are separate, a flow can have low load and high distance, or moderate load and low distance. Booking travel involves several steps, yet most people understand the sequence because it matches everyday planning logic. By contrast, a short form can feel difficult if its order and language reflect internal data structures rather

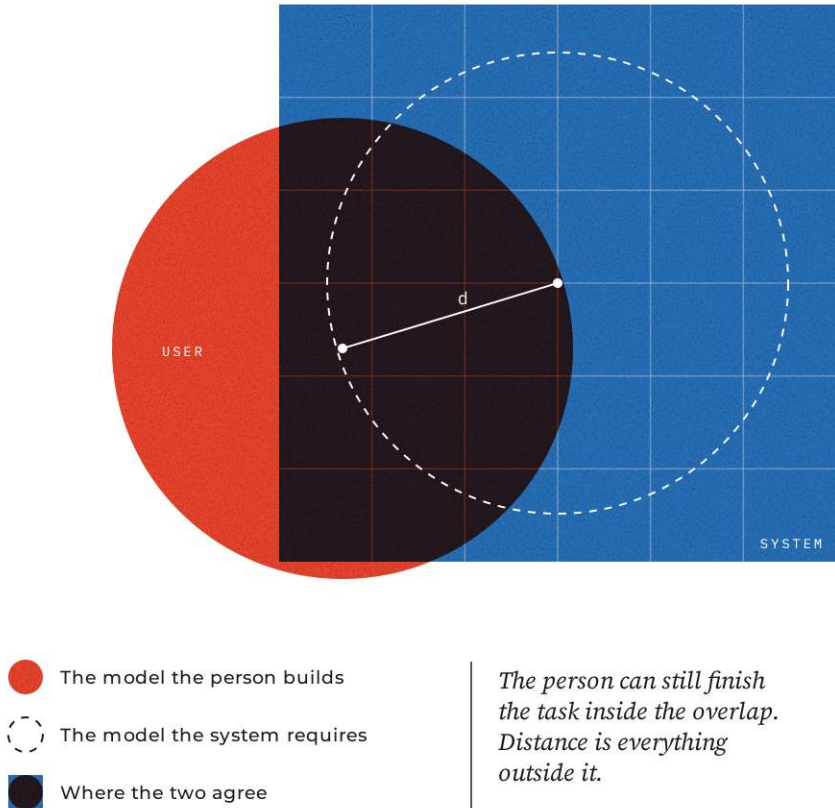


Figure 1.1. Cognitive distance. The person's model (red) and the model the system requires (dashed). The task can still be completed inside the overlap; the distance is everything outside it.

than user goals. Design improves fastest when teams measure both effort and alignment. Figure 1.2 shows the four conditions that result.

	LOW LOAD	HIGH LOAD
HIGH DISTANCE	Easy but opaque Interaction looks easy, but understanding is fragile. Users finish without confidence. This is the failure load-based evaluation misses.	Demanding and opaque The most fragile condition. Tasks feel difficult and the system's logic feels alien.
LOW DISTANCE	Effortless and clear Fast, clear interactions. Users understand what to do and why it works.	Demanding but transparent Effort is high, but users build stable understanding because the structure is coherent.

Figure 1.2. The load–distance matrix. Effort and conceptual alignment vary independently, producing four interaction conditions. The upper-left quadrant is the one this book is most concerned with.

1.7 Where the Idea Comes From, and What It Is Not

The phrase *cognitive distance* is newer than most of the ideas it brings together, and it is not new in every field. Readers who have met it elsewhere deserve a clear account of how this book’s use relates to earlier ones.

Norman’s gulfs. The closest ancestor in Human-Computer Interaction is Donald Norman’s distinction between the *gulf of execution*, the gap between what a person wants to do and the actions a system makes available, and the *gulf of evaluation*, the gap between what a system shows and what the person can conclude about its state (Norman 2013; Norman and Draper 1986). Hutchins, Hollan and Norman went further and described two kinds of distance a user must bridge: semantic distance, between what the user wants to express and what the interface’s language can express, and articulatory distance, between that meaning and the form of the input. They argued that direct manipulation feels natural because it shortens both (Hutchins et al. 1985). Cognitive distance, as used here, is closest to the gulf of

evaluation, but it differs in two ways. It concerns the user's accumulated model of *why* the system behaves as it does, not only whether its current state is legible. And it is meant to be measured, with the instrument described in Chapter 6, rather than only described.

Mental model research. A long tradition in cognitive psychology studies the internal representations people build of devices and situations (Gentner and Stevens 1983; Johnson-Laird 1983; Kieras and Bovair 1984). That research asks a descriptive question: what model has this person built? Cognitive distance asks an evaluative one: how far is that model from the one the system requires? The second question is more directly useful to designers, because it supplies both a criterion for comparison and a signal for intervention.

Spatial cognition. In environmental psychology, cognitive distance means a person's mental estimate of the physical distance between places, which departs systematically from measured distance (Montello 1991). The methods Montello reviewed for eliciting and validating such estimates are a useful reminder that a mental quantity can be measured rigorously, but the construct itself is unrelated: it concerns geography, not systems.

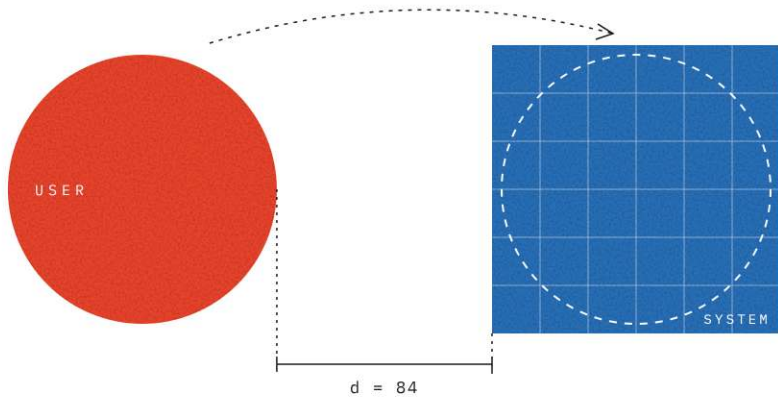
Organizational learning. In economics and innovation studies, Bart Nooteboom uses cognitive distance for the difference in knowledge and perspective between organizations, arguing that partners learn most at an intermediate distance: close enough to understand each other, far enough to have something new to offer (Nooteboom 2000; Nooteboom et al. 2007). This book borrows nothing from that model except the spatial metaphor. Here, distance is never a resource to be optimized. It is a cost that falls on the person trying to use a system, and the design goal is always to reduce it.

Cognitive distance does not replace any of these contributions. It gives a single name, a working definition and a measurement approach to a failure that each of them touches but none of them isolates: the user who succeeds procedurally and fails interpretively.

1.8 Why Cognitive Distance Matters Now

Today, digital systems sit inside almost every important human process. Healthcare, education, finance, employment and public services all depend on tools that people must understand under pressure. Meanwhile, systems are becoming more adaptive and more opaque. Recommendation engines, automated decisions and AI-generated content can feel useful yet remain hard to interpret (Weiser 1991; Zhang et al. 2025). As opacity grows, conceptual alignment becomes more important, not less.

If users cannot form stable mental models, they cannot build durable trust, and when trust erodes, adoption becomes shallow and fragile. Chapter 2 explores how misalignment grows during design and delivery, and why even well-intentioned teams drift away from human reasoning over time.



CHAPTER 2

How Systems Drift Away From Human Thinking

Digital systems rarely begin with the intention of confusing anyone. In the early stages of design, teams imagine clear workflows, intuitive navigation and smooth interactions. Yet somewhere between the first idea and the final product, something shifts, and a system that seemed obvious during development feels strangely foreign to the people who eventually use it.

This drift does not happen because designers are careless. It happens because systems are shaped by several forces that gradually pull them away from the structure of everyday human reasoning. Understanding those forces is the first step towards reducing cognitive distance. Figure 2.1 summarizes a common path.

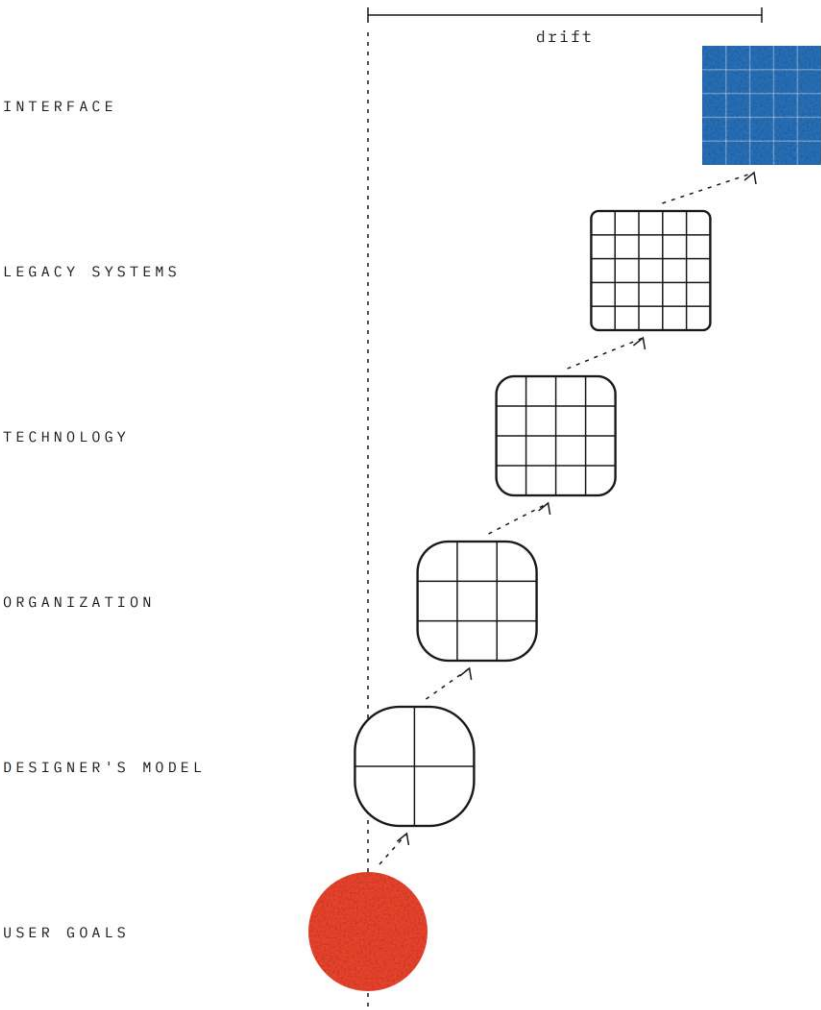


Figure 2.1. Drift. Each layer, from the designer's model through the organization, technology and legacy systems to the interface, turns the user's goal a little more into the system's grid and moves it further from where it started.

2.1 The Designer's Mental Model

Designers inevitably design for themselves first

Every designer begins with a mental image of how a system should work. That image is shaped by their training, their professional tools and the internal logic of the organization building the product. When a designer imagines a system, the structure usually feels perfectly reasonable, because it fits neatly inside their own conceptual framework.

Users approach the same system from very different starting points. Designers think in terms of components, flows and architecture. Engineers think in terms of data structures and system states. Users think in terms of goals, situations and outcomes. These perspectives rarely align perfectly. A designer may organize an interface around functional modules while the user simply wants to complete a real-world task, and when that happens the system reflects the designer's logic rather than the user's. This is one of the earliest sources of cognitive distance.

Expertise quietly reshapes perception

As professionals gain experience, they perceive systems differently from novices. Experts learn to compress complex information into meaningful patterns, so what looks complicated to a beginner looks simple to someone with years of practice. Research on expertise has repeatedly found that experts are poor at predicting how hard a task will be for novices, and that the bias persists even when they are warned about it (Hinds 1999). Education researchers call the same effect the *expert blind spot* (Nathan and Petrosino 2003).

In design, it means that designers and engineers underestimate the conceptual effort a new user faces. A navigation structure that feels self-explanatory to the team may appear arbitrary to someone meeting it for the first time. The distance between expert perception and novice reasoning becomes embedded in the interface, and over time that gap hardens into the structure of the product itself.

Internal system logic becomes the interface

Another common pattern appears when internal architecture directly shapes the interface. Engineering teams organize systems according to data models, database schemas or back-end services. These structures are efficient for development and maintenance, but they do not necessarily reflect how people think about tasks. A database may separate information into categories that make sense for storage, and users then experience those categories as artificial divisions that interrupt the flow of what they are trying to do. When internal architecture leaks into the interface, users are forced to navigate the system's logic rather than their own, and every task acquires a quiet layer of translation.

2.2 Organizational Forces That Increase Distance

Products are built by organizations, not individuals

Even when designers recognize a potential problem, the final structure of a system is rarely decided by one person. Large digital systems emerge from organizations in which several teams contribute features, policies and constraints, each with its own priorities. Product managers focus on roadmaps and business goals, engineers on stability, legal teams on compliance and marketing teams on messaging. Each of these priorities can reshape the interface. The result is often a layered system in which the user's mental model is only one influence among many, and the interface ends up reflecting organizational structure as much as user needs.

Conway's Law quietly shapes interfaces

In 1968 the computer scientist Melvin Conway observed that organizations design systems that mirror their own communication structures (Conway 1968). The observation became known as Conway's Law, and its consequences are visible across digital products. A website may have separate sections controlled by different teams; a software platform may include features developed independently by different groups. From inside the organization these divisions feel natural. From outside, the system looks fragmented: each section follows slightly different conventions, navigation behaves inconsistently, and

terminology changes from one context to the next. Users have to adjust their mental model every time they cross an internal boundary they cannot see.

Incremental growth produces structural drift

Very few systems are designed all at once. Most evolve gradually as features are added, workflows expand and infrastructure adapts to new requirements. Software engineering research has long recognized that a system's structure degrades as it is changed, unless effort is spent to counter that decay (Lehman 1980). Early decisions were made under one set of assumptions and later additions respond to new goals, so the system becomes a patchwork of design eras. One part follows a task-based structure, another relies on hierarchical navigation, and a third reflects internal administrative categories. Each layer adds to the translation users must perform, and the result is a system that works in fragments but is difficult to understand as a whole.

2.3 The Hidden Influence of Technology

Technical constraints shape interaction

Every interface exists within technical boundaries. Processing power, database architecture, security protocols and integration requirements all limit what designers can realistically build, and sometimes these constraints force interaction patterns that feel unnatural. Security requirements may demand frequent authentication; data synchronization may introduce delays that interrupt a task; legacy systems may restrict how information can be reorganized. Users rarely see these limitations. They only experience the resulting behavior, and from their perspective the system simply feels confusing or inconsistent. Such compromises can widen cognitive distance even when designers work hard to minimize it.

Legacy systems shape modern interfaces

Many digital systems are not built from scratch but on top of older infrastructure that was never designed for modern interaction. Government systems provide many examples. Public service portals often

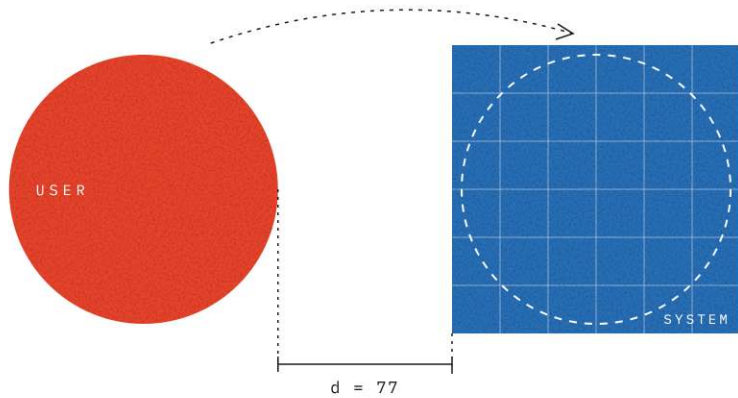
connect decades-old databases to newer web interfaces, and while the interface may look modern, the underlying logic still reflects the legacy system. This mismatch produces strange workflows. Users enter the same information several times, tasks are split across unrelated pages, and navigation follows administrative categories rather than real-world goals. The interface acts as a thin layer over an older conceptual model, and users sense the misalignment immediately.

Automation introduces new conceptual gaps

As systems incorporate automation and artificial intelligence, another source of distance emerges. Automated systems often operate through statistical models or algorithmic rules that do not map onto traditional interface metaphors. A recommendation engine suggests products based on behavioral patterns; an AI assistant generates text by predicting likely sequences of words. To engineers these mechanisms make sense. To users the behavior can appear unpredictable or mysterious, and without clear explanations people struggle to build stable mental models of it. The gap is no longer only between interface structure and human reasoning. It also includes the invisible logic of algorithms operating behind the interface, a theme Chapter 9 takes up in detail.

2.4 Recognizing the Pattern

Taken together, these forces show that cognitive distance is rarely the result of a single mistake. It grows gradually as systems evolve under several influences at once. Designer expertise introduces subtle assumptions, organizational structures shape navigation and feature boundaries, technical constraints impose unexpected interaction patterns, and automation adds opaque decision processes. Each moves the system slightly further from the user's way of thinking. The distance accumulates until people encounter a system that technically works but conceptually feels foreign. Seeing the process clearly is the first step towards designing systems that stay aligned with human reasoning.



CHAPTER 3

Mental Models and the Architecture of Understanding

When people encounter a new system, they do not begin from zero. They immediately start forming explanations. Sometimes these explanations are correct, often they are incomplete, and occasionally they are entirely wrong, yet they still guide behavior. Researchers call this internal explanation a *mental model* (Gentner and Stevens 1983; Johnson-Laird 1983).

Mental models let people predict what will happen when they act. They provide a sense of structure even when the system itself is complex, and without them interaction would feel chaotic. Cognitive distance emerges precisely when the model a user constructs cannot align with the conceptual structure embedded in the system,

which makes understanding how mental models form central to understanding the distance problem.

3.1 How People Build Mental Models

Understanding evolves through repeated cycles of action and feedback, summarized in Figure 3.1.

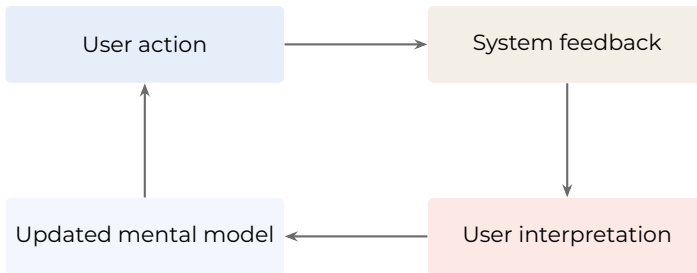


Figure 3.1. The mental-model loop. Each action–feedback cycle confirms or revises the user’s explanation of how the system works.

Humans naturally search for causal explanations

When people use a tool, they instinctively try to understand how it works, and they build explanations even when information is incomplete. Someone using a microwave oven may know nothing about electromagnetic radiation, yet they still develop a working account of how the device behaves: pressing certain buttons produces certain outcomes. Digital systems trigger the same process. When users interact with software, they observe outcomes and form hypotheses. If clicking a button produces a result, they associate the two; if an action produces an unexpected response, they revise their explanation. Over time these observations form a personal model of the system. It may not reflect the system’s true internal logic, but it allows the user to operate within it (Kieras and Bovair 1984).

Experience from the physical world shapes digital expectations

Human reasoning about tools developed long before digital technology existed, and people carry expectations from physical interaction into digital environments. Opening a folder on a computer borrows

from the physical act of organizing documents; dragging a file to a trash icon echoes the act of discarding something. Early interface design relied on these metaphors deliberately. Douglas Engelbart's work on augmenting human intellect, Alan Kay's vision of personal dynamic media, and the Xerox Star's desktop of documents and folders all tried to make computers understandable by connecting digital actions to familiar concepts (Engelbart 1962; Kay and Goldberg 1977; Smith et al. 1982).

These metaphors helped people form mental models quickly. As systems have grown more complex, however, the connection to physical experience has weakened. Many modern interfaces rely on abstract concepts with no real-world counterpart, and when that happens users struggle to build reliable models.

Learning through exploration and error

Most people do not read manuals before using software. They explore: they click, observe the result, and gradually infer how the system behaves. The process resembles scientific experimentation in miniature. A user forms a hypothesis about what an action might do, performs it, and observes the outcome. If the outcome matches the expectation, the model strengthens; if it contradicts it, confusion appears. When systems behave consistently, exploration is a powerful way to learn. When they behave unpredictably, users cannot refine their understanding, and the model stays unstable. That instability is one of the clearest signs that cognitive distance is present.

3.2 When Mental Models Break Down

Systems that hide their internal structure

Some systems obscure the logic behind their behavior, showing results without revealing how they were produced. Search engines are a classic example. A user enters a query and receives a ranked list, but the system rarely explains how the ranking was made. Users therefore construct simplified explanations. They may believe certain keywords always produce better results, or assume the first result is always the most accurate. These shortcuts help them navigate, even

though the true mechanism is far more complex. But if the system changes its behavior, the user's model can collapse suddenly.

Interfaces that contradict user expectations

A different breakdown occurs when an interface violates intuitive expectations. A user expects the back button to return them to the previous screen. If it instead takes them somewhere else or resets part of their progress, their model becomes unreliable. These small violations accumulate. After several unexpected outcomes, users lose confidence in their understanding and stop reasoning through the interface. Instead, they rely on memorized sequences, following steps they learned earlier without knowing why those steps work. The shift is subtle but important: the system has become operable rather than understandable.

Terminology that reflects system logic rather than human language

Language plays a powerful role in shaping mental models. When labels reflect the vocabulary of developers or organizations rather than users, interpretation becomes difficult. Consider an option labeled "data reconciliation module". An engineer knows exactly what it means. A user who simply wants to check their financial information has to translate the system's language into their own, and every such translation adds to the distance. This is why usability guidelines insist on using the user's language rather than the system's; Jakob Nielsen's heuristic of a "match between system and the real world" states the principle directly (Nielsen 1994).

3.3 Mental Models in Complex Systems

The challenge of invisible processes

Modern systems rely on processes that operate far beyond the visible interface. Cloud computing, machine learning models and distributed data processing add layers of complexity that users never see. From the user's side, the system simply produces outputs while the reasoning behind them stays hidden, so people must guess. They may attribute human-like reasoning to algorithmic processes, or assume

the system understands their intentions when it is following statistical patterns. That gap between perceived intelligence and actual mechanism increases cognitive distance dramatically.

Artificial intelligence and unstable models

Generative AI adds another layer (Zhang et al. 2025). These systems produce responses that appear coherent and conversational, and users often read that fluency as evidence of deep understanding, when the underlying mechanism is a statistical model trained to predict likely continuations of text (LeCun et al. 2015). However powerful such systems are, their internal logic is rarely transparent to everyday users, and the mental models people build of them are unstable. Users believe they understand the system until it behaves unexpectedly. When it gives a wrong answer or contradicts itself, their explanation collapses and the interaction suddenly feels unreliable.

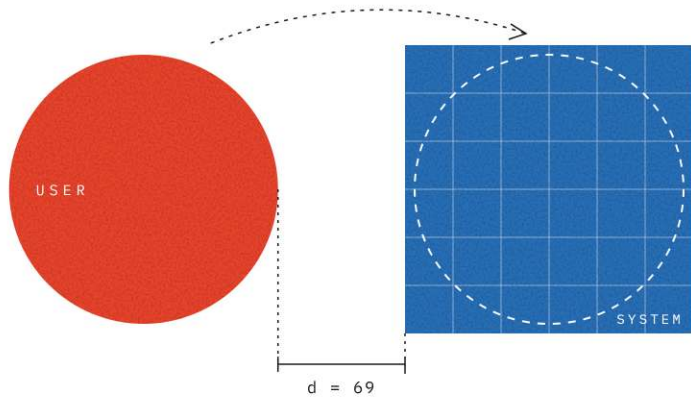
Trust and cognitive alignment

Mental models are closely tied to trust. When people believe they understand how a system behaves, they feel confident predicting its outcomes, and that confidence supports consistent use. When they cannot build a stable model, trust weakens. They may keep using the system, but cautiously: they double-check results and hesitate before relying on automated decisions (Lee and See 2004). In high-stakes domains such as healthcare and financial services, that uncertainty can have serious consequences. Reducing cognitive distance therefore matters beyond usability. It affects how reliably people can work with increasingly powerful systems.

3.4 The Designer's Challenge

Designers face a difficult task. They must create interfaces that let people build accurate mental models without exposing the full complexity underneath. Too much technical detail overwhelms; too little explanation leaves people guessing. The goal is not to teach users the entire architecture of a system. It is to provide conceptual structures that let them predict outcomes and understand the logic of their interactions. That balance requires careful decisions: metaphors

chosen thoughtfully, terminology drawn from real user language, and interaction patterns that behave consistently. When these elements align, users develop mental models that mirror the system's structure. When they do not, cognitive distance grows.



CHAPTER 4

Usability Revisited: When Easy Does Not Mean Understandable

For decades, usability has been one of the central pillars of Human-Computer Interaction. Whole industries formed around improving it, organizations hired usability specialists, and universities built research labs to study it. The effort produced real progress. Interfaces became cleaner, navigation improved, and tasks that once required training manuals became manageable through visual cues.

Yet a persistent tension remained. People often reported that systems were easy to use while expressing uncertainty about how those systems actually worked. They could follow the steps and complete the task, but the logic underneath stayed unclear. The tension reveals an important distinction: ease of use and conceptual understanding are not the same thing. A system may be operationally

simple and still leave people conceptually disoriented, and that gap is where cognitive distance becomes visible.

4.1 The Rise of Usability as a Discipline

Early usability research focused on efficiency

Usability research grew from early studies of how people interact with machines. As computers spread through workplaces in the late twentieth century, researchers began measuring how quickly people could complete tasks with different interface designs (Card et al. 1983). The measures were quantitative: task completion time, error rates, and the number of steps needed to finish a task. They exposed inefficient structures and showed that careful design could substantially reduce the effort of using a system.

Jakob Nielsen did more than anyone to bring these ideas into mainstream practice. Through books, research and consulting, he popularized evaluation methods designers could apply quickly, and showed that even simple testing with a handful of participants could uncover major design problems (Nielsen 1993).

The emergence of usability heuristics

To help designers find common problems before testing, Nielsen and Rolf Molich proposed a short set of usability heuristics, later refined by Nielsen into the list most practitioners know today (Nielsen 1994; Nielsen and Molich 1990). They include visibility of system status, a match between the system and the real world, consistency and standards, and error prevention. Each names a common source of friction, and when systems violate them people tend to become confused or frustrated. The heuristics remain highly influential because they give teams a fast, practical way to evaluate an interface before it reaches the public.

Usability success stories

Many widely used products improved dramatically through this work. Early computer interfaces required people to understand file paths and command syntax; modern ones hide those details behind visual metaphors and simplified navigation. The graphical interface

popularized by the Apple Macintosh, building on the Xerox Star, made windows, folders and icons the everyday vocabulary of computing, letting people work with familiar conceptual structures instead of abstract commands (Smith et al. 1982).

These successes demonstrated the value of human-centered design. They also reinforced a particular way of thinking about usability, one that associated it above all with efficiency and simplicity.

4.2 When Usability Metrics Hide Deeper Problems

Task completion does not guarantee understanding

Usability studies typically rely on task-based evaluation. Participants perform specific actions while researchers record success rates and completion times, and if most participants succeed the interface is considered usable (Hornbæk 2006). This method does not always reveal how well people understand the system. A participant may succeed simply by following visible cues: clicking the only obvious button, or tracing a highlighted path through the screens. By the metrics, the interaction succeeded. But the participant may still lack a clear model of how the system works, and if the interface changes slightly, or the task appears in a different context, they may struggle again. The evaluation captured surface performance, not conceptual understanding.

The phenomenon of procedural learning

Many digital interactions rely on procedural learning rather than conceptual understanding. People memorize a sequence of actions that reliably produces the result they want. An employee using enterprise software may remember that generating a report means opening a particular menu, selecting two options and clicking a confirmation button. They know the sequence works, but not why the system is structured that way, and if the menu moves or the interface is redesigned the procedure breaks and they must rebuild it from scratch. Cognitive psychology has documented this dissociation for decades: people can improve at a task without acquiring verbalizable knowledge of the rules that govern it (Berry and Broadbent 1984). The

system never became part of a coherent mental model. It remained a set of instructions.

Usability optimization can simplify the wrong structure

Teams sometimes try to reduce cognitive load by simplifying interface elements without reconsidering the structure underneath. Buttons get larger, menus get shorter and visual clutter disappears. These changes can improve efficiency, but if the conceptual structure is still misaligned with how people reason, the fundamental problem persists. The interface becomes easier to operate while still requiring people to adapt to its logic. The experience feels smoother without becoming more understandable, and the cognitive distance is largely unchanged.

4.3 The Limits of Simplicity

Simplicity can conceal complexity

Modern interface design celebrates minimalism. Clean layouts, limited options and streamlined workflows are widely treated as hallmarks of good design, and they can indeed reduce cognitive load. But minimalism sometimes conceals complexity rather than resolving it. When systems hide too much, people lose the ability to see how actions connect to outcomes. The interface looks simple while remaining conceptually opaque, and people can perform tasks without being able to explain the logic behind them.

Automation reduces visible effort but increases abstraction

Automation complicates the relationship further. Automated systems remove steps people used to perform themselves, such as organizing files, recommending actions or generating content, and this often improves efficiency. But it also introduces new conceptual layers. When a system acts on the user's behalf, the user may lose sight of how the action happened. The interaction becomes effortless but less transparent. Generative AI tools that produce complex outputs from a single prompt are the clearest current example, and Chapter 9 examines them in detail. Cognitive distance grows when system logic becomes invisible.

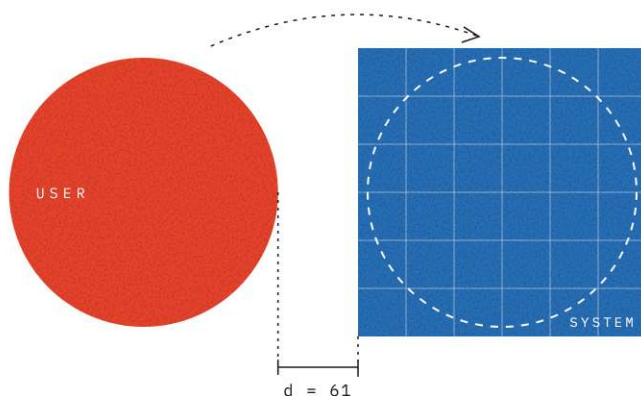
The illusion of intuitive design

Designers often describe successful interfaces as intuitive, implying that people can operate them naturally without conscious effort. But intuition does not appear spontaneously. It emerges from familiarity: an interface feels intuitive when it matches patterns people have already learned elsewhere. When designers assume their own sense of simplicity is universal, problems follow, because what feels obvious to one group can feel unfamiliar to another. Without research into users' actual mental models, "intuitive" becomes a misleading word.

4.4 A New Question for Usability

None of this suggests abandoning usability principles. Efficiency, clarity and error prevention remain essential. They are simply not sufficient on their own. Usability traditionally asks a straightforward question: *can users complete the task?* Cognitive distance adds a second: *do users understand what the system did, and why?*

The second question shifts the focus of design. It asks researchers and practitioners to examine not only how easily people perform actions, but how well those actions align with their reasoning. When usability and conceptual alignment move together, interaction is both efficient and meaningful. When they diverge, systems can look successful while quietly producing confusion.



CHAPTER 5

Accessibility and the Hidden Dimension of Cognitive Barriers

Accessibility is widely recognized as a moral, legal and social responsibility in digital design. Governments mandate it, organizations publish accessibility statements, and design teams apply accessibility guidelines during development. These efforts have improved many aspects of digital interaction. Text can be read by screen readers, images carry descriptive labels, and interfaces support keyboard navigation. For people with visual or motor impairments, these improvements are essential.

Yet many people still meet barriers in systems that technically comply with accessibility standards. The interface is readable, navigation works, and assistive technologies behave as intended, but the system is still hard to understand. The problem lies not in sensory access but

in conceptual alignment, and this is where cognitive distance enters the conversation about accessibility.

5.1 The Evolution of Digital Accessibility

Accessibility emerged from civil rights movements

The modern accessibility movement grew out of broader struggles for equal participation in public life. Advocates argued that physical spaces, public services and institutions should be designed so that people with disabilities could take part fully in society. In the United States these principles were codified in the Americans with Disabilities Act of 1990, which required public and private institutions to remove barriers that kept people with disabilities from services and facilities (United States Congress 1990). As daily life moved online, the same principles extended into digital environments, and websites, applications and online services gradually became subject to accessibility requirements.

The rise of web accessibility standards

To guide designers and developers, international bodies created formal standards for digital accessibility. The most widely recognized is the Web Content Accessibility Guidelines (WCAG), developed by the World Wide Web Consortium (W3C Web Accessibility Initiative 2023b). WCAG organizes accessibility around four principles: content should be perceivable, interfaces operable, information understandable, and systems robust enough to work across technologies. Many governments and institutions now require conformance, and the framework has substantially improved the accessibility of digital services.

Accessibility successes and limitations

These standards have produced real gains. Screen readers let blind and partially sighted people navigate websites, captions give deaf people access to video, and keyboard navigation helps people who cannot use a mouse. In practice, though, accessibility evaluation tends to focus on technical conformance. Teams check that images have alt text, that color contrast meets the required ratio, and that every interactive element can be reached from the keyboard. These

checks are necessary, but they do not reach deeper conceptual barriers. A system can meet every technical requirement while presenting information in ways many people find hard to interpret. Cognitive distance explains why.

5.2 Accessibility Beyond Sensory Barriers

Cognitive accessibility remains underexplored

Visual and motor accessibility receive most of the attention; cognitive accessibility has historically been harder to define. Cognitive barriers arise when systems demand forms of reasoning or memory that many people find difficult. They affect people with learning disabilities, people experiencing cognitive fatigue, older adults navigating unfamiliar technology, and anyone using a system under stress or time pressure. Even people with no diagnosed condition meet them when systems present complex structures or unfamiliar terminology. Accessibility frameworks increasingly recognize these issues (W3C Web Accessibility Initiative 2023a), but practical methods for addressing them remain far less developed than those for sensory access.

Complexity as a hidden barrier

Many digital services assume that people can easily navigate hierarchies, interpret administrative terminology and remember multi-step processes. Those assumptions do not always hold. Consider an online government service that requires citizens to pass through several nested categories before reaching the right form. Each step may be technically accessible and work perfectly with assistive technology, yet the structure may still be hard to interpret, because users must understand the administrative logic of the organization providing the service. When that logic differs from how citizens think about their own needs, confusion follows. The system is accessible in a technical sense while remaining conceptually distant.

Cognitive overload versus cognitive distance

Discussions of cognitive accessibility usually focus on overload. Interfaces that present too much at once overwhelm people, particularly those with attention or memory limitations, so guidelines encourage

less clutter and simpler layouts. These practices help, but overload and distance are different problems. A system can present very little information and still require people to interpret an unfamiliar conceptual structure. The difficulty then lies not in the quantity of information but in the mismatch between how the system is organized and how people reason. Reducing visual complexity does not solve it. Addressing cognitive distance means examining how systems represent tasks and information in the first place.

5.3 Accessibility in Public Systems

Government services and structural complexity

Public service platforms often reveal the consequences of cognitive distance more clearly than commercial products. Government systems are typically built around administrative categories rather than everyday goals. A citizen simply wants to renew a license, apply for a benefit or report a change of address, but the system is organized by department. They must first work out which agency handles the service and then pass through procedural steps that each reflect internal logic. Public administration researchers describe the cumulative cost of this as *administrative burden*: the learning, compliance and psychological costs that institutions shift onto the people they serve (Herd and Moynihan 2018; Moynihan et al. 2015). Recent work shows that bureaucratic language in digital public encounters measurably affects how citizens understand and respond to them (Li and Liu 2025). Even when these systems conform to accessibility guidelines, many citizens struggle with them, and the problem lies less in technical accessibility than in conceptual misalignment.

Healthcare portals and patient understanding

Healthcare platforms offer another example. Many hospitals and clinics now provide portals where people can view test results, book appointments and message clinical staff. These systems improve access to personal health information, yet reviews of the research have found that patients' ability to understand and act on what they see is uneven, and that health literacy shapes who benefits (Irizarry et al. 2015). Medical terminology, complex data displays and fragmented

workflows all create confusion. A patient may log in and navigate every page successfully and still be unsure what the information actually means. Here cognitive distance affects not only usability but health outcomes, since misreading medical information can influence treatment decisions and confidence.

Digital inclusion requires conceptual alignment

As governments and institutions rely more on digital systems, accessibility becomes inseparable from digital inclusion. If citizens cannot understand how a system works, they may be excluded from services that are technically available to them. This form of exclusion is subtle. The system appears open to everyone, but people who cannot bridge the conceptual gap abandon the process or depend on intermediaries to complete it for them. Reducing cognitive distance is therefore part of inclusive design. Accessibility is not only about enabling interaction; it is about enabling understanding.

5.4 Externalized Processing: When Systems Hand Their Work to Users

Some of the clearest examples of cognitive distance appear in tasks that look trivial. Consider a document submission form, the kind used to upload an identity card for a bank account, a medical report for an insurance claim, or proof of address for a public service.

From the system's side, the requirements are simple and precise. The file must be a PDF, JPG or PNG. It must be under two megabytes. It may need a particular name or a single page. From the user's side, the same requirements arrive as a sequence of implicit challenges. A format requirement assumes the person knows what a file format is, how formats differ and how to convert between them. A size limit assumes they know what file size represents and how to change it. Before the person can decide whether to perform the required action, they must first understand what the action means.

What people do next is revealing. They search for online converters, install compression apps, photograph the document again at a lower resolution, or send the file to themselves through a messaging app because they have noticed it makes images smaller. Call this

observable behavior *user-side preprocessing*. It is a workaround for a design decision: the system could reasonably convert, compress or normalize the file itself, within its technical, legal and security constraints, but instead it filters input and rejects what does not fit (Cavalcanti 2026b; Zaheri et al. 2025).

This book uses two terms for the pattern. *Externalized processing* is the allocation of reasonably automatable transformation work to users rather than to the system. It does not cover the input people necessarily provide to state their intent, supply source material or authorize an action; it refers only to avoidable work the system could absorb but does not. *Constraint without accommodation* names the interaction pattern that results: the system states the conditions valid input must meet but offers no internal means of meeting them. Figure 5.1 pictures the transfer.

Externalized processing raises both cognitive load and cognitive distance, and it does so unevenly. A user with technical experience mostly experiences extra effort: they understand what is wanted and spend time producing it. A user without that experience mostly experiences distance: they do not know what the system is asking, how to comply, or why their last attempt failed, and more effort does not help, because what they lack is a conceptual basis for acting. The pattern therefore falls hardest on the people accessibility standards are meant to protect, while passing every technical conformance check.

The research behind this book states the relationship as a proposition rather than a tested result: systems that externalize automatable preprocessing without explanation or built-in assistance are likely to show higher cognitive distance, visible as lower predictive accuracy, lower interaction confidence and lower conceptual transparency (Cavalcanti 2026b). Those three terms are dimensions of the measurement instrument introduced in Chapter 6.

Seen more broadly, externalized processing is one instance of a general move: the transfer of interpretive work from systems to users. A status code in place of a status explanation hands the user the job of translating the code. An administrative stage label in place of a next step hands them the job of decoding institutional logic. Chapter 8

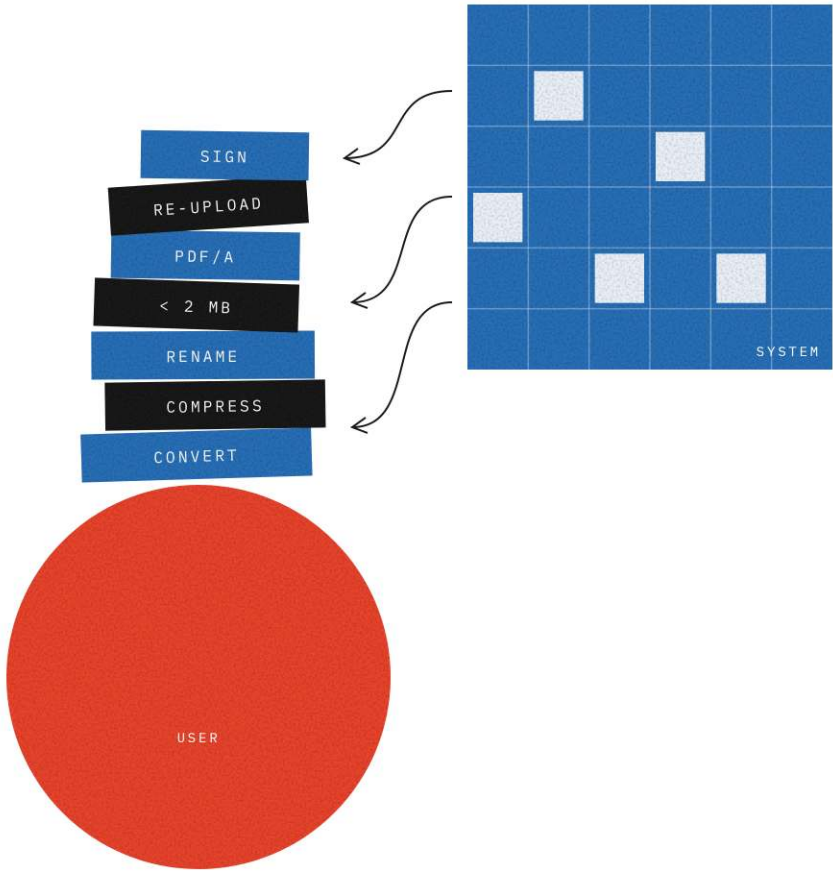


Figure 5.1. Externalized processing. Work the system could do itself (conversion, compression, renaming) is handed to the person, who has to carry it.

returns to the design response, which can be stated in two words: *internalize explanation*.

Accessibility debt

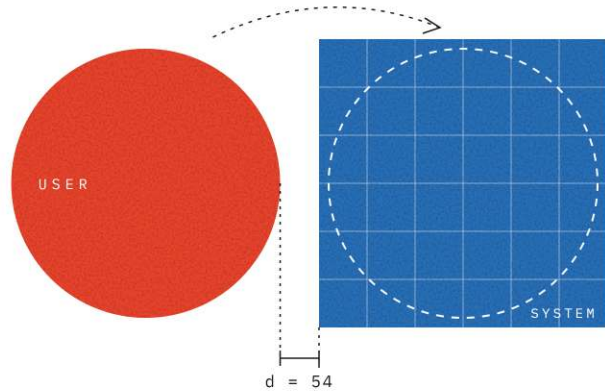
Because distance is cumulative, its costs behave less like isolated defects and more like debt. Each design decision that leaves a small interpretive barrier in place may be tolerable on its own. Together they compound across workflows, resist late remediation, and produce systems from which many people are effectively excluded even though every screen passes review. Earlier work in this research program calls this *accessibility debt* (Cavalcanti 2026a). Like technical debt, it is cheapest to avoid at design time, it grows quietly, and it is eventually paid, usually by the users least able to afford it and by the support teams who answer their calls.

5.5 Rethinking Accessibility Through Cognitive Distance

The accessibility movement has made remarkable progress against sensory and physical barriers. The next step is to extend its focus to conceptual barriers, and cognitive distance offers a framework for doing so. Beyond asking whether people can reach interface elements, designers must ask whether people can build meaningful mental models of the system. Does its structure reflect how people think about their goals? Do its labels match the language people use? Do its workflows follow the logic of the real-world task, and does it do the work it reasonably can instead of handing that work to the user? When these questions guide design, accessibility expands beyond technical compliance and becomes a matter of conceptual alignment between systems and the people who rely on them. Table 5.1 shows how this lens complements usability and accessibility rather than replacing them.

	Usability	Accessibility	Cognitive distance
Primary question	Is the task easy to perform?	Can diverse users access and operate the system?	Does the system's logic match how users think?
Typical measures	Time on task, completion rate, error count	Conformance checks, assistive technology compatibility	Prediction accuracy, explanation quality, confidence, hesitation
Blind spot	Efficient performance can hide shallow understanding	Technical compliance can hide conceptual confusion	Effort and access problems if measured alone
Design implication	Reduce friction in actions	Remove exclusion barriers	Align structure, language and feedback with human reasoning

Table 5.1. Usability, accessibility and cognitive distance as complementary design lenses.



CHAPTER 6

Measuring the Invisible: Detecting Cognitive Distance in Real Systems

Cognitive distance is easy to recognize once someone points it out. People hesitate, second-guess their actions, and complete tasks without being able to explain how the system works. Measuring it is harder (Hornbæk 2006; Paas et al. 2003). Traditional usability metrics focus on observable outcomes such as task completion, time on task and error frequency. They capture performance but not how people conceptualize the system. Cognitive distance sits in the gap between visible behavior and internal understanding, and detecting that gap requires different research strategies.

6.1 Why Traditional Usability Metrics Fall Short

Task success hides conceptual confusion

In most usability tests, participants complete set tasks while researchers observe, and if a participant reaches the correct outcome the attempt is recorded as a success. Success does not imply comprehension. A participant may succeed by exploring until they stumble on the right option, or by following visual cues without knowing why those cues lead to the result. When the interface later changes slightly, the same person may struggle again. The test recorded success, but the conceptual model never stabilized. The pattern is common in complex enterprise systems, where people memorize procedures rather than understand workflows.

Efficiency metrics capture speed, not alignment

Faster completion suggests efficient interaction, but speed alone says nothing about how people interpret the system. Someone may be quick because they have memorized the steps, an interaction closer to muscle memory than to understanding. When the system changes or the scenario shifts slightly, the learned sequence fails and they must reconstruct their strategy. Apparent efficiency can hide considerable cognitive distance.

Error rates reveal symptoms, not causes

Errors are rightly read as signals that an interface needs work, but error counts do not say why the mistakes happened. A person may have misread a term, misunderstood the structure of the workflow, or held a wrong model of how the system processes information. All of these produce the same observable error. The cause lies in the conceptual gap between the person's reasoning and the system's organization, and without examining that gap directly, teams fix surface issues and leave the real problem in place.

6.2 Observing Mental Models**Think-aloud protocols**

One of the most useful techniques is to ask participants to say what they are thinking while they use a system (Ericsson and Simon 1993). In a think-aloud session, people describe what they expect to

happen before acting, explain why they choose particular options, and voice their assumptions about the system's behavior. These verbalizations expose the mental model guiding the interaction, and moments of uncertainty or surprise show researchers exactly where cognitive distance appears. Think-aloud studies are long established in usability research and cognitive psychology, and they supply qualitative evidence that quantitative metrics cannot.

Concept mapping

Another approach asks participants to describe a system's structure after using it, for example by drawing how they believe it is organized. These diagrams often differ strikingly from the system's actual architecture. People group features by the goals they associate with them and describe relationships that do not exist in the design. Comparing the drawings with the real structure makes cognitive distance directly visible: large discrepancies indicate that the system's organization is hard to infer.

Interviews that probe understanding

Post-task interviews offer a third window. After a scenario, participants can be asked questions such as: How does this system decide what to show you? Why do you think this step comes before the next one? What would happen if you did these steps in a different order? Such questions push people to articulate their models. When they struggle to explain the system's behavior, or give conflicting explanations, researchers have evidence that alignment is weak, evidence that rarely surfaces in traditional metrics.

6.3 Behavioral Signals of Cognitive Distance

Hesitation during navigation

One of the earliest observable signs is hesitation. People move the cursor across several options without choosing one, pause to reread labels, or scan the screen repeatedly. They are trying to map their intention onto the structures the interface offers, and when that mapping is unclear, hesitation grows. Watching for these moments shows where alignment breaks down.

Repeated backtracking

Frequent backtracking is another signal. People move forward through several steps, then return to earlier screens to reconsider their choices. They are testing hypotheses about how the system works: trying a path, observing the outcome, revising their understanding. Some exploration is a natural part of learning, but excessive backtracking means the system's structure is hard to infer and people must keep revising their model as they go.

Reliance on external memory

When distance becomes large, people compensate with external aids. They write down instructions, keep reference notes and build personal checklists for routine tasks. This is common in enterprise environments, where employees maintain private documentation for actions they perform every week. Those notes become an external mental model. The system never became conceptually transparent, so its users built their own interpretation outside it.

6.4 Quantifying Conceptual Alignment

Mental model similarity analysis

Researchers have developed ways to compare users' mental models with system structures. One approach asks participants to sort interface elements into categories that make sense to them and compares their groupings with the interface's actual organization. High similarity suggests people perceive the system as it was designed; low similarity suggests substantial distance. The method turns conceptual alignment into data that can be compared across designs.

Cognitive walkthroughs

The cognitive walkthrough offers a structured way to evaluate how well an interface supports learning and first use, and teams can apply it during design without recruiting participants (Polson et al. 1992). Evaluators step through each action from the perspective of a new user and ask: Will the user know what to do at this point? Will they understand why the action is needed? Will they recognize that the

outcome matches their goal? Examining each step this way reveals the moments where alignment is likely to fail.

Longitudinal observation

Some forms of distance only appear over time. A system may seem easy in initial testing and reveal conceptual weaknesses as people meet new scenarios. Longitudinal studies follow people over extended periods to see whether they develop stable mental models or keep relying on procedural shortcuts. If people repeatedly struggle with new tasks in a system they already use, cognitive distance is likely present. Long-term observation is an important complement to short usability tests.

6.5 The Cognitive Distance Index

The techniques above reveal cognitive distance, but they do not put a number on it. Teams that want to compare two designs, track a system across releases, or report to people who were not in the research sessions need a measure. The *Cognitive Distance Index* (CDI) is one.

Cognitive distance cannot be observed directly. It has to be inferred from indicators that together reflect the gap between system logic and user understanding, and no single indicator is enough. Self-reported understanding is the most direct, but people do not always know what they do not know, and a successful task can inflate confidence when comprehension is shallow. Prediction and comprehension questions are less vulnerable to that bias but only test what they ask about. Open explanations capture the actual content of understanding but must be scored by hand. The CDI combines four dimensions, shown in Table 6.1, so that no single measure carries the construct alone.

Two details of the instrument matter more than they first appear. The prediction questions are asked *before* the result is shown. A person who predicts correctly beforehand and explains correctly afterwards holds a causal model of the system; a person who answers correctly only after seeing the result may simply be repeating what the screen said. And the comprehension questions test causal rather than factual knowledge. Knowing that an appointment is on Thursday at 14:30 is recall. Knowing why the system booked a general consultation

Dimension	Question it answers	Typical instrument
Perceived understanding	Can the user explain, in plain words, what the system just did?	Agreement items after the task, plus an open explanation (“In your own words, what just happened?”) scored against a rubric.
Predictive accuracy	Can the user anticipate the outcome before it is shown?	Multiple-choice prediction questions asked after the user confirms but before the result appears.
Interaction confidence	Does the user feel certain about what happened and what to do next?	Confidence ratings on each prediction, plus an “Is the task done?” item (certain / think so / not sure).
Conceptual transparency	Can the user identify <i>why</i> the outcome occurred, not only what it was?	Comprehension questions that test the rule behind the outcome, plus the causal content of the open explanation.

Table 6.1. The four dimensions of the Cognitive Distance Index.

instead of the specialist the user asked for requires understanding the rule that governs its behavior.

Computing the score. Each dimension is converted to a distance score from 0 to 100, where 0 means complete alignment and 100 means maximal misalignment. Predictive accuracy, for example, becomes 100 minus the percentage of correct predictions. The CDI for a participant or group is the mean of the available dimension scores, and it is not reported on fewer than two dimensions, because partial composites are hard to compare. Table 6.2 gives the interpretation bands.

The bands are a working proposal, not a validated standard. They are anchored to the logical extremes of the construct and to pilot observations, and the equal weighting of the four dimensions has not yet been tested against alternatives. The CDI is also baseline-relative: it is most informative when used to compare versions of the same system, or the same system across user groups, rather than to rank

CDI	Band	What it looks like	Response
0–25	Strong alignment	Users explain outcomes correctly and predict future behavior reliably.	Monitor.
26–45	Early warning	Scattered confusion; some users cannot account for outcomes.	Review terminology and feedback wording; run targeted tests.
46–65	Needs attention	Misunderstanding recurs across users.	Redesign result screens, explanatory copy and causal labels.
66–100	Serious misalignment	Users routinely succeed operationally while failing interpretively.	Restructure how outcomes are communicated and how information is organized.

Table 6.2. CDI interpretation bands.

unrelated products. Teams should treat a CDI score the way a clinician treats a single blood test: as a reason to look closer, not as a diagnosis.

The four dimensions can also diverge, and when they do the pattern is diagnostic. The most concerning combination is what this book calls the *calibration gap*: moderate or high confidence alongside prediction accuracy near chance. People who know they do not understand will ask for help. People who are wrong but confident cannot correct themselves, because nothing tells them anything needs correcting (Kruger and Dunning 1999). In a license renewal or a medical booking, that is the person who leaves believing the task is finished when it is not.

6.6 Testing the Idea: The Everyday Services Study

A construct earns its place by making predictions that could turn out to be wrong. The Everyday Services Study is designed to test the central claim of this book under controlled conditions (Cavalcanti 2026c).

The study isolates one variable: the language a system uses to report what it has done. Participants complete three everyday tasks on working prototypes: a bank transfer, a medical appointment booking

and a driver’s license renewal. Half see Version A, written in plain language. Half see Version B, written in the institutional register common in deployed public-service, banking and healthcare systems (Li and Liu 2025). Everything else is identical: task structure, number of clicks, form fields and actual outcomes. Only the language changes, and it changes most on the result screens, where the system has its last chance to pass on what it knows (Table 6.3).

Service	Version A: plain language	Version B: institutional language
Banking	Done. Mary receives \$200.00 today by 18:00. The money has left your account. You do not need to do anything else.	Operation processed successfully. Authentication: 8F2-9921-B. Ref: TX-29381.
Medical	Booked: General consultation, Thursday 14:30, Central Clinic, Dr. Lima. Bring your ID. At this visit, ask the doctor for the skin specialist referral.	Booking registered. Protocol 5512A. Type: Triage, General Practice. Status: Pending allocation.
License	Payment received. You are done. We review your renewal within 5 days, then mail your new license, which arrives in about 10 days.	Request filed under protocol 2026-88412. Status: Stage 2 of 6, Administrative instance.

Table 6.3. Result screens in the Everyday Services Study. Task structure and outcome are identical; only the language differs.

The Version B screens are not caricatures. Each is a pattern found in real systems, and each leaves unanswered questions that matter to the person reading it. The license screen does not say whether the fee was paid, whether the renewal was approved, or whether anything else is required. Version A answers all three.

The study runs online, unsupervised, in English, Spanish and Brazilian Portuguese, because cognitive distance matters most precisely when people use systems alone. Every participant completes all three services in one of six counterbalanced orders. Measurement is built into the flow using the CDI instruments described above: prediction questions after the user confirms but before the result appears, comprehension questions immediately after, and a short

questionnaire ending with an open explanation that is scored without knowing which version the participant saw.

The study makes five directional predictions. Compared with Version B, participants who see Version A should report higher understanding, give better open explanations, predict outcomes more accurately and feel more confident, and their CDI should be lower in all three services. Version B participants are expected to predict at or near the 25 percent chance level. Just as important are three checks that must *not* differ between versions: completion rates, self-reported ease and effort, and task duration. If Version B were simply harder, the study would be measuring load. The design only tests distance if effort stays equal while understanding diverges.

What the first pilot showed. An earlier iteration of the study ran as a small qualitative pilot. Its results are not evidence in a statistical sense, but they illustrate the pattern this book describes. Participants using the institutional version completed tasks at normal speed and reported no unusual difficulty. Their open explanations told a different story. Many described what had happened in the system's own words without interpreting them: "it said the protocol was filed," "it showed stage 2 of 6." These are explanations that quote the system back to itself. They are what high cognitive distance sounds like.

The pilot also changed the study. The institutional screens were revised so that their ambiguity reflects real systems rather than exaggerated ones, task instructions were standardized so that any advantage for Version A comes from the result screens alone, and the medical comprehension questions were rewritten after almost everyone answered them correctly in both versions.

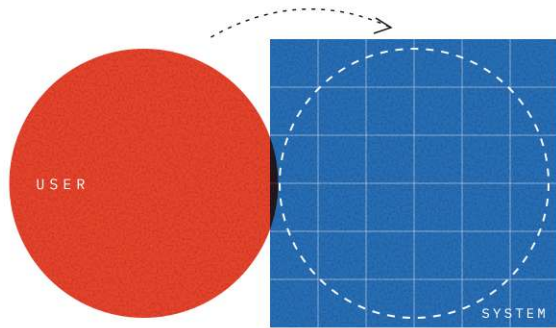
Limits. The study uses prototypes rather than live systems, measures distance at one moment rather than over months of use, and relies on hand-scored explanations that will need a second independent scorer. Whichever way its results fall, it will answer a narrow question well rather than a broad question loosely. That is the right order in which to build evidence for a new idea.

6.7 Diagnosing Cognitive Distance in Practice

Design teams do not always have the resources for a controlled study, but many signs of cognitive distance can be found with simple practices: watching for hesitation during usability tests, holding short interviews about how people believe the system works, and asking participants to explain a workflow in their own words. Adding even two or three prediction questions to an existing test, asked before the result appears, brings a lightweight version of the CDI into everyday practice. These small steps reveal conceptual gaps that traditional metrics overlook, and once teams recognize the patterns they can start addressing the causes. Table 6.4 is a practical checklist for spotting conceptual friction early.

Signal	What it looks like	Likely cause	How to measure
Hesitation	Long pauses before obvious actions	Unclear mapping between labels and intent	Time to first action; think-aloud transcripts
Backtracking	Frequent returns to earlier steps	Navigation does not match the task model	Path analysis; loop detection
Procedural dependence	Users repeat sequences but cannot explain them	Completion without understanding	Explanation request after the task
Support reliance	Help desk calls for routine tasks	The system cannot explain itself	Support ticket categories and trends
Terminology mismatch	Users rephrase interface language to make sense of it	Internal vocabulary in user-facing text	Interview coding; language comparison
Calibration gap	Confident users who predict wrongly	Outcomes reported as states, not consequences	Prediction accuracy against confidence

Table 6.4. Diagnostic signals of conceptual friction and how to detect them.



CHAPTER 7

Design Assumptions and the Architecture of Misunderstanding

Design teams rarely set out to create confusing systems. Most designers genuinely want their products to feel clear and supportive, and yet misunderstanding appears again and again. Interfaces launch after careful design reviews, usability testing and accessibility checks, and still people misinterpret workflows, misread terminology, or rely on memorized steps instead of understanding.

These signs suggest that the problem does not come from isolated mistakes. It comes from deeper assumptions embedded in the way systems are conceived and built. Those assumptions shape how teams interpret user behavior, how they structure information and how they define success, and over time they create conditions in which

cognitive distance becomes almost inevitable. This chapter examines four of them.

7.1 The Assumption of Rational Navigation

Designers assume users think hierarchically

Many systems are organized as hierarchies. Information is grouped into categories, categories contain subcategories, and users move downward until they find what they need. From a design perspective this looks logical and orderly. Hierarchies are easy to diagram and provide clear boundaries between sections. But people do not always reason about tasks hierarchically. They think in terms of goals and situations. Someone visiting a government website may not know which department handles a service; they simply want to get something done. If the system forces them to interpret administrative categories before they can proceed, they must adapt to the system's conceptual structure, and that adaptation is cognitive distance.

Real-world goals rarely map cleanly to interface categories

A single goal often cuts across several organizational categories. A small business owner updating licensing information on a government platform may be dealing at once with regulatory compliance, financial reporting and local administrative requirements. Internally, these belong to separate departments, so the system organizes information along those lines. For the owner, it is one task. When a system splits that task across several categories, people must reconstruct the path through trial and interpretation, and the interface ends up reflecting institutional logic rather than human intention.

Hierarchies create fragile navigation

Hierarchical navigation works best when people can predict where information will be. When categories overlap, or terminology differs from everyday language, they hesitate. They choose one category only to discover that the option they need is somewhere else, and repeated misclassification erodes their confidence in the structure. Eventually they stop reasoning about the hierarchy and start exploring at random.

At that point the navigation no longer works as a conceptual guide; it has become a collection of possible paths.

7.2 The Assumption of Shared Vocabulary

Designers often speak a different language than users

Language is one of the most powerful tools in interface design. Labels guide navigation, instructions explain actions and feedback messages clarify responses. Yet the vocabulary in these elements often reflects internal organizational language rather than everyday speech. Engineers and product teams become deeply familiar with technical terms during development, the terms start to feel natural, and when they appear in the interface the team assumes they are self-explanatory. People meeting the system for the first time may read them very differently, and even small differences in wording reshape how they understand what the system does.

Terminology influences mental models

Words do more than name functions; they shape how people conceptualize what the system is doing. If a financial application calls a feature “account reconciliation”, users may imagine a complex accounting process. If the same function is labeled “check your balance records”, the mental model becomes far more approachable. Both labels refer to the same operation, but they signal different conceptual structures. When terminology reflects internal processes rather than user goals, people must translate the system’s language into their own, and the distance grows.

Misinterpretation accumulates over time

A single unfamiliar term rarely causes serious problems, but systems never rely on one label alone. Interfaces contain dozens or hundreds of textual elements, and if several require interpretation, the effort adds up. People start second-guessing their understanding and hesitate before choosing options, because they cannot confidently predict what each term means. Language becomes one more layer of translation between human reasoning and system logic.

7.3 The Assumption of Linear Workflows

Many systems assume tasks occur in fixed sequences

Teams often structure workflows as a series of sequential steps: the user starts on one screen, completes an action and moves to the next. The structure simplifies development and testing, since each stage can be evaluated independently and progress is easy to track. Real tasks rarely unfold that neatly. People gather information, reconsider decisions and adjust their plans as new information appears. When a system enforces a rigid sequence, they cannot move between steps to refine their understanding, and the interface dictates a process that may not match the way they reason.

Exploration becomes difficult

Linear workflows also reduce opportunities for exploration. If people cannot preview later steps or revisit earlier ones without starting over, they see only one piece of the process at a time and struggle to build a model of the whole. They may complete the workflow without understanding its structure, following instructions step by step without integrating them into a coherent picture.

Rigid workflows amplify uncertainty

When people cannot anticipate what comes next, uncertainty increases and each step becomes a moment of interpretation. If the process involves important decisions or irreversible actions, that uncertainty can turn into anxiety, and people proceed cautiously or abandon the process altogether. The system functions, yet the relationship between person and interface remains fragile.

7.4 The Assumption of Static Understanding

Designers assume users eventually learn the system

A common assumption is that early confusion disappears with experience: people will learn where features are and how workflows operate. There is some truth in this, since repeated exposure does improve performance. But familiarity does not always produce understanding. People can memorize procedures without forming a coherent model,

knowing which buttons to press but not why. If the interface changes even slightly, their strategy collapses, because what they learned was procedural rather than conceptual.

Organizational turnover reveals hidden problems

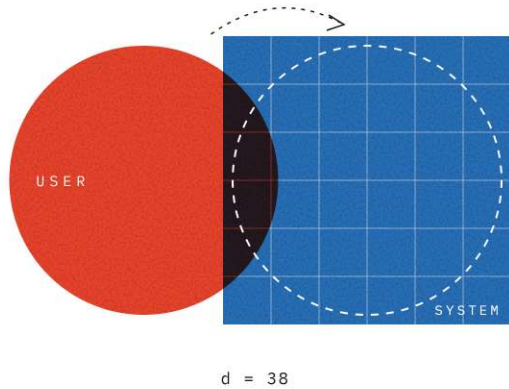
The effect is especially visible in large organizations. Long-serving employees navigate internal systems efficiently, while new employees struggle. Training sessions tend to teach specific procedures rather than the conceptual structure of the system, so each new group must memorize the same steps. The system never becomes understandable; it simply accumulates layers of learned routine.

Systems that require training signal cognitive distance

When a system needs extensive training just for routine tasks, that is often a sign that its conceptual model differs significantly from how people reason. Training can compensate for the difference, but it does not remove it. It teaches people to operate within the system's framework, and the underlying distance remains, quietly shaping every interaction.

7.5 Recognizing the Pattern

Each of these assumptions feels reasonable during design. Hierarchies look organized, technical terminology feels precise, linear workflows simplify implementation and training seems a practical fix. None of them necessarily causes a serious problem on its own. Together, though, they gradually shift a system away from the structure of human reasoning, until the interface reflects internal logic rather than user understanding. People adapt by memorizing procedures, relying on external instructions and learning to navigate without understanding, and that adaptation hides the distance from the very metrics meant to find it. The system appears functional while the relationship between person and machine remains fragile.



CHAPTER 8

Designing for Alignment: Reducing Cognitive Distance

Once designers begin noticing cognitive distance, it becomes difficult to ignore. Familiar usability issues appear in a different light: confusing menus, fragmented workflows and opaque terminology stop looking like isolated flaws and start looking like symptoms of a deeper mismatch between human reasoning and system organization. Reducing that mismatch requires more than cosmetic improvement. The question shifts from “Is this interface easy to use?” to something more fundamental: does the system work in a way people can follow?

This chapter explores practical strategies for answering that question. Table 8.1 connects common symptoms to concrete interventions.

Observed symptom	Design intervention	Expected experience
Users complete tasks but still feel unsure	Clear cause-and-effect feedback at key moments	Confidence rises; reliance on reassurance falls
Users cannot tell where to start	Navigation organized around intentions and life events	Faster orientation; a clearer path through the task
Users misread terminology and labels	Replace internal jargon with language drawn from user research	Clearer interpretation, fewer detours
Users rely on memorized click sequences	Expose process structure through previews and progress maps	Better transfer to new scenarios
Users are handed work the system could do	Absorb conversion, formatting and validation internally	Fewer workarounds and abandoned tasks
Users distrust automated outcomes	Explain limits and uncertainty in plain language	More calibrated trust

Table 8.1. From observed symptoms to alignment-oriented design moves.

8.1 Designing Around Human Goals

Start with intentions, not categories

One of the most effective ways to reduce cognitive distance is to organize systems around what people want to do rather than around internal categories. Traditional information architectures mirror the organization: departments become navigation sections and internal processes become workflows. People rarely think in those terms. They arrive with intentions: pay a bill, renew a document, find a piece of information. Designing around these intentions produces a structure that matches how they reason. Instead of asking people to find the right administrative category, the system starts from the action they want to take.

Many public service platforms have moved in this direction. Several national government portals now organize services around life events, such as starting a business, moving house or applying for education,

rather than around the agencies that deliver them. Goal-centered structures like these reduce the translation work people must do.

Map the natural sequence of real-world tasks

Once the goals are clear, the next step is to understand how people naturally approach them. Real tasks follow informal sequences shaped by context and habit. Someone planning a trip may first explore destinations, then compare prices, and only later choose dates. A travel platform that forces date selection first interrupts that reasoning. Workflows that mirror real decision patterns preserve alignment, and finding those patterns means observing people outside the interface: ethnographic research, contextual inquiry and task analysis reveal how problems are approached in everyday settings.

Show the structure of the journey

Even when a workflow follows the natural sequence, people benefit from seeing its overall shape. Progress indicators, overview pages and navigable steps show where someone is in the process. Instead of a series of isolated screens, the user sees how the stages relate. That visibility supports mental model formation, reduces uncertainty, and encourages exploration, because people are confident they can move around without losing their place.

8.2 Language That Matches Human Thinking

Use vocabulary drawn from user research

Teams often invest heavily in layout and interaction patterns while treating interface language as an afterthought, yet the words in labels, instructions and feedback messages strongly shape the models people build. The most reliable source of good terminology is observation of how people describe their own tasks. Interviews, support conversations and usability sessions reveal the language people naturally use, and interfaces that adopt it reduce translation work: people recognize familiar phrases and map them quickly onto their intentions.

Avoid system-centered phrasing

Many confusing labels come from internal descriptions of what a function does. Consider a feature labeled “data synchronization”. Technically, it is accurate. For the user, the action may simply mean “update your information”. Shifting terminology towards user-centered descriptions means the interface communicates outcomes rather than mechanisms.

Explain unfamiliar concepts gradually

Not every system can rely entirely on familiar vocabulary. Some involve specialized concepts that people must eventually learn. In these cases designers can introduce terminology progressively. Early interactions focus on outcomes and tasks, and as people become comfortable, further explanation adds depth. Good educational interfaces work this way, starting with simple interactions that illustrate the core ideas and revealing the underlying structure over time, so that stable mental models form without overwhelming newcomers.

8.3 Designing Transparent Systems**Reveal cause and effect**

One of the most powerful ways to support alignment is to make the relationship between actions and outcomes visible. When people act, the system should show how that action shaped the result. Feedback messages, visual transitions and explanatory cues all contribute. When a filter changes the visible results, for example, the interface can say which criteria produced the new view. People begin associating specific actions with specific responses, and over time those associations strengthen the model guiding their interaction.

Make system status visible

People also need to know what the system is doing. Waiting without explanation creates uncertainty: Is the system processing? Has the request failed? Did anything happen at all? Progress indicators and status updates reduce this ambiguity, reassure people that the system is working, and give them a glimpse of the sequence of operations

behind the screen. This is the first of Nielsen's heuristics for good reason (Nielsen 1994).

Clarify automated decisions

Automation adds conceptual complexity, because people struggle to understand how automatic decisions are made. Brief explanations help. Recommendation systems often say why an item appears: a streaming service notes that a suggestion is based on what you watched, and an online store says a recommendation reflects a recent purchase. Research on recommender systems has catalogued these explanation styles and the different goals they serve, from transparency to trust (Tintarev and Masthoff 2007). The explanations are simplified, but they offer conceptual cues without exposing the full technical machinery.

8.4 Internalize Explanation

The strategies in this chapter share a single underlying rule. When a system knows something the user needs in order to understand an outcome, the system, not the user, should do the work of making it understandable.

Most systems already hold a precise causal model of what they just did. A bank knows that the transfer executed instantly, that the security code was required because the recipient was new, and that the customer has nothing further to do. A licensing system knows that the fee was received and how long review takes. A clinic's booking system knows that the general consultation is a step towards a specialist referral. Cognitive distance appears when none of that reaches the screen, and the person is left with a protocol number and a status label.

Internalizing explanation means three things in practice. First, report outcomes as consequences for the person ("The money has left your account") rather than as system states ("Operation processed"). Second, state the reason when an outcome is not the one the person asked for ("We booked a general consultation because specialist visits need a referral"). Third, say explicitly whether anything else

is required, because the most dangerous misunderstanding is the person who believes they are finished when they are not.

The same rule applies to the externalized processing described in Section 5.4. Where a system can reasonably convert a file, compress an image or normalize a date format itself, it should. Where it genuinely cannot, it should explain the constraint in terms of what the user can do, and offer the means to do it inside the interface instead of sending people to search for third-party tools. In both cases the test is the same: after the interaction, can the person explain what happened and what, if anything, comes next?

8.5 Encouraging Exploration Without Confusion

Support safe experimentation

People learn systems by exploring, and designers can encourage that by making experimentation safe. Undo, reversible actions and clear confirmations reduce the perceived risk of trying something new. When people know they can recover from mistakes, they explore more, and exploration strengthens mental models by letting them observe cause and effect directly. The system becomes a learning environment rather than a rigid set of instructions.

Provide contextual guidance

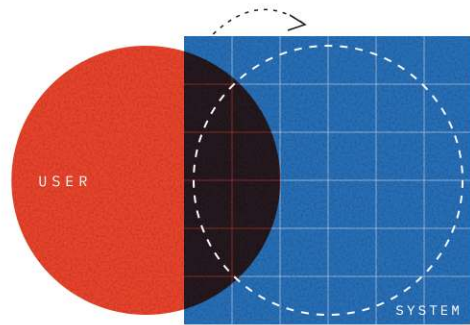
Contextual guidance offers help at the moment it becomes relevant. Instead of front-loading long instructions, the system introduces cues as people encounter new concepts. Tooltips, short explanatory messages and interactive hints bridge conceptual gaps without interrupting the flow, and as the user's model stabilizes they rely on these cues less and less.

Allow multiple paths to the same outcome

Rigid workflows force everyone through the same sequence, which simplifies development but conflicts with the different ways people approach tasks. Some prefer to search directly for a feature; others navigate structured menus. Supporting both acknowledges that people reason differently, and lets the interface adapt to the user rather than forcing the user to adapt entirely to the system.

8.6 Designing Systems That Stay Understandable

Reducing cognitive distance is not a one-time task. Systems evolve, features expand and organizational priorities shift, and without deliberate attention even well-designed interfaces drift away from human reasoning, as Chapter 2 described. Keeping a system aligned requires ongoing observation and adjustment through user feedback, usability studies, behavioral analytics and periodic measurement of understanding itself. When teams treat conceptual understanding as a central design goal, these insights become powerful tools for continuous improvement.



$d = 31$

CHAPTER 9

Artificial Intelligence and the Expanding Distance

Artificial intelligence is transforming how people interact with digital technology (Chignell et al. 1989; Zhang et al. 2025). For decades, most software followed predictable patterns: people acted and systems responded according to clearly defined rules. If you understood the structure of the interface, you could usually anticipate what would happen next.

AI changes that relationship. Instead of executing fixed procedures, many modern systems generate responses dynamically, drawing on patterns learned from large datasets and producing outputs that adapt to each situation. This flexibility lets systems do things that once seemed impossible, but it also adds a new layer of uncertainty. People often cannot predict how an AI system will behave, and the

reasoning behind its responses stays hidden. Interaction begins to feel like a conversation with an unfamiliar mind, and in that environment cognitive distance can expand rapidly.

9.1 From Deterministic to Probabilistic Systems

Traditional software follows explicit rules

For most of computing history, software was deterministic. Developers defined conditions and actions, and a given input triggered a specific instruction. Even complex systems followed logic that could be traced through code, so people who understood the rules could form fairly accurate mental models. A spreadsheet applies formulas exactly as written; a search filter shows results that match the chosen criteria. These interactions take learning, but they remain conceptually stable.

Machine learning systems operate through statistical patterns

Systems built on machine learning work differently. Rather than relying solely on explicit rules, they learn patterns from large sets of examples, and their outputs reflect statistical relationships rather than fixed instructions. Language models, image recognition and recommendation engines all work this way. The neural network methods behind them, developed over decades by researchers including Geoffrey Hinton, Yoshua Bengio and Yann LeCun, are remarkably powerful (LeCun et al. 2015). They are also hard to interpret. An output may depend on millions or billions of learned parameters, and from an everyday user's perspective the system becomes a black box.

Predictability becomes harder to maintain

Because AI systems generate responses dynamically, their outcomes are not always consistent. Similar inputs can produce different results. A recommendation system's suggestions shift as it learns from new data. A language model may interpret the same question differently depending on context. These variations are natural consequences of probabilistic systems, but they disrupt the mental models people try to build. People expect digital systems to behave consistently, and when behavior shifts unexpectedly they lose confidence in their

understanding. Table 9.1 summarizes why probabilistic behavior feels less conceptually stable.

	Deterministic systems	Probabilistic systems
Response generation	Executes rules defined in advance	Predicts outputs from patterns learned from data
Repeatability	The same input usually gives the same output	Similar inputs can give different outputs
Visibility of reasons	Logic can be traced through procedural steps	Reasoning is distributed across many parameters
User strategy	Learn the rules and apply them	Keep testing assumptions and verifying results
Basis of trust	Consistency and predictability	Transparency, explanation and safeguards

Table 9.1. Deterministic and probabilistic systems and what they mean for cognitive distance.

9.2 Anthropomorphism and the Illusion of Understanding

Humans attribute intention to intelligent behavior

When systems produce language or make recommendations that seem thoughtful, people tend to read that behavior as evidence of understanding. People readily treat computers as social actors, applying to them the expectations they hold of other people (Reeves and Nass 1996). The tendency is old: in the 1960s, users of Joseph Weizenbaum’s simple ELIZA program confided in it as if it understood them, to its creator’s dismay (Weizenbaum 1966). When a conversational system answers fluently today, people may assume it grasps the meaning behind its words, when it is producing statistically likely text. The appearance of understanding emerges from modeling, not from reasoning in the human sense.

Conversational interfaces strengthen the illusion

Modern AI systems often communicate through conversation. People ask questions, the system answers, and the exchange unfolds in a

natural linguistic format. Conversational assistants now support everything from writing to programming, and the format makes them accessible and engaging. It also strengthens the impression that the system understands the conversation as a person would, and people may take confident language as evidence of reliability. When the system then produces incorrect or inconsistent information, the mismatch between expectation and reality becomes visible. Cognitive distance appears in the gap between perceived intelligence and actual mechanism, illustrated in Figure 9.1.

Misinterpretation creates fragile trust

Trust in AI systems often forms quickly. Early interactions produce useful responses, and people begin relying on the system for increasingly complex tasks. That trust may rest on an incomplete model. People may not know the system's limitations, or may assume it verifies information or applies logic when generating responses. Research on automation has long warned that trust which outruns a system's actual reliability leads to misuse, and trust that falls below it leads to disuse (Lee and See 2004). When the system produces an inaccurate result, miscalibrated trust can collapse suddenly. The person realizes their understanding was incomplete, and that moment of disillusionment reveals the cognitive distance that was present all along.

9.3 Opacity and the Challenge of Explanation

Many AI systems cannot easily explain their decisions

A defining challenge of modern AI is the difficulty of explaining how particular outputs are produced. Deep neural networks work through layered mathematical transformations that resist direct interpretation, and even the engineers who build them cannot always identify the factors behind a specific prediction. That opacity is a design problem. People often want to know why a system produced a particular result, and if it cannot offer a meaningful explanation they must accept the outcome without understanding it.

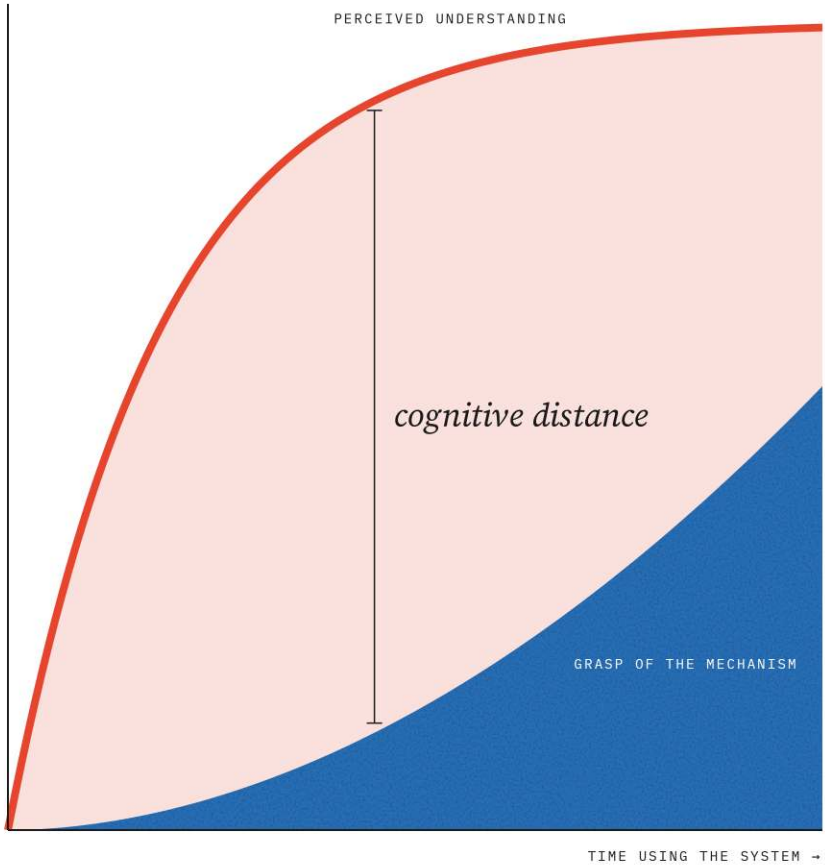


Figure 9.1. The trust gap in conversational AI. Perceived understanding (red line) rises quickly, while grasp of the mechanism (blue) grows slowly. The shaded space between them is cognitive distance. The curves are illustrative, not measured.

Explainable AI as an emerging field

Researchers have begun developing methods to make AI behavior more interpretable, a field usually called explainable AI (Doshi-Velez and Kim 2017). The goal is not to expose every internal parameter, but to provide explanations that help people interpret what the system did. Image classifiers can highlight the regions of a picture that influenced a decision (Selvaraju et al. 2017); other methods identify which input features most affected a particular prediction (Ribeiro et al. 2016); recommendation systems can indicate which of a user's behaviors led to a suggestion. These explanations provide conceptual cues that help people build models of how the system works, and work on interpretability increasingly measures whether they actually improve human understanding rather than assuming they do (Poursabzi-Sangdeh et al. 2021). Figure 9.2 outlines a practical progression from raw output to an explanation people can act on.

- 1. Outcome statement.** State clearly what the system produced or decided.
- 2. Primary reason.** Name the strongest contributing factor in user-facing terms.
- 3. Confidence qualifier.** Indicate certainty, limits and known conditions of uncertainty.
- 4. Verification guidance.** Suggest what the person should check before acting.
- 5. Next step.** Offer a concrete way to refine the input or request an alternative.

Figure 9.2. The explanation ladder for AI interactions, from disclosing an output to helping the person act on it.

Explanations must match human reasoning

Providing explanations does not guarantee understanding. A technical description may accurately reflect a model's workings and still be impossible for most people to interpret. Effective explanations

connect system behavior to concepts people recognize. A recommendation system that says a suggestion appears because you watched a particular film, or bought a similar product, reflects a pattern anyone can follow. It translates complex statistical relationships into familiar reasoning, and that translation is what reduces cognitive distance. Studies of end users have found that explanations which are sound and complete improve the accuracy of people's mental models, while explanations that oversimplify can leave them with confident but wrong ideas (Kulesza et al. 2013).

9.4 Designing AI Systems That Humans Can Understand

Communicate system capabilities clearly

One of the simplest ways to reduce cognitive distance in AI systems is to state openly what they can and cannot do. Guidelines for human–AI interaction begin exactly here: make clear what the system can do, and how well it can do it (Amershi et al. 2019). An AI writing assistant might explain that it generates text from patterns in its training data and can produce inaccurate information. That transparency encourages people to treat the system as a helpful tool rather than an infallible authority, and clear expectations support more stable mental models.

Provide visible reasoning cues

Interfaces can also reveal aspects of system reasoning through small design choices. Search engines show suggested queries related to the user's input, hinting at how the request was interpreted. Recommendation platforms label rows “because you watched” or “similar to items you viewed”. These cues expose the logic guiding the system's behavior, and people gradually develop a sense of how it organizes information and generates results, even when the full technical details stay hidden.

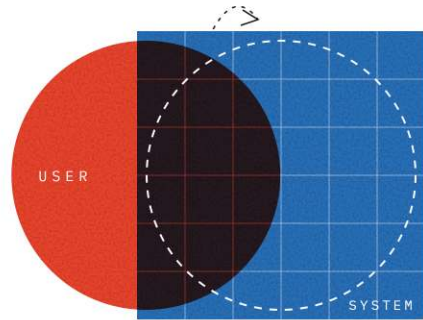
Encourage critical engagement

AI systems should not encourage passive acceptance of their outputs. Designers can promote critical engagement by letting people question results, refine their inputs and explore alternatives. Conversational

systems can invite clarification or additional context; recommendation systems can let people adjust the signals they are based on. These opportunities make the interaction collaborative rather than opaque.

9.5 The Future of Human–AI Interaction

Artificial intelligence brings remarkable capabilities into digital systems, and it challenges long-standing assumptions about predictability and understanding. When behavior emerges from statistical patterns rather than explicit rules, people have to construct new kinds of mental models. Designers therefore carry an important responsibility: to keep AI-powered systems interpretable enough for people to reason about. Without that effort, cognitive distance will keep expanding, and people will rely on systems whose logic they cannot grasp. The conceptual transparency dimension of the Cognitive Distance Index (Section 6.5) offers one way to measure whether explanation is actually working. Reducing that distance may become one of the defining challenges of Human-Computer Interaction. The next chapter widens the lens to the institutions whose decisions increasingly reach people through such systems.



d = 23

CHAPTER 10

Systems, Institutions, and the Distance Between People and Digital Power

Digital systems no longer operate only as tools; they increasingly function as institutional infrastructure. Government services, financial systems, healthcare platforms, educational portals, transport networks and public information services all run on complex digital architectures, and for many citizens these systems are now the main way institutions reach them. Applying for benefits, submitting tax information, viewing medical records and managing bank accounts all happen through an interface. When people use such a platform, they are often dealing with the institution itself.

The design of these systems therefore shapes how people experience institutional power. Aligned with human reasoning, they can empower people and widen access to services. When cognitive dis-

tance grows, the relationship between people and institutions changes. Processes become opaque, decisions feel distant and unexplained, and what used to be administrative procedure starts to resemble an invisible bureaucracy embedded in code.

10.1 Digital Bureaucracy

Institutions encode procedures into software

Traditional bureaucracies relied on people. Citizens submitted forms, officials reviewed them, and decisions followed established procedures. In digital environments, many of those procedures are written directly into software. Applications evaluate eligibility automatically, platforms process requests by predefined rules, and data moves between institutions without anyone touching it. Public administration scholars have described this as a shift from *street-level* to *system-level* bureaucracy, in which discretion once exercised by individual officials is increasingly built into the design of information systems (Bovens and Zouridis 2002).

The shift can bring large gains in efficiency. Processing times fall, administrative costs drop and services are available at any hour. But it also changes how citizens experience the process. Instead of a person who can explain a decision, they meet an interface that displays an outcome. Figure 10.1 shows where institutional logic becomes hard for citizens to interpret.

Procedural logic becomes invisible

In traditional settings, administrative procedures were often visible through documentation or explanation; a clerk could walk someone through the steps a service required. In digital systems, those procedures are hidden inside the software. People see the input fields and the result, but not the reasoning in between. When an application is rejected or a request fails, the system may offer only a minimal explanation, while the logic behind the decision sits in code beyond the user's reach. That invisibility increases cognitive distance, and citizens must trust processes they cannot interpret.

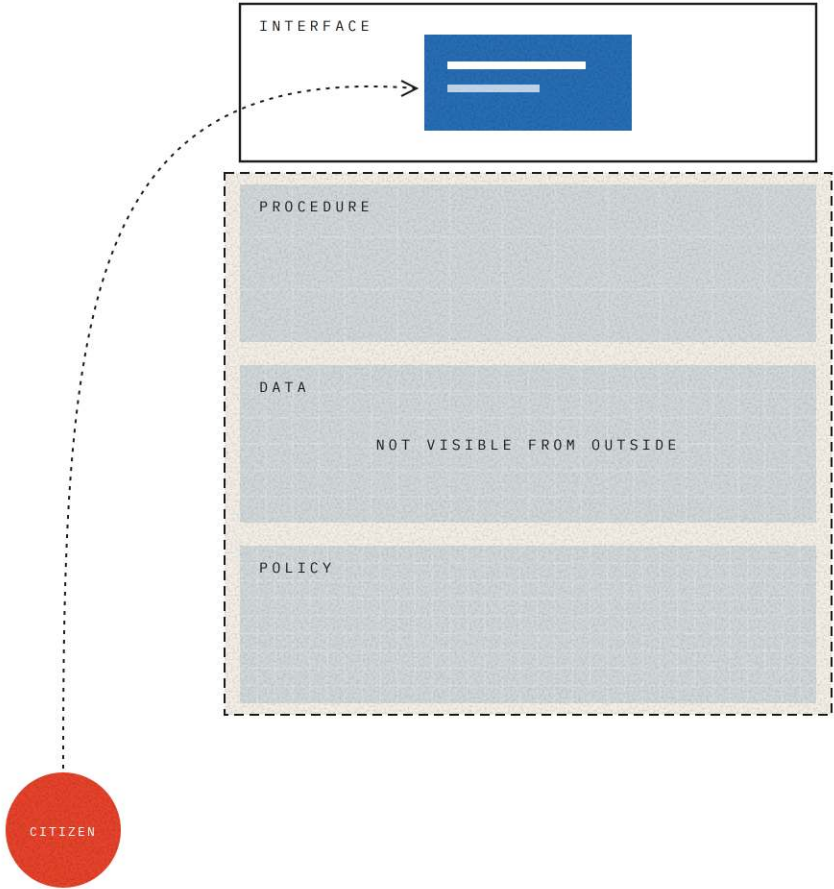


Figure 10.1. The institutional decision stack. The citizen sees the outcome on the interface; the procedure, data and policy behind it are not visible from outside.

Algorithmic governance

As institutional systems become more sophisticated, organizations increasingly rely on algorithmic decision-making to assess financial risk, triage patients and allocate public resources. Scholars of digital governance have shown how such systems can shape social outcomes in powerful and often unequal ways. Cathy O’Neil documented how opaque, large-scale scoring models can reinforce disadvantage while escaping scrutiny (O’Neil 2016), and Virginia Eubanks traced how automated eligibility systems in public services fall hardest on poor and working-class people (Eubanks 2018). When algorithms determine access to services or opportunities, understanding how they work becomes critically important. Yet many remain hard for ordinary people to interpret, and cognitive distance becomes a civic concern, not only a usability one.

10.2 The Experience of Institutional Opacity

Citizens encounter systems, not organizations

People once dealt with institutional representatives directly. Visiting a government office, they could ask questions, request clarification and work through procedural complexity in conversation. Digital platforms replace much of that with structured interfaces: forms, uploads and automated workflows. Efficiency increases, but opportunities for clarification shrink. When something goes wrong, people may not know where to turn, because they are dealing with a system rather than a person.

Lack of explanation erodes trust

Institutional trust depends heavily on perceived transparency. People tend to trust systems when they believe decisions follow understandable rules, and grow suspicious when outcomes seem arbitrary. Digital systems can create that impression without meaning to. A person submits an application, the system processes it, and a result appears with no explanation. Even if the process was fair, the absence of visible reasoning creates doubt, and people start to wonder whether hidden factors decided the outcome. Cognitive distance therefore affects institutional legitimacy: understanding how systems make

decisions becomes part of public trust. Table 10.1 summarizes how opacity turns into concrete consequences.

Symptom	Likely hidden cause	Consequence for users	Design remedy
Unexplained rejection	Eligibility logic encoded in opaque rules or models	Perceived unfairness; loss of confidence in the institution	Show the key criteria and which ones were not met
Status uncertainty	State changes between submission and outcome are invisible	Repeated contact, anxiety, duplicate submissions	Show the stage, expected timeline and anything still required
Inconsistent outcomes	Cross-system data and changing rules are not communicated	Users infer arbitrariness and turn to workarounds	Show where data came from and what changed the outcome
Dependence on intermediaries	Language and process reflect institutional internals	Unequal access for people without expert help	Rewrite flows around citizen goals in plain language

Table 10.1. How hidden institutional logic becomes trust breakdowns and practical barriers.

The rise of digital intermediaries

When systems become hard to interpret, a new kind of intermediary emerges: consultants, support agents and informal experts who help others through complex digital processes. Tax professionals interpret government filing systems, financial advisers navigate investment platforms, and technical specialists assist with enterprise software. These intermediaries play valuable roles, but their existence often reflects the conceptual complexity of the systems themselves. When digital infrastructure requires specialized interpreters, cognitive distance has grown large, and the system no longer communicates its logic directly to the people it serves.

10.3 Power, Knowledge, and System Design

Understanding systems becomes a form of power

Where digital infrastructure dominates, knowing how systems work becomes a valuable resource. People who understand a platform's structure gain practical advantages: they move through processes efficiently, know which inputs produce which outcomes, and read system feedback correctly. For others, the system stays opaque. They may complete the same processes, but with more uncertainty and effort, and that difference can affect access to opportunities, services and resources. Cognitive distance therefore intersects with broader questions of digital inequality.

Data-driven systems amplify complexity

Many modern institutional systems depend on large-scale data integration, with information flowing between government agencies, banks, healthcare providers and digital platforms. These connections let institutions coordinate services more effectively, but they also make the overall environment more complex. A decision shown on one screen may depend on data processed by several systems behind it, and the person looking at the screen sees a single outcome with no view of the network that produced it. That layered architecture deepens cognitive distance.

Transparency as a design responsibility

Addressing these problems takes more than technical efficiency. Designers and institutions must consider how digital infrastructure communicates its logic to the people who depend on it. Transparency does not mean exposing every line of code, but systems should give meaningful explanations for important outcomes, and people should be able to understand why a process behaves as it does. Designers therefore hold a kind of institutional responsibility. The interface is where complex infrastructure meets human understanding, and reducing cognitive distance at that boundary supports not only usability but democratic accountability.

10.4 Toward Human-Centered Digital Institutions

Designing systems citizens can reason about

As institutions continue to digitize their services, designers face a crucial challenge: keeping systems interpretable enough for ordinary citizens to understand. That means communicating processes clearly, giving meaningful feedback and explaining decisions accessibly. An application system might show which eligibility criteria were evaluated and how each affected the decision, so that people see the reasoning as well as the result. The result screens of the Everyday Services Study (Table 6.3) show how small that change can be on the page and how large its effect on understanding is expected to be.

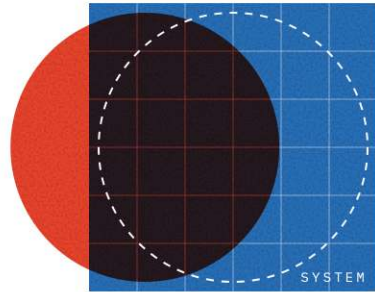
Participatory design in public systems

Another promising approach is to involve citizens directly in designing the services they use. Participatory design invites people to describe their goals, interpret prototypes and help identify conceptual barriers alongside designers. The tradition has roots in Scandinavian workplace projects of the 1970s and 1980s, associated with researchers such as Kristen Nygaard and Pelle Ehn (Ehn 1988; Schuler and Namioka 1993). Involving people in design decisions reduces the risk of embedding organizational assumptions that diverge from public reasoning, and helps digital infrastructure reflect the needs and mental models of the communities it serves.

Digital systems as civic interfaces

Increasingly, the interface through which citizens meet institutions is not a building or a person but a digital system. These systems function as civic interfaces: they shape how people understand public processes and how institutions express their authority. When they support conceptual clarity, citizens can engage with institutions confidently. When they do not, digital infrastructure becomes a barrier rather than a bridge, and closing the distance becomes essential to keeping institutional life open to everyone.

The next chapter returns from institutional analysis to design practice. If cognitive distance shapes so much of digital life, how can teams work with it directly during development?



$$d = 15$$

CHAPTER 11

A Practical Framework for Evaluating Cognitive Distance

Up to this point, cognitive distance has been explored mainly as an explanatory concept. We have seen how it emerges from mismatched mental models, hidden institutional logic and increasingly complex technology; how usability success can coexist with misunderstanding; and how accessibility compliance does not guarantee comprehension. Chapter 6 introduced a way to measure it. The remaining question is practical: how can designers, researchers and organizations work with cognitive distance throughout the development of a system?

A concept becomes useful only when it changes practice. This chapter proposes a framework for identifying, evaluating and reducing cognitive distance across a system's lifecycle. It does not replace

existing usability methods. It extends them with a diagnostic layer focused on conceptual alignment.

11.1 From Usability Testing to Conceptual Evaluation

Shift the evaluation question

Traditional usability testing asks whether people can complete tasks. A cognitive distance evaluation asks something slightly different: do people understand what the system is doing, and why? That shift changes how research sessions are run. Instead of focusing only on performance, evaluators watch how participants reason about the system, listening for explanations, assumptions and interpretations. Moments of confusion become opportunities to study conceptual mismatch rather than isolated usability bugs.

Evaluate understanding, not only behavior

Behavioral success can hide conceptual uncertainty, and a participant may complete a workflow successfully while holding a wrong model of it. To reveal the gap, evaluators add reflective questions during or after tasks: What do you think happened when you clicked that button? Why do you think the system showed that result? What would happen if you did it differently? The answers show how people interpret the system's logic. When their explanations differ significantly from what the system actually does, cognitive distance becomes visible. Asking a prediction question *before* the result appears, as the Cognitive Distance Index does, makes the test sharper still.

Capture confidence as a metric

Confidence is a signal traditional usability testing rarely measures. Two people may complete the same task equally fast, one certain of the outcome and the other unsure and afraid of having made a mistake. The second is experiencing greater cognitive distance, even though the performance metrics are identical. Confidence can be measured with simple rating scales or short interviews, and tracking it alongside performance reveals hidden barriers. Tracking it alongside prediction accuracy reveals something more dangerous: the confident user who is wrong.

11.2 The Cognitive Distance Diagnostic Model

The Cognitive Distance Index tells a team *how much* distance exists and for whom. It does not say where the distance comes from. A high CDI on a license renewal flow might be caused by navigation organized around government departments, by labels written in administrative language, by a sequence of steps that makes no sense to the applicant, or by a result screen that explains nothing. Each cause calls for a different fix.

The diagnostic model in this section addresses that second question. It looks at the system rather than the user and examines four sources of distance: *structural* (how information is organized), *linguistic* (the words the interface uses), *procedural* (the order and logic of steps) and *explanatory* (how the system accounts for its outcomes). The CDI measures; the diagnostic model tells teams where to look. Table 11.1 shows how the two connect. The mapping is not one-to-one, since a single flaw usually shows up in more than one CDI dimension, but the dominant links are consistent enough to guide a first investigation.

If this CDI dimension is high	Look first at these sources
Perceived understanding	Linguistic (labels and result copy in system vocabulary); explanatory (outcomes reported as codes or states)
Predictive accuracy	Procedural (steps whose purpose is hidden); structural (features grouped in ways users cannot anticipate)
Interaction confidence	Explanatory (no confirmation of what happened or what comes next); procedural (no visible overview of the process)
Conceptual transparency	Explanatory (rules behind outcomes never shown); linguistic (causal information buried in administrative terms)

Table 11.1. From measurement to diagnosis: which sources of distance most often drive each CDI dimension.

In practice, a team measures first, using the CDI or a lighter version of it, then uses the diagnostic model to trace a high score back to

the design decisions that produced it, then measures again after the change. Table 11.2 turns each of the four sources into a practical lens.

Source	Diagnostic question	Observable symptom	Design response
Structural	Does the system's organization match user expectations?	Backtracking, hesitation, repeated searching	Reorganize around user goals and task intent
Linguistic	Do labels and instructions use the users' vocabulary?	Rereading, term confusion, requests for clarification	Replace jargon with outcome-centered language
Procedural	Does the workflow follow natural task reasoning?	Step-order errors, uncertainty about next actions	Show the workflow's logic; allow flexible progression
Explanatory	Can users explain why outcomes occurred?	Surprise, distrust, rote memorization	Cause-and-effect feedback and decision explanations

Table 11.2. The diagnostic scorecard: four sources of cognitive distance.

Structural alignment. This source concerns how closely the system's organization matches what people expect. Evaluators ask whether users can predict where information will be, whether navigation reflects their goals, and whether they can describe the system's structure accurately. Frequent navigation errors, hesitation and repeated exploration of unrelated sections indicate misalignment. The response is to reorganize information around user intentions rather than institutional categories.

Linguistic alignment. This source concerns how interface language shapes interpretation. Evaluators check whether the terminology matches the users' own vocabulary. Rereading labels, asking what common terms mean and misreading instructions are all signals of distance. The response is to adopt language drawn from user research and to replace system-centered terms with descriptions of outcomes.

Procedural alignment. This source concerns whether workflows correspond to real-world reasoning. Evaluators ask whether the sequence of steps matches how people approach the task, whether they can anticipate upcoming stages, and whether they understand why each step exists. Rigid or surprising sequences indicate mismatch. Responses include flexible navigation, visible overviews of the workflow, and alternative paths. Externalized processing (Section 5.4) is a procedural problem: steps the system could perform appear in the user's workflow instead.

Explanatory alignment. This source concerns whether system behavior is understandable after the fact. Evaluators examine whether people can explain the outcomes the system produced. Surprise, distrust and reliance on memorized procedures are signs of weakness here. The response is clearer feedback, visible cause and effect, and meaningful explanations of automated decisions: in short, internalized explanation (Section 8.4).

11.3 Methods Designers Can Apply Immediately

Conceptual walkthroughs

A conceptual walkthrough extends the cognitive walkthrough (Polson et al. 1992). Beyond asking whether users will know which action to take, evaluators ask whether users will understand why that action makes sense. The team simulates first-time reasoning and records each point where interpretation becomes uncertain. The method costs little and fits into existing design reviews.

Mental model interviews

Short interviews reveal conceptual gaps quickly. After completing tasks, participants describe how they believe the system works, and researchers compare these explanations with the system's actual structure. Large discrepancies indicate high cognitive distance, and even small samples tend to reveal recurring patterns.

Explanation testing

In explanation testing, participants are asked to teach the system to someone else: to explain how to complete a task and why each step

matters. If their explanations rely on memorized instructions rather than reasons, or simply repeat the system's own terms back, conceptual alignment is weak. The technique exposes whether learning has become procedural instead of conceptual, and it is the qualitative counterpart of the open explanation question in the CDI.

11.4 Integrating Cognitive Distance Into Development Cycles

Integration works best when each stage of development has a clear method for checking alignment, as Figure 11.1 shows.

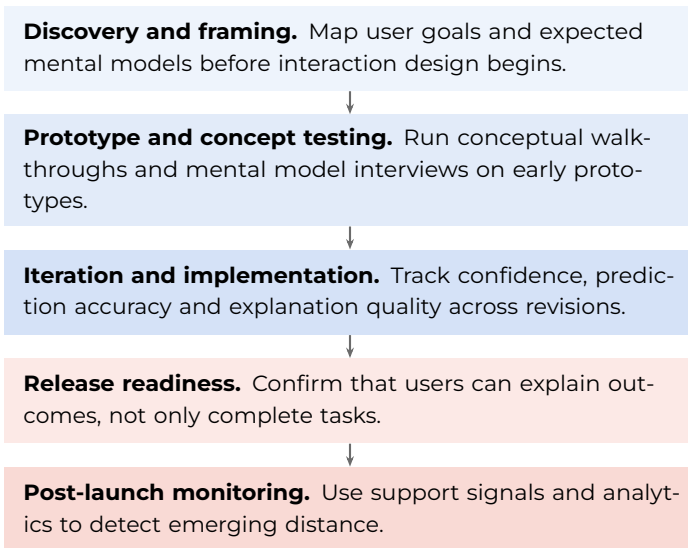


Figure 11.1. Integrating cognitive distance evaluation from discovery through post-launch learning.

Early design stages

Evaluation should begin before interfaces are fully built. Concept sketches and prototypes already reveal structural mismatches, and testing at this stage lets teams change the conceptual organization before implementation makes it expensive. Low-fidelity prototypes often work well, because they emphasize structure over visual detail.

Iterative development

During iterative cycles, cognitive distance measures can sit alongside usability testing. Teams track not only task success but understanding, confidence and the quality of explanations across versions, and see whether alignment improves over time. That longitudinal view keeps teams from optimizing performance metrics while conceptual problems persist.

Post-launch monitoring

Cognitive distance does not disappear after release. Analytics reveal behavioral signals such as repeated backtracking or abandonment at particular stages, and customer support provides more. Frequently asked questions often point to conceptual misunderstandings rather than technical faults, and support teams can tag tickets accordingly. Organizations that analyze these signals can catch emerging distance as their systems evolve.

11.5 Organizational Adoption

Building shared awareness

For cognitive distance evaluation to take hold, teams need a shared understanding of why it matters. Designers, developers, product managers and stakeholders should recognize that usability success alone does not guarantee comprehension. Workshops and review sessions that use real interaction examples, such as the two versions of a result screen in Table 6.3, introduce the concept quickly, and once teams start noticing conceptual misalignment, their design discussions tend to change on their own.

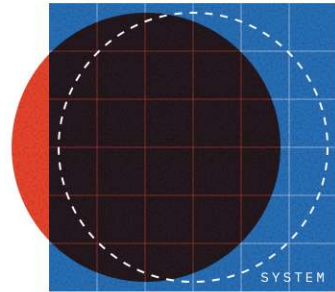
Aligning success metrics

Organizations usually measure success through efficiency indicators such as engagement, completion rates and productivity. Adding conceptual alignment measures produces a more balanced picture. Teams might track confidence ratings, the clarity of user explanations, CDI scores on key flows, and reductions in support requests caused by misunderstanding. These measures reinforce the importance of designing systems people can reason about.

Long-term cultural change

Adopting cognitive distance as a design lens gradually reshapes how an organization thinks. Rather than asking how quickly users adapt to systems, teams begin asking how systems can adapt to human reasoning. The shift is subtle but meaningful: design becomes less about optimizing the efficiency of interaction and more about supporting understanding.

The next chapter moves towards synthesis, asking what cognitive distance means for the future of Human-Computer Interaction as a field.



d = 8

CHAPTER 12

Toward a New Paradigm in Human-Computer Interaction

Every field evolves through moments of conceptual shift. For long periods, a particular idea organizes research questions, design practices and evaluation methods. Then new technological environments expose its limits, and researchers notice patterns the dominant framework cannot fully explain.

In Human-Computer Interaction, cognitive load has long played this organizing role. It helped designers understand limits on memory and attention and the effects of task complexity, and it gave them a language for overload, distraction and inefficient interaction. As digital systems have grown more complex, another pattern has become increasingly visible. Many difficulties are not caused primarily by mental overload; they emerge from conceptual misalignment. People

struggle not because they cannot process information, but because the structure of the system diverges from the structure of their reasoning. That is the space in which cognitive distance becomes useful.

The aim of this book has never been to discard cognitive load. It is to propose that Human-Computer Interaction now needs an additional lens for problems that load alone cannot explain. This chapter outlines what that lens could mean for theory, education and research.

12.1 Expanding the Theoretical Landscape

From capacity limits to conceptual relationships

Early cognitive models of interaction focused on the limits of human information processing (Card et al. 1983; Miller 1956). How much can people hold in working memory? How quickly can they process stimuli? How easily can they shift attention? These questions remain important, but they cover only one dimension of cognition. Interaction also involves interpretation, reasoning and conceptual mapping. A person approaching a system tries to understand what it represents, how its logic operates and how their actions influence its outcomes. Cognitive distance focuses directly on those relationships. Instead of asking how much information people can process, it asks how well a system's structure aligns with the mental models they construct.

Complementing established HCI frameworks

The field has produced many valuable frameworks. Donald Norman's gulfs of execution and evaluation describe the gaps between intentions, actions and system feedback (Norman and Draper 1986). Stuart Card, Thomas Moran and Allen Newell's Model Human Processor explained interaction in terms of perceptual, cognitive and motor processes (Card et al. 1983). Cognitive distance does not replace them. As Chapter 1 argued, it sits in the same intellectual tradition, closest to the gulf of evaluation, and adds a precise definition and a way of measuring the conceptual gap between system logic and user understanding.

Bridging usability and accessibility

Cognitive distance also bridges two areas that have often developed separately. Usability research focuses on efficiency, performance and satisfaction; accessibility research focuses on removing barriers for people with disabilities. They share many goals, but their theoretical frameworks have sometimes diverged. Cognitive distance connects them. When a system's structure diverges from human reasoning, everyone experiences difficulty, and for people with certain cognitive or learning differences, or who are using a system in a second language or under stress, the divergence is even more pronounced. Reducing cognitive distance therefore benefits usability and accessibility at the same time. Figure 12.1 places it among the traditions it connects.

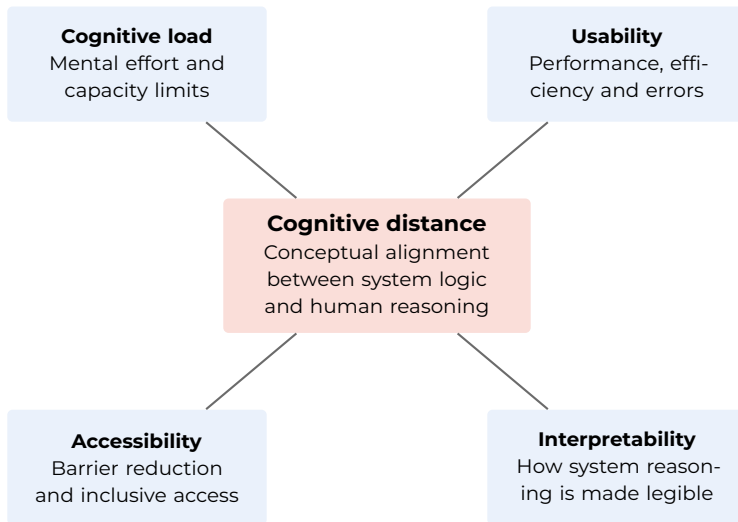


Figure 12.1. Cognitive distance as a bridge between load, usability, accessibility and interpretability.

12.2 Implications for Design Education

Teaching conceptual alignment

Design education has traditionally emphasized visual communication, layout and interaction flows. Those skills remain essential, but future curricula may give more weight to conceptual architecture: how

systems represent processes, relationships and decisions. Students need to ask what mental model people will construct when they meet an interface, whether the system's structure reinforces or contradicts that model, and where misunderstanding is likely to emerge. Teaching designers to recognize cognitive distance early can prevent many problems from appearing later.

Integrating research methods into practice

Design education also benefits from closer integration of research and practice. Students should learn the methods that reveal how people interpret systems, such as think-aloud studies, mental model interviews, concept mapping and simple prediction questions. These techniques move designers beyond assumptions about what users understand and show how people actually reason about digital systems.

Encouraging interdisciplinary thinking

Human-Computer Interaction has always drawn on psychology, computer science, design, sociology and information science, and cognitive distance reinforces that breadth. Understanding conceptual alignment requires knowledge of human reasoning, language, learning and technological architecture, and as Chapter 10 showed, of how public institutions work. Future practitioners will need to move comfortably between these traditions, combining technical expertise with a deep awareness of human cognition.

12.3 Implications for Research

Measuring conceptual misalignment

The most immediate research task is to test and refine measurement. The Cognitive Distance Index is a first proposal, and it raises questions that only data can answer: whether its four dimensions hang together statistically, whether equal weighting is justified, where the bands should really fall, and how reliably open explanations can be scored by different people. Other approaches, including structured interviews, cognitive mapping and computational analysis of user explanations, could complement it. Quantifying misalignment well would let

researchers evaluate how design changes affect understanding as rigorously as the field already evaluates speed and errors.

Studying long-term interaction

Many usability studies examine short interactions in controlled settings, but cognitive distance often reveals itself over longer periods. People begin with instructions, tutorials or trial and error, and only over time do stable mental models emerge, or fail to. Longitudinal research is needed to see how understanding evolves, how misunderstandings persist across repeated use, and whether accessibility debt accumulates the way this book suggests.

Exploring emerging technologies

New technological environments create fresh questions. AI systems, autonomous platforms and data-driven infrastructures introduce forms of interaction that are still poorly understood. Conversational systems let people express goals in natural language, yet the processes behind them remain hard to interpret. Researchers need to examine how people conceptualize these technologies and where misalignment occurs, and that work will shape the next generation of interaction design.

12.4 A Future of More Understandable Systems

Systems should communicate their logic

As digital systems grow more powerful, conceptual clarity matters more. People should not need specialized expertise to understand how everyday technologies behave. Interfaces should communicate their logic in ways that align with human reasoning. That does not mean simplifying systems beyond their function; it means designing representations that make the underlying processes intelligible.

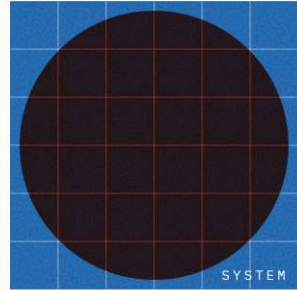
Designers as translators

One way to think about this role is as translation. Technical systems operate through formal logic, computational rules and complex architectures. People reason through experience, language, analogy and mental models. Designers work at the boundary between these two

worlds, and their task is to translate system logic into forms people can interpret. In that translation, reducing cognitive distance becomes a central responsibility.

Toward a more humane digital environment

When systems align with human reasoning, interaction becomes smoother and more empowering. People feel capable rather than confused, can predict how a system will behave, adapt to new features and stay confident while they work. At a societal level, clearer systems widen access to services, strengthen institutional trust and reduce technological exclusion. Cognitive distance is therefore not merely an academic concept. It is a design challenge with significant social consequences, and the final chapter draws together what this book has argued about how to meet it.



$$d = 0$$

CHAPTER 13

Closing the Distance

This book began with a recurring paradox. Interfaces become faster, cleaner and technically more usable, yet confusion persists. Accessibility standards expand, yet exclusion remains visible. Artificial intelligence promises natural interaction, yet people often feel less certain about what systems are actually doing.

The argument developed across these chapters is straightforward, even if its implications are not. Many failures in modern interaction are not failures of effort; they are failures of alignment. Cognitive load explains when systems demand too much mental work. Cognitive distance explains when systems make sense in a way that is fundamentally different from how people think: when people succeed procedurally and fail interpretively.

13.1 What We Have Learned

Three lessons stand out. The first is that *ease without comprehension produces dependency*. A system may let people complete tasks quickly while leaving them unsure of the outcome and unable to transfer what they did to a new situation. Research on mental models has long shown that people reason about complex environments through internal representations (Johnson-Laird 1983); when systems do not support accurate ones, performance is fragile, and people succeed temporarily rather than confidently.

The second is that *accessibility includes cognition*. Standards such as WCAG have greatly improved perceivability and operability across the web (W3C Web Accessibility Initiative 2023b), but conformance does not guarantee interpretability. A system can meet every technical requirement and still present structures that are conceptually confusing, or hand people work, like converting and compressing files, that the system could do itself. Accessibility has to include alignment with human reasoning, not only accommodation of sensory and motor differences.

The third is that *complexity is unavoidable, but confusion is not*. Digital systems increasingly manage finance, healthcare, education, governance and communication, and their internal complexity will keep growing. The goal cannot be to eliminate it. It is to make complexity legible, so that people can form accurate expectations about how a system will behave even when its mechanisms are sophisticated.

13.2 A Practical Framework for Closing Cognitive Distance

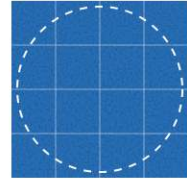
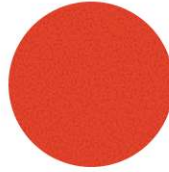
The recurring principles of the earlier chapters can be consolidated into four, summarized in Table 13.1 and pictured in Figure 13.1.

Design for mental model formation. Interfaces should help people build coherent explanations: show cause and effect clearly, keep conceptual metaphors consistent, and give feedback that explains outcomes rather than merely confirming actions.

Internalize explanation. Many systems hide their decision logic in the name of simplicity, but opacity increases uncertainty. Systems should

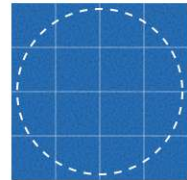
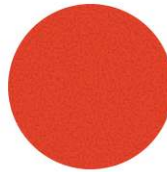
**Design for mental
model formation**

$d = 69$



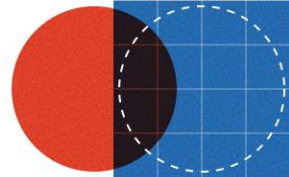
**Internalize
explanation**

$d = 46$



**Support learning
over time**

$d = 23$



**Validate understanding,
not only performance**

$d = 0$

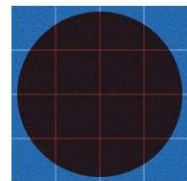


Figure 13.1. Four principles for closing cognitive distance. Each one brings the person's model closer to the one the system requires.

Principle	Design moves	Signals to track
Design for mental model formation	Show cause and effect, keep metaphors consistent, explain outcomes in context	Fewer moments of uncertainty; better transfer to new tasks
Internalize explanation	Report consequences, not states; say why; say whether anything else is needed; absorb work the system can do	Prediction accuracy and confidence rise together; fewer support requests about hidden logic
Support learning over time	Progressive disclosure, contextual guidance, interfaces that grow with expertise	Rising confidence; less relearning after updates
Validate understanding, not only performance	Add prediction and explanation questions to usability evaluation; track the CDI on key flows	Better user explanations; more resilient behavior under change

Table 13.1. Four principles for closing cognitive distance.

reveal why a recommendation appears, what data influenced an outcome and how actions changed their state. Studies of explanation suggest that well-designed explanations, even partial ones, improve the accuracy of people’s mental models and help their trust match the system’s real reliability (Kulesza et al. 2013; Lim et al. 2009).

Support learning over time. Interaction does not end after first use. Progressive disclosure of advanced concepts, explanations introduced at meaningful moments and interfaces that evolve with expertise turn use into an accumulating experience instead of repeated rediscovery.

Validate understanding, not only performance. Traditional testing stops at completion. Evaluating cognitive distance means also asking whether people can predict what the system will do, explain what happened and adapt what they learned to new tasks. When people demonstrate understanding, systems become resilient to change.

13.3 Implications Beyond Design

Cognitive distance does not originate only at the interface. Organizations create it when their internal structures shape systems around institutional logic instead of user reasoning: government portals that mirror departments rather than citizens' goals, enterprise tools that mirror database structures rather than workflows. Closing the distance therefore requires organizational awareness as well as design skill. Where digital systems function as public infrastructure, conceptual accessibility becomes a matter of equity, and policy may increasingly need to treat cognitive accessibility alongside technical accessibility. Figure 13.2 sketches how organizations can progress from basic usability towards accountable, explainable systems.

Level 4: Accountable. The organization monitors conceptual clarity and continuously corrects opacity.

Level 3: Explainable. The system gives meaningful reasons and states its uncertainty.

Level 2: Understandable. Users can explain the workflow's logic and predict likely outcomes.

Level 1: Usable. Users can complete tasks with acceptable efficiency.

Figure 13.2. A maturity ladder for cognitive distance in institutions and product teams.

13.4 The Role of the Designer and Researcher

Designers are often described as problem solvers, but another description may be more accurate: interpreters between two cognitive worlds. One works through computation, abstraction and formal logic; the other through experience, intuition and meaning-making. The designer's task is not merely to arrange interface elements but to align these worlds so that interaction becomes intelligible. Researchers play a complementary role. They identify patterns of misunderstanding, develop models that explain them, and build and test tools, like the

Cognitive Distance Index, for improving alignment. Together, design and research shape how society experiences technology.

13.5 The Distance Ahead

Human-Computer Interaction began by asking how people could adapt to machines. Over time it shifted towards making machines easier for people. The next step may be deeper: designing systems that people can understand well enough to think with, rather than merely operate. Cognitive distance offers a language for recognizing when that partnership fails, and guidance for repairing it.

Closing the distance does not mean erasing the difference between people and technology. It means ensuring that interaction stays understandable, learnable and meaningful. When people understand systems, they gain agency, and when systems align with human reasoning, technology becomes less intimidating and more empowering.

Every new technology will introduce new forms of distance, and the work will never be finished. But the direction is clearer now: not towards lighter interfaces alone, nor towards faster interaction alone, but towards understanding. That is a distance worth closing.

References

- Amershi, S. et al. (2019). "Guidelines for Human-AI Interaction". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 1–13. DOI: 10.1145/3290605.3300233.
- Berry, D. C. and D. E. Broadbent (1984). "On the Relationship Between Task Performance and Associated Verbalizable Knowledge". In: *Quarterly Journal of Experimental Psychology A* 36.2, pp. 209–231. DOI: 10.1080/14640748408402156.
- Bovens, M. and S. Zouridis (2002). "From Street-Level to System-Level Bureaucracies: How Information and Communication Technology Is Transforming Administrative Discretion and Constitutional Control". In: *Public Administration Review* 62.2, pp. 174–184. DOI: 10.1111/0033-3352.00168.
- Card, S. K., T. P. Moran, and A. Newell (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates.
- Cavalcanti, I. (2026a). *Accessibility Debt Is Real Debt*. Preprint, ResearchGate. URL: <https://www.researchgate.net/publication/400450994>.
- Cavalcanti, I. (2026b). *Cognitive Distance in Document Submission Systems: Why Users Struggle with Simple Digital Tasks*. SSRN preprint. URL: <https://ssrn.com/abstract=6663858>.
- Cavalcanti, I. (2026c). *Cognitive Distance in Information Systems: From Concept to Measurement to Field Evidence*. Preprint, ResearchGate. URL: <https://www.researchgate.net/publication/408144928>.
- Ceaparu, I., J. Lazar, K. Bessiere, J. Robinson, and B. Shneiderman (2004). "Determining Causes and Severity of End-User Frustration". In: *International Journal of Human-Computer Interaction* 17.3, pp. 333–356. DOI: 10.1207/s15327590ijhc1703_3.

- Chignell, M. H., P. A. Hancock, and A. Loewenthal (1989). "An Introduction to Intelligent Interfaces". In: *Intelligent Interfaces: Theory, Research, and Design*. Ed. by P. A. Hancock and M. H. Chignell. North-Holland.
- Conway, M. E. (1968). "How Do Committees Invent?" In: *Datamation* 14.4, pp. 28–31.
- Doshi-Velez, F. and B. Kim (2017). *Towards a Rigorous Science of Interpretable Machine Learning*. arXiv:1702.08608.
- Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts*. Arbetslivscentrum.
- Engelbart, D. C. (1962). *Augmenting Human Intellect: A Conceptual Framework*. Tech. rep. Stanford Research Institute.
- Ericsson, K. A. and H. A. Simon (1993). *Protocol Analysis: Verbal Reports as Data*. Revised. MIT Press.
- Eubanks, V. (2018). *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin's Press.
- Gentner, D. and A. L. Stevens, eds. (1983). *Mental Models*. Lawrence Erlbaum Associates.
- Herd, P. and D. P. Moynihan (2018). *Administrative Burden: Policymaking by Other Means*. Russell Sage Foundation.
- Hinds, P. J. (1999). "The Curse of Expertise: The Effects of Expertise and Debiasing Methods on Predictions of Novice Performance". In: *Journal of Experimental Psychology: Applied* 5.2, pp. 205–221. DOI: 10.1037/1076-898X.5.2.205.
- Hornbæk, K. (2006). "Current Practice in Measuring Usability: Challenges to Usability Studies and Research". In: *International Journal of Human-Computer Studies* 64.2, pp. 79–102. DOI: 10.1016/j.ijhcs.2005.06.002.
- Hutchins, E. L., J. D. Hollan, and D. A. Norman (1985). "Direct Manipulation Interfaces". In: *Human-Computer Interaction* 1.4, pp. 311–338. DOI: 10.1207/s15327051hcio104_2.
- Irizarry, T., A. DeVito Dabbs, and C. R. Curran (2015). "Patient Portals and Patient Engagement: A State of the Science Review". In: *Journal of Medical Internet Research* 17.6, e148. DOI: 10.2196/jmir.4255.
- Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge University Press.
- Kay, A. and A. Goldberg (1977). "Personal Dynamic Media". In: *Computer* 10.3, pp. 31–41. DOI: 10.1109/C-M.1977.217672.
- Kieras, D. E. and S. Bovair (1984). "The Role of a Mental Model in Learning to Operate a Device". In: *Cognitive Science* 8.3, pp. 255–273. DOI: 10.1016/S0364-0213(84)80003-8.

- Kirschner, P. A. (2002). "Cognitive Load Theory: Implications of Cognitive Load Theory on the Design of Learning". In: *Learning and Instruction* 12.1, pp. 1–10. doi: 10.1016/S0959-4752(01)00014-7.
- Kruger, J. and D. Dunning (1999). "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments". In: *Journal of Personality and Social Psychology* 77.6, pp. 1121–1134. doi: 10.1037/0022-3514.77.6.1121.
- Kulesza, T., S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong (2013). "Too Much, Too Little, or Just Right? Ways Explanations Impact End Users' Mental Models". In: *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 3–10. doi: 10.1109/VLHCC.2013.6645235.
- LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep Learning". In: *Nature* 521, pp. 436–444. doi: 10.1038/nature14539.
- Lee, J. D. and K. A. See (2004). "Trust in Automation: Designing for Appropriate Reliance". In: *Human Factors* 46.1, pp. 50–80. doi: 10.1518/hfes.46.1.50_30392.
- Lehman, M. M. (1980). "Programs, Life Cycles, and Laws of Software Evolution". In: *Proceedings of the IEEE* 68.9, pp. 1060–1076. doi: 10.1109/PROC.1980.11805.
- Li, Z. and L. Liu (2025). "Avoiding Bureaucratese: The Impact of Bureaucratic Language in Digital Public Encounters". In: *Public Management Review*. doi: 10.1080/14719037.2025.2607616.
- Lim, B. Y., A. K. Dey, and D. Avrahami (2009). "Why and Why Not Explanations Improve the Intelligibility of Context-Aware Intelligent Systems". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 2119–2128. doi: 10.1145/1518701.1519023.
- Miller, G. A. (1956). "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information". In: *Psychological Review* 63.2, pp. 81–97. doi: 10.1037/h0043158.
- Montello, D. R. (1991). "The Measurement of Cognitive Distance: Methods and Construct Validity". In: *Journal of Environmental Psychology* 11.2, pp. 101–122. doi: 10.1016/S0272-4944(05)80071-4.
- Moynihan, D., P. Herd, and H. Harvey (2015). "Administrative Burden: Learning, Psychological, and Compliance Costs in Citizen-State Interactions". In: *Journal of Public Administration Research and Theory* 25.1, pp. 43–69. doi: 10.1093/jopart/muu009.
- Nathan, M. J. and A. Petrosino (2003). "Expert Blind Spot Among Preservice Teachers". In: *American Educational Research Journal* 40.4, pp. 905–928. doi: 10.3102/00028312040004905.

- Nielsen, J. (1993). *Usability Engineering*. Academic Press. doi: 10.1016/C2009-0-21512-1.
- Nielsen, J. (1994). "Enhancing the Explanatory Power of Usability Heuristics". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 152–158. doi: 10.1145/191666.191729.
- Nielsen, J. and R. Molich (1990). "Heuristic Evaluation of User Interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 249–256. doi: 10.1145/97243.97281.
- Nooteboom, B. (2000). "Learning by Interaction: Absorptive Capacity, Cognitive Distance and Governance". In: *Journal of Management and Governance* 4.1–2, pp. 69–92. doi: 10.1023/A:1009941416749.
- Nooteboom, B., W. Van Haverbeke, G. Duysters, V. Gilsing, and A. van den Oord (2007). "Optimal Cognitive Distance and Absorptive Capacity". In: *Research Policy* 36.7, pp. 1016–1034. doi: 10.1016/j.respol.2007.04.003.
- Norman, D. A. (1988). *The Psychology of Everyday Things*. Basic Books.
- Norman, D. A. (2013). *The Design of Everyday Things*. Revised and expanded. Basic Books.
- Norman, D. A. and S. W. Draper, eds. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates.
- O'Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown.
- Paas, F. and J. J. G. van Merriënboer (1994). "Variability of Worked Examples and Transfer of Geometrical Problem-Solving Skills: A Cognitive-Load Approach". In: *Journal of Educational Psychology* 86.1, pp. 122–133. doi: 10.1037/0022-0663.86.1.122.
- Paas, F., J. E. Tuovinen, H. Tabbers, and P. W. M. van Gerven (2003). "Cognitive Load Measurement as a Means to Advance Cognitive Load Theory". In: *Educational Psychologist* 38.1, pp. 63–71. doi: 10.1207/S15326985EP3801_8.
- Polson, P. G., C. Lewis, J. Rieman, and C. Wharton (1992). "Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces". In: *International Journal of Man-Machine Studies* 36.5, pp. 741–773. doi: 10.1016/0020-7373(92)90039-N.
- Poursabzi-Sangdeh, F., D. G. Goldstein, J. M. Hofman, J. Wortman Vaughan, and H. Wallach (2021). "Manipulating and Measuring Model Interpretability". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 1–52. doi: 10.1145/3411764.3445315.

- Reeves, B. and C. Nass (1996). *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge University Press.
- Ribeiro, M. T., S. Singh, and C. Guestrin (2016). ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. doi: 10.1145/2939672.2939778.
- Schuler, D. and A. Namioaka, eds. (1993). *Participatory Design: Principles and Practices*. Lawrence Erlbaum Associates.
- Seffah, A. and E. Metzker (2004). “The Obstacles and Myths of Usability and Software Engineering”. In: *Communications of the ACM* 47.12, pp. 71–76. doi: 10.1145/1035134.1035136.
- Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2017). “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626. doi: 10.1109/ICCV.2017.74.
- Shneiderman, B. (1983). “Direct Manipulation: A Step Beyond Programming Languages”. In: *Computer* 16.8, pp. 57–69. doi: 10.1109/MC.1983.1654471.
- Smith, D. C., C. Irby, R. Kimball, B. Verplank, and E. Harslem (1982). “Designing the Star User Interface”. In: *Byte* 7.4, pp. 242–282.
- Sweller, J. (1988). “Cognitive Load During Problem Solving: Effects on Learning”. In: *Cognitive Science* 12.2, pp. 257–285. doi: 10.1207/s15516709cog1202_4.
- Sweller, J. (2011). “Cognitive Load Theory”. In: *The Psychology of Learning and Motivation*. Ed. by J. P. Mestre and B. H. Ross. Vol. 55. Academic Press, pp. 37–76. doi: 10.1016/B978-0-12-387691-1.00002-8.
- Tintarev, N. and J. Masthoff (2007). “A Survey of Explanations in Recommender Systems”. In: *IEEE 23rd International Conference on Data Engineering Workshop*, pp. 801–810. doi: 10.1109/ICDEW.2007.4401070.
- United States Congress (1990). *Americans with Disabilities Act of 1990*, Pub. L. No. 101-336, 104 Stat. 327.
- W3C Web Accessibility Initiative (2023a). *Cognitive Accessibility at W3C*. URL: <https://www.w3.org/WAI/cognitive/>.
- W3C Web Accessibility Initiative (2023b). *Web Content Accessibility Guidelines (WCAG) 2.2*. URL: <https://www.w3.org/TR/WCAG22/>.
- Weiser, M. (1991). “The Computer for the 21st Century”. In: *Scientific American* 265.3, pp. 94–104. doi: 10.1038/scientificamerican0991-94.

- Weizenbaum, J. (1966). "ELIZA: A Computer Program for the Study of Natural Language Communication Between Man and Machine". In: *Communications of the ACM* 9.1, pp. 36–45. DOI: 10.1145/365153.365168.
- Zaheri, M., M. Famelis, and E. Syriani (2025). "Workarounds in Software Forms: A User Study". In: *Proceedings of the 2025 IEEE International Conference on Collaborative Advances in Software and Computing (CASCON)*, pp. 439–447. DOI: 10.1109/CASCON66301.2025.00073.
- Zhang, S., H. Wang, and X. Yi (2025). "Exploring Collaboration Patterns and Strategies in Human–AI Co-Creation Through the Lens of Agency: A Scoping Review of the Top-Tier HCI Literature". In: *Proceedings of the ACM on Human-Computer Interaction* 9.7, pp. 1–43. DOI: 10.1145/3757594.

About the Author

Izaías Cavalcanti researches cognitive distance in information systems, drawing on a background in UX and product design. The work examines why people complete digital tasks without understanding what the system did, and how that gap can be measured and designed away, with a particular focus on public services, banking and healthcare.

Cavalcanti is the author of a series of papers developing the cognitive distance framework, including the Cognitive Distance Index and the Everyday Services Study, and also writes about product design for practitioners. *From Cognitive Load to Cognitive Distance* is Cavalcanti's first book.

<https://book.izaiaas.xyz>

People can finish a task and still not understand what just happened.

For decades, usability and accessibility have been judged by effort: clicks, time, memory load. Yet people often fail not because a system asks too much of them, but because its logic is foreign to them.

This book introduces cognitive distance: the mismatch between the conceptual structure a system requires and the mental model a person actually builds while using it. Distance is relational, it accumulates across an interaction, and it can be high even when effort is low. That is why compliant, efficient systems still leave people behind.

INSIDE THIS BOOK

- Why task success can hide failed understanding
- How systems quietly offload interpretive work onto users
- A framework for measuring distance and designing for alignment

ABOUT THE AUTHOR

Izaias Cavalcanti researches cognitive distance in information systems, drawing on a background in UX and product design.



First Edition · 2026

barcode area (KDP)