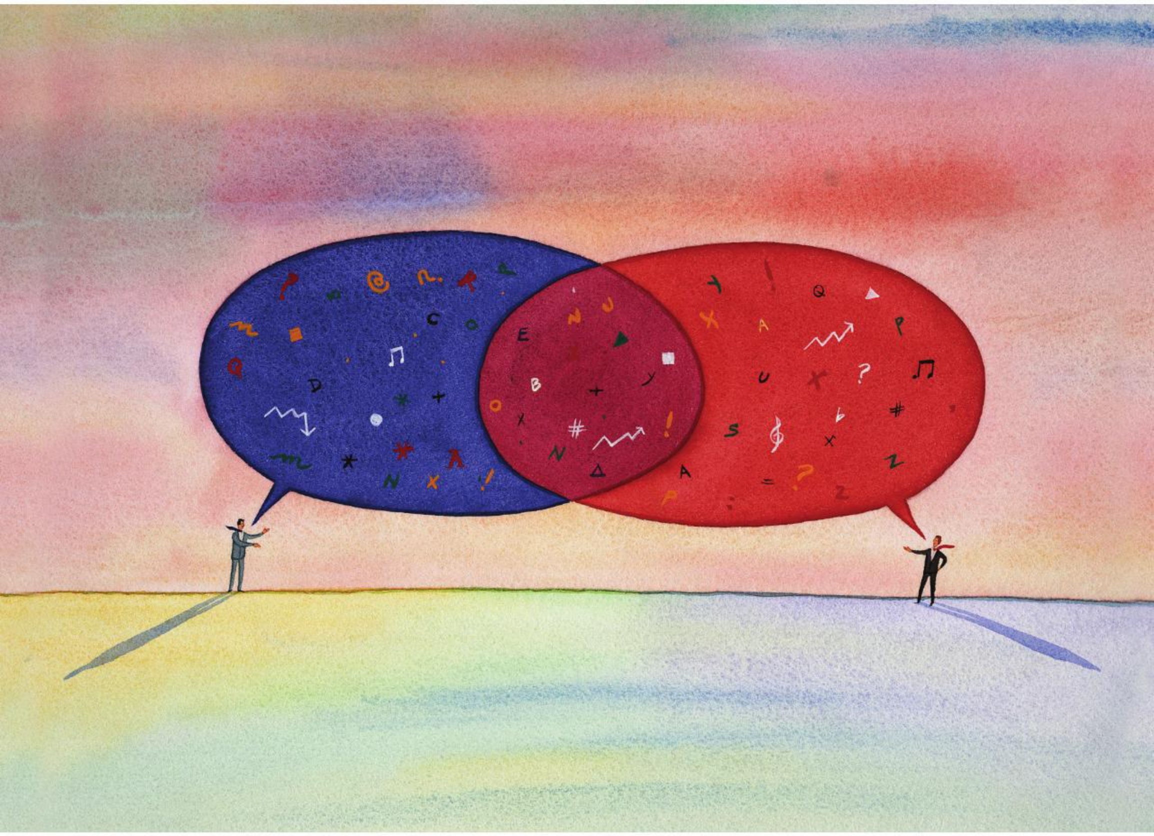
O'REILLY®

"Truly, a blueprint for the systems of tomorrow."
–Mike Dvorkin, Distinguished Engineer at Cisco

THINKING IN PROMISES

DESIGNING SYSTEMS FOR COOPERATION



MARK BURGESS

Thinking in Promises

Mark Burgess

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo O'REILLY®

Thinking in Promises

by Mark Burgess

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editors: Mike Loukides and Brian Anderson	**Indexer:** Ellen Troutman-Zaig
Production Editor: Melanie Yarbrough	**Interior Designer:** Monica Kamsvaag
Copyeditor: Sonia Saruba	**Cover Designer:** Ellie Volkhausen
Proofreader: Charles Roumeliotis	**Illustrator:** Rebecca Demarest

June 2015: First Edition

Revision History for the First Edition

2015-06-09: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491917879* for release details.

978-1-491-91787-9

[LSI]

Contents

Foreword

Throughout much of human history, our technological achievement was driven by the quest for certainty and precision. Chaos in all its forms had to be defeated by the clockwork-like precision of the machines we built. We altered the course of flow of rivers. We achieved space flight. We created hierarchical societal structures with protocols and processes. But despite the mathematical precision, we never managed to achieve the robustness and efficiency of natural processes.

Our human intuition tells us that precise understanding of the underlying processes and consistency of their execution are key to well-functioning systems. This belief gives us a seemingly practical mental framework where all conditions affecting such systems are known and understood, or, at least, can be approximated or anticipated in some way. We further quantize the behaviors into sets of well-understood rules and actions and combine them in hierarchically organized global knowledge. A system of consistency hierarchies.

This approach led us to industrialization, but it also gave us an equivalent of industrialized micromanagement. Our control and automation systems turned into the equivalent of sequenced linear execution plans, where each step is a precise action—certain, predictable, and consistent. Any significant inconsistency leads to local and often cascading failure, at times, with global consequences—all with little or no ability to reconverge.

Our cultural preference for understanding the complete picture led us to the creation of control systems that function as large, centralized, all-knowing brains that make precise decisions based on assumptions of availability and consistency, resulting in actions that are elemental and imperative in nature—our intuitive strive to micromanage the reality. Such centralization, even if it's logical, leads to inadvertent limitations in scale. A centralized control brain, as large as it can be, still has finite processing capacity. A busy brain can't make decisions fast enough to carry out micromanaging actions in time. Latency of actions creates imprecision and inconsistency, often with unknown consequences.

In this book, Mark Burgess discusses a different philosophical approach to solving large-scale control problems that entirely eliminates the need for a big almighty brain and the very need for micromanagement. Instead, he focuses on breaking the system down into a large number of primitive functions, each fully autonomous, self-enforced, with the ability to deal with imprecision at the lowest level possible. Like in nature, these functions are not centrally orchestrated—they act like organisms that interact with each other in the form of promises, not subjected to absolute universal obligations, not requiring precise consistent knowledge.

—Mike Dvorkin
Distinguished Engineer at Cisco
June 2015

Letter to the Reader

Dear reader, this book is an introduction to what has come to be known as Promise Theory. It is addressed from a mostly nontechnical perspective. It is not a book that gives management advice, nor is it even about technical recipes; its goal is to help you to think without prejudice about cooperative systems of any kind.

We live in a marketing age where we believe that brands mean something, so today you'll read about Complexity Theory and Promise Theory and Theory of Constraints, and any number of other theories. Many believe that these ideas are all separate, competing sports teams, from which we are allowed to choose only one. Add to that Lean, Agile, Six Sigma, and any number of other management philosophies, and you are practically forced into primitive tribalism.

Promise Theory is not a management ideology; it is an engineering framework for coping with uncertainty in information systems. It is a set of principles, based on formal reasoning, without the over-constrained ideas of logic. Some people (not mathematicians) might call it mathematical because it has formal rule- or constraint-based approaches.[1] Others might say it is heuristic. These are subjective assessments, which are unimportant. In fact, Promise Theory has a lot in common with physics, being a mixture of both, but who cares about categories?

One of the reasons I started Promise Theory was to escape from the tyranny of fuzzy words that float around in management-speak. Words are verbose and often ambiguous, and people hear what they want to hear from them. Symbolic languages, like mathematics, express themselves simply and clearly, but many

1 In this book, I was instructed (on pain of retribution) to avoid anything that might resemble math. You have no idea the oppression that mathematicians face in society.

have been scared away from that precision by a poor experience with mathematics in school. For that reason, I started Promise Theory symbolically, but was quickly asked to go back to using words. That is what this book tries to do. Hopefully, having done a sufficient amount of symbolic homework, I can avoid some of the ambiguities that come from muddling words, but I am still wary. You should be too.

So why this word *promise*? Well, a promise is a word that everyone intuitively understands. It represents an intended outcome: something you might actually be able to get. This is pretty general, and indeed that is its value. For some, it is also confusing. The term does not have the arrogance or hubris of the more frequently used *guarantee*, and that is good. A broad swathe of society actually believes that the concept of a guarantee has merit (that merely promising without guarantee is somehow shirking responsibility). In fact, unless one has access to irresistable force and infinite speed, there are no guarantees.

Promise Theory has nothing to do with being nice. It is not about being moral. It is not about democracies rather than dictatorships. It is simply about realism, and attaching the responsibility for outcomes to the agencies that have the chance of being able to deliver. It doesn't tell you how to be immune to uncertainty, it just helps to clear away a veil of delusion, to see more clearly what kind of a mess you are in. The rest (getting yourself out of the mess) is up to good old human creativity (there you can hug as much as you like). What I will claim is that if, like me, you live in daily fear of appearing foolish by making unwarranted assumptions, then Promise Theory is the cold shower of common sense. It turns out to be a pragmatic way of coming to terms with the complexities of a world where you are not certain of having your way.[2]

Promise Theory came out of a need I had while trying to explain IT infrastructure and desired-state configuration in the early millennium years: studying how to manage human-computer systems. It became clear that computer science models, which were based on traditional logical reasoning, simply weren't able to describe computer behaviour, except in very isolated circumstances. Initially more mathematical, the ideas grew in scope and applicability, and over the

2 I have confirmed over the years that the wish for deterministic mastery over our world often works as a kind of homeopathy (trying to treat a malady with its own poisons). When we try to fight force with force (uncertainty with hopeful certainty), we end up in some kind of boxing match, often getting our proverbial rear-ends kicked. Smart martial artists try to use their opponents' real-time behaviours to their own advantage, and that is sort of the essence of Promise Theory. It's not what you *want* or *hope* for, but what you can *get* that matters. Pragmatism rules.

following decade began to be appreciated in a heuristic way, too. In a sense, Promise Theory is about one of the most general, yet contentious, issues of philosophy: causation. How things come to happen, and then remain, even in a noisy world.

As I was having my private eureka moment about promises in 2004, I was visited by Jan Bergstra, whose background was in logic and mathematical computer science. He also became interested in this issue from a background of rigorous logic and process algebra. Now my close friend and partner in what amounts to computer science heresy, Jan has been instrumental in boiling the Promise Theory story down to its essentials, and avoiding the pitfalls by which science becomes pseudoscience.

Amazingly, the idea of promises seems to have caught on in IT. However, for some, the idea has also become a manifesto for advocating ideas like decentralization and swarm intelligence. Although I am a big fan of those ideas, one should not think of Promise Theory as advocating them. Science does not advocate; it measures by theory and numbers, then tentatively concludes in context. An application of Promise Theory might lead to the conclusion that one design or another is better under particular circumstances, but never unilaterally.[3]

Looking at the philosophical and economic literature on promises, philosophers have muddled the idea of promises with the notions of obligation and morality. What caught our attention early on was that promises are an independent and simpler idea than obligations: something suitable for engineers.

The aim of this book, then, is to strip away the formalism—which I worked so hard to erect(!)—and introduce the pattern of thinking behind Promise Theory for engineers by words and pictures alone. This has been a difficult challenge because words alone lack the concision and clarity to make clear sense of a complex world. A more symbolic approach has been given in the book *Promise Theory: Principles and Applications* (CreateSpace), by myself and Jan Bergstra, and I recommend that book to clear up any questions you might have.

Writing a book for as general an audience as possible can easily result in the paradox of pleasing no one by trying to please all, so please forgive the compromises. I address the book to an audience of fellow technocrats. I hope, on the

3 It is perhaps natural that people gravitate to emergent phenomena and decentralization. These are interesting concepts, yet traditional computer science struggles to understand these issues with its deterministic "ballistic" reasoning.

other hand, that large parts of it can be read and understood by just about anyone in the modern world.

I am grateful to Michael Nygaard, Jeff Sussna, Paul Borrill, Mike Dvorkin, and, of course, Jan Bergstra for comments.

Promises and Impositions

Imagine a set of principles that could help you understand how parts combine to become a whole, and how each part sees the whole from its own perspective. If such principles were any good, it shouldn't matter whether we're talking about humans in a team, birds in a flock, computers in a data center, or cogs in a Swiss watch. A theory of cooperation ought to be pretty universal, so we could apply it to both technology and the workplace.

Such principles are the subject of Promise Theory. The goal of Promise Theory is to reveal the behaviour of a whole from the sum of its parts, taking the viewpoint of the parts rather than the whole. In other words, it is a bottom-up constructionist view of the world. You could describe it as a discipline for documenting system behaviours from the bottom up.[1]

Promise Engineering

The idea of using promises as an engineering concept came up in 2004, as I was looking for a model of distributed computing to describe CFEngine. The word *promise* seemed a good fit for what I needed: a kind of atom for intent that, when combined, could represent a maintainable policy. However, it quickly became clear (having opened Pandora's box on the idea) that there was something more general going on that needed to be understood about promises. Promises could also be an effective way of understanding a whole range of related issues about

1 There have been many theories of promises in the past, but here we refer to my work with collaborators. I described a more formal or mathematical account of Promise Theory in *Promise Theory: Principles and Applications*.

how parts operate as a whole, and it promised[2] something not previously taken seriously: a way to unify human and machine behaviours in a single description.

Unlike some other modelling methods, such as in business and computer science, Promise Theory is not a manifesto, nor is it a political statement or a philosophical agenda. The magic lies in the application of a simple set of principles. It is little more than a method of analysis and engineering for picking systems apart into their essential pieces and then putting them back together. Along the way, we find a method of representing and questioning the viability of our intended outcomes. For some people, this is what computer programming is about, and there have been many philosophies around this, like OO, SOA, UML, and so on. Many of these have failed because they set philosophical agendas ahead of understanding.

The purpose of this book is to ask what can an understanding in terms of promises tell us about cooperation in human-machine systems, organizations, and technology, and how can we apply that understanding to the real-life challenges of working together?

From Commands to Promises

The cultural norm, at least in Western society, is to plan out intended outcomes in terms of the commands or steps we believe we need to accomplish in order to get there. We then program this into methods, demanding milestones and deliverables to emphasize an imperative approach to thinking. This is because we think in stories, just as we relate stories through language. But stories are hard to assess. How do we know if a story succeeded in its intent?

If, on the other hand, we change focus away from the journey to think in terms of the destination, or desired outcome, assessment and success take on a whole new meaning.

Let's look at an example. Consider the following instructions for cleaning a public restroom:

Wash the floor with agent X.

Mop and brush the bowls.

Put towels in the dispenser.

2 Perhaps the most important thing about Promise Theory is that it drives people to the most terrible puns, without realizing that those puns say something involuntarily insightful, too.

Refill soap.

Do this every hour, on the hour.

Now let's convert this into a promise formulation:

I promise that the floor will be clean and dry after hourly checks.

I promise that the bowls will be clean and empty after hourly checks.

I promise that there will be clean towels in the dispenser after hourly checks.

I promise that there will be soap in the dispenser after hourly checks.

What's the point of this? Isn't the world about forces and pushing changes? That has been the received learning since the time of Newton, but it is an oversimplification, which is not accurate even in modern physics.

The first thing we notice is that some agent (a person or robot) has to make the promise, so we know who is the active agent, and that by making the promise, this agent is accepting the responsibility for it. The second thing we notice is a lack of motivation to make the promise. Cooperation usually involves dialogue and incentive. What is missing from these promises is a counterpart like: I promise to pay you if the bowls are clean. Thus a promise viewpoint leads to a question: how do we document incentives?

Why Is a Promise Better than a Command?

Why a promise? Why not an obligation, command, or requirement? In Promise Theory, these latter things are called impositions because they impose intentions onto others without an invitation.

Promises expose information that is relevant to an expected outcome more directly than impositions because they always focus on the point of causation: the agent that makes the promise and is responsible for keeping it.

Commands and other impositions fail us in two ways: they tell something how to behave instead of what we want (i.e., they document the process rather than the outcome), and they force you to follow an entire recipe of steps before you can even know what the intention was.

So, why is a promise better than a command? A promise expresses intent about the end point, or ultimate outcome, instead of indicating what to do at the starting point. Commands are made relative to where you happen to be at the moment they are issued, so their relevance is limited to a brief context. Promising the end state is independent of your current state of affairs (see Figure 1-1).

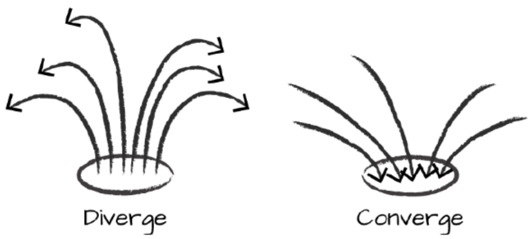


Figure 1-1. A theme we'll revisit several times in the book is the divergence or convergence of timelines. On the left, commands fan out into unpredictable outcomes from definite beginnings, and we go from a certain state to an uncertain one. On the right, promises converge towards a definite outcome from unpredictable beginnings, leading to improved certainty.

Promises also express intent from a viewpoint of maximum certainty. Promises apply to you (self) — the agent making them. By the definition of autonomy, that self is what every agent is guaranteed to have control over. Impositions or commands are something that apply to others (non-self). That, by definition, is what you don't control.

It's possible to promise something relative to a starting point: "I promise to stand on my head right now." "I will leave the house at 9:00 a.m." But promises represent ongoing, persistent states, where commands cannot. They describe continuity.[3]

Autonomy Leads to Greater Certainty

A promise is a claim made by someone or something about an intended outcome. It is not an insistence on action, or an attempted coercion. It is an expression of what I'll call *voluntary behaviour*. As we look at more examples, the sense in which I use the term voluntary should become clearer. Another word for the same thing would be *autonomous*.

An agent is autonomous if it controls its own destiny (i.e., outcomes are a result of its own directives, and no one else's). It is a "standalone" agent. By

3 You can always twist the usual meanings of command and promise to contradict these points, so we agree to use limited meanings that correspond to normal usage. In other words, this is not about the words, but about the ideas they usually represent. You can command yourself to empty the trash every Wednesday. Or you can promise someone that you are going to do a somersault right now. But it is hard to command yourself to do something persistently.

dividing the world up into autonomous parts, we get a head start on causality. When a change happens, we know that it happens within an autonomous region; it could not happen from the outside without explicitly promising to subordinate itself to an outside influence.

Note that a behaviour does not have to be explicitely intended for it to be intentional. It only needs to be something that *could* be intended or has the appearance of being intended.

The Observer Is Always Right

When we make a promise, we want to communicate to someone that they will be able to verify some intended outcome. Usually a promise is about an outcome that has yet to happen (e.g., "I promise to get you to the church on time for your wedding"). We can also promise things that have already happened, where it is the verification that has yet to happen (e.g., if your accounts department says, "I promise that I paid the bill on time"); the outcome has already happened, but the promisee has not yet verified the claim.

Why is this important? In fact, every possible observer, privy to relevant information, always gets to make an independent assessment of whether a promise was kept or not. The promisee might be the one especially affected by the promise outcome, but is not the only one who can form an opinion.

For example, suppose Alice promises Bob that she paid him some money. Bob was not there when she transferred the money to his account, so he cannot assess the promise until he checks his account. However, Carol heard the promise as a bystander, and she was present when Alice made the transfer. Thus, she can assess the promise as kept. Bob and Carol thus differ in their assessments because they each have access to different information.

This idea, that each autonomous agent has its own independent view, means that agents form expectations independently, too. This, in turn, allows them to make judgements without waiting to verify outcomes. This is how we use promises in tandem with trust. Every possible observer, with access to part of the information, can individually make an assessment, and given their different circumstances, might arrive at different conclusions.

This is a more reasonable version of the trite business aphorism that "the customer is always right." Each observer is entitled to his own viewpoint. A useful side effect of promises is that they lead to a process of documenting the conditions under which agents make decisions for later use.

Culture and Psychology

There are cultural or psychological reasons why promises are advantageous. In the imperative version of the restroom example, you felt good if you could write down an algorithm to bring about the desired end state, even once. Your algorithm might involve a checklist of items, like scrubbing bowls, using a special detergent, and so on. Writing down the steps feels pedagogical because it tells you *how*. It might be good for teaching someone how to keep a promise in the future, but it does not make clear what the final outcome should be, or whether there is more than one way to achieve the outcome. Thus, without a promise, one cannot assess the algorithm. With a promise, we can be clear about the desired end state, and also discuss alternative ways to bring it about.

The *how* is the designer part. What about the running and maintenance part of keeping a promise over time, and under difficult circumstances? In the world of information technology, design translates into "development" and maintenance translates into "operations," and understanding both together is often called DevOps.

If you imagine a cookbook, each page usually starts with a promise of what the outcome will look like (in the form of a seductive picture), and then includes a suggested recipe. It does not merely throw a recipe at you, forcing you through the steps to discover the outcome on trust. It sets your expectations first. In computing programming, and in management, we are not always so helpful.

Promises fit naturally with the idea of services.[4] Anyone who has worked in a service or support role will know that what you do is not the best guide to understanding: "Don't tell me what you are doing, tell me what you are trying to achieve!" What you are actually doing might not be at all related to what you are trying to achieve.

A simple expression of intent is what we call a *promise proposal*. By telling it like you mean it, it becomes a promise.

Nonlocality of Obligations

A major issue with impositions, especially "obligations," is that they don't reduce our uncertainty of a situation. They might actually increase it. Obligations

4 When I first proposed the concept in 2004, it was met with the response: this is just Service-Oriented Architecture (SOA). Although SOA is about promises, Promise Theory goes far beyond SOA's scope and goals.

quickly lead to conflicts because they span a region of our world about which we certainly have incomplete information.

Imagine two parents and a child. Mum and Dad impose their speech patterns on their innocent progeny as follows. Mum, who is American, tells the child, "You say tomaetoe," while English Dad says, "I say tomahtoe." Mum and Dad might not even be aware that they are telling the child different things, unless they actually promise to communicate and agree on a standard. So there is a conflict of interest.

But the situation is even worse than that. Because the source of intent is not the child, there is nothing the child can do to resolve the conflict; the problem lies outside of her domain of control: in Mum and Dad. Obligations actually increase our uncertainty about how the child will behave towards another agent.

The solution is to invoke the autonomy of all agents. Neither the child nor the Mum or Dad have to obey any commands or obligations. They are free to reject these, and choose or otherwise make up their own minds. Indeed, when we switch to that viewpoint, the child has to promise Mum or Dad what she intends to say. In fact, she is now in possession of the information and the control, and can promise to say one thing to Mum and another to Dad without any conflict at all.

Isn't That Quasi-Science?

Scientists (with the possible exception of certain social scientists) and engineers will bristle uncomfortably at the idea of mixing something so human, like a promise or intention, with something that seems objectively measurable, like outcomes in the real world. We are taught to exorcize all reference to humanity in natural science to make it as objective as possible. Part of the reason for this is that we have forgotten a lot of the philosophy of science that got us to the present day, so that we now believe that natural science is in some sense "objective" (more than just impartial).

In my book, *In Search of Certainty* (O'Reilly), I describe how the very hardest natural sciences have forced science to confront the issues of observer relativity (or what we might call *subjective issues*), in an unexpected twist of fate. As a physicist myself, it took me a while to accept that human issues really have to be represented in any study of technology, and that we can even do this without descending into talk about feelings, or moral outrage over privileged class systems, and so on.

The idea that a promise is more fundamental than a command or an obligation is not difficult to understand. It has to do with simple physics: promises are local, whereas obligations are distributed (nonlocal).

The goal of Promise Theory is to take change (dynamics) and intent (semantics) and combine these into a simple engineering methodology that recognizes the limitations of working with incomplete information. Who has access to what information?

When we describe some behaviour, what expectations do we have that it will persist over time and space? Is it a one-off change, like an explosion, or a lasting equilibrium, like a peace treaty?

Semantics and Dynamics

Dynamics are the aspects of a system that can be measured, like sizes, speeds, rates, and so on. Dynamical aspects of systems can be characterized objectively by numbers (“data”), and these numbers exist independently of an interpretation. In science, the words *mechanics* and *kinematics* are also used, but they are less familiar to a wide audience.

Semantics are about how we interpret something: what does it mean, what is its function, what significance do we attach to it? Semantics are subjective (i.e., in the eye of the beholder); hence one agent might assess a promise to be kept, while another assesses it to be not kept, based on the same dynamical data.

Is Promise Theory Really a Theory?

Like any scientific method, Promise Theory is not a solution to anything specific; it is a language of information to describe and discuss cooperative behaviour among different agents or actors. If you operate within the framework of its assumptions and idioms, it will help you frame assumptions and find possible solutions to problems where distributed information is involved.

Promise Theory is, if you like, an approach to modelling cooperative systems that allows you to ask: “How sure can we be that this will work?” and “At what rate?” You can begin to answer such questions only if some basic prerequisites can be promised by an agency involved in the possibly collaborative promise to “make it work.”

Promise Theory is also a kind of *atomic theory*. It encourages us to break problems down into a table of elements (basic promises), from which any substantial outcome can be put together like a chemistry of intentions (once an intention about self is made public, it becomes a promise). SOA is an example of a promise-oriented model, based on web services and APIs, because it defines autonomous services (agents) with interfaces (APIs), each of which keeps well-documented promises.

The principles behind Promise Theory exist to maintain generality and to ensure that as few assumptions as possible are needed to predict an outcome. They also take care of the idea that every agent's worldview is incomplete (i.e., there are different viewpoints), limited by what different parties can see and know.

What is unusual about Promise Theory, compared to other scientific models, is that it models human intentions, whether they are expressed directly by a human or through a technological proxy, and it does this in a way that is completely impersonal. By combining Promise Theory with game theoretic models, we can also see how cooperation can ultimately have an economic explanation (sometimes referred to as *bounded rationality*). Why should I keep my promises? What will I get out of it?

The Main Concepts

We will refer to the following key concepts repeatedly:

Intention
: This is the subject of some kind of possible outcome. It is something that can be interpreted to have significance in a particular context. Any agent (person, object, or machine) can harbour intentions. An intention might be something like "be red" for a light, or "win the race" for a sports person.

Promise
: When an intention is publicly declared to an audience (called its *scope*) it then becomes a promise. Thus, a promise is a stated intention. In this book, I'll only talk about what are called promises of the first kind, which means promises about oneself. Another way of saying this is that we make a rule: no agent may make a promise on behalf of any other (see Figure 1-2).

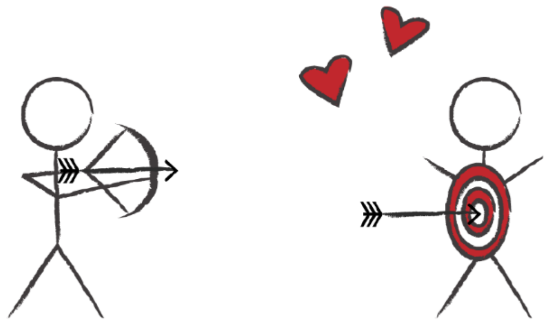


Figure 1-2. A promise to give is drawn like Cupid's arrow...

Imposition

This is an attempt to induce cooperation in another agent (i.e., to implant an intention). It is complementary to the idea of a promise. Degrees of imposition include hints, advice, suggestions, requests, commands, and so on (see Figure 1-3).

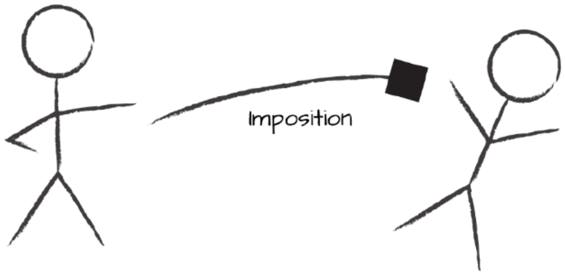


Figure 1-3. An imposition is drawn like a fist.

Obligation

An imposition that implies a cost or penalty for noncompliance. It is more aggressive than a mere imposition.

Assessment

A decision about whether a promise has been kept or not. Every agent makes its own assessment about promises it is aware of. Often, assessment involves the observation of other agents' behaviours.

There are other levels of interaction between agents. One could, for example speak of an attempt to force an agent to comply with an imposition, which might be termed an attack; however, we shall not discuss this further as it leads to discussions of morality, which we aim to avoid as far as possible.

Promises are more common than impositions and hence take precedence as the primary focus. Impositions generally work in a system of preexisting promises. Moreover, promises can often be posited to replace impositions with equivalent voluntary behaviours.

How Much Certainty Do You Need?

Promise Theory is still an area of research, so we shouldn't imagine it has an answer to everything. Moreover, it synthesizes ideas also discussed in other theories, like graph theory and relativity, so we should not imagine it is something completely new. It starts with a minimal set of assumptions, and then goes on to describe the combined effect of all the individual promises, from the viewpoint of the different parts of the whole, to form a network of cooperation. If you would like to understand it more deeply, I encourage you to study it in a more formal, mathematical language.

The word *promise* is one that seems familiar, and invites certain associations. In Promise Theory,[5] it has a specific and clear meaning. Others have taken the word for technical usage, too: futures and promises are discussed in concurrent programming. These also take the common word and attribute specialized meaning to it. We need to be careful not to project too many of our own imaginings into the specialized meanings.

As you read this book, you will find that Promise Theory says a lot of things that seem obvious. This is a good thing. After all, a theory that does not predict the obvious would not be a very good theory. Then there will be other conclusions that stretch your mind to think in unfamiliar ways, perhaps cutting through cultural prejudices to see more clearly. You might be disappointed that there there are no stunning revelations, or you might be wowed by things you have never realized before. It all depends on where you start your thought process. Whatever your experience, I hope this book will offer some insights about formulating and designing systems for cooperation.

5 Promise Theory, in the sense of this book, refers to a specific theory that emerged from my work around distributed computing. There are other theories about the meaning of promises in philosophy, which I promise to politely ignore throughout this book.

A Quick User Guide

Let's briefly sketch how to start thinking in promises, before delving into the details. It boils down to a few rules of thumb:

Identify the key players (agents of intent)

The first step in modelling is to identify the agencies that play roles within the scope of the problem you are addressing. An agent is any part of a system that can intend or promise something independently, even by proxy. Some agents will be people, others will be computers, policy documents, and so on—anything that can document intent regardless of its original source.

To get this part of the modelling right, we need to be careful not to confuse intentions with actions or messages. Actions may or may not be necessary to fulfill intentions. Maybe inaction is necessary!

Also, you shouldn't be squeamish about attributing promises to blunt instruments. We might be talking about the parts of a clock, or even an HTTP request, as agencies within a system. The motivations that play a role in making a bigger picture are not necessarily played out by humans in the end game.[6]

To be independent, an agent only needs to think differently or have a different perspective, access to different information, and so on. This is about the separation of concerns. If we want agents that reason differently to work together, they need to promise to behave in a mutually beneficial way. These agents can be humans (as in the business-IT bridge) or computers (as in a multitier server queue).

Deal with the uncertainties and obstacles

How likely is it that the agent will be able to keep the promise? In the real world there is no absolute certainty, so forget about that right now! Dealing with uncertainty is what science is really for, so roll up your sleeves and prepare to engineer your promises to make the best of what you have to work with. There are techniques for this.

6 All intentions originate with human observers if we trace them back far enough. But many day-to-day objects can be vehicles of those intentions, and therefore act as proxies. A cup is just a piece of ceramic; its intent to act as a cup is something a designer (acting on behalf of a user) decided. From a modelling perspective, that chain of provenance is not usually important, so we simply attach the promise to the inanimate cup. Now that it exists, that is the promise it makes to potential users.

The bottom line is that promises might or might not be kept (for a hundred different reasons). After all, they are only intentions, not irresistible forces.

Machines and people alike can break down and fail to keep a promise, so we need to model this. Each promise will have some kind of likelihood (perhaps even a formal probability) associated with it, based on our trust or belief in its future behaviour.[7]

Agents only keep promises about their own behaviour, however. If we try to make promises on others' behalf, they will most likely be rejected, impossible to implement, or the other agent might not even know about them. So it is a *pull* or *use what's promised* model of the world rather than a *push* or *try to impose on others* model. It assumes that agents only bend to external imposition if they want to (i.e., control cannot be pushed by force). That means we have to look more realistically upon illusions like forcible military command structures, and see them as cases where there is actually a consensus to voluntarily follow orders—even when these sometimes fail.

From requirements to promises (top-down to bottom-up)

Promise Theory focuses attention on the active agents for two reasons: first, because these are the ones that know most about their own ability to keep promises. Second, because the active agents are the atomic building blocks that can be added easily into any larger act of cooperation. Requirements get imposed top-down. Promises are kept bottom-up.

This is analogous to the insight made by atomic theory. Think of chemistry and the table of atomic elements. No one can invent a new element by imposing a requirement. Imagine designing a plane that requires a metal with twice the strength of steel but half the weight of aluminium. You can try writing to Santa Claus to get it for Christmas, but the laws of physics sort of get in the way. We can dream of things that are simply not possible, but if we look at what the world promises and try to build within that, instead of dreaming arbitrarily, we will make real progress. From the promised chemistry of the basic elements, we can build combinations of

7 The simplistic two-state model of faults, where manufacturers like to talk of all their 9s, the expressions Mean Time Before Failure (MTBF) and Mean Time To Repair (MTTR) are coined. These are probabilistic measures, so they have to be multiplied by the number of instances we have. In today's massive-scale environments, what used to be a small chance of failure or MTBF gets amplified into a large one. To counter this, we need speedy repair if we are going to keep our promises.

elements with new material properties, just by understanding how the individual types of atoms with their different properties (i.e., promises to behave) combine.

This is a bottom-up strategy. When you work from the top down, your whole viewpoint is nonlocal, or distributed. You are not thinking clearly about where information is located, and you might make assumptions that you have no right to make; for example, you might effectively make promises on behalf of agents you don't control.

On the other hand, when you work from the bottom up, you have no choice but to know where things are because you will need to document every assumption with an explicit promise. Thus, a promise approach forces a discipline.

Isn't this just an awkward way of talking about requirements? Not really. It is the opposite. A requirement is an obligation from a place of high-level generalization onto a place of more expert execution. There is an immediate information gap or disconnect between requirer and requiree. The important information about the likely outcome is at the wrong end of that gap. From a promise viewpoint, you force yourself to think from the point of execution and place yourself in the role of keeping the promise, confronting all the issues as they appear. It is much harder to make unwarranted assumptions when you do this.

Thinking in promises also makes you think about contingency plans. What if your first assumptions fail?

The promise position is an extreme position, one that you might object to on some grounds of convention. It is because it is an extreme position that it is useful. If we assume this, we can reconstruct any other shade of compliance with outside influence by documenting it as a promise. But once we've opened the door to doubt, there is no going back. That's why this is the only rational choice for building a theory that has any predictive power.

The goal in Promise Theory is thus to ensure that agents cooperate by making all the promises necessary to collectively succeed. A magical onlooker, with access to all the information, would be able to say that an entire cooperative operation could be seen as if it were a single organism making a single promise. How we coax the agents to make promises depends on what kinds of agents they are. If they are human, economic incentives are the generic answer. If the agents are programmable, then they need to be programmed to keep the promises. We call this *voluntary cooperation*. For humans, the economics are social, professional, and economic.

Is this crazy? Why not just force everyone to comply, like clockwork? Because that makes no sense. Even a computer follows instructions only because it was constructed voluntarily to do so. If we change that promise by pulling out its input wires, it no longer does. And, as for humans, cooperation is voluntary in the sense that it cannot be forced by an external agent without actually attacking the system to compromise its independence.

Deal with conflicts of intent

If all agents shared the same intentions, there would not be much need for promises. Everyone would get along and sing in perfect harmony, working towards a common purpose. The fact that the initial state of a system has unknown intentions and distributed information means that we have to set up things like agreements, where agents promise to behave in a certain way. This is what we call orchestration.

But what if promises in different locations affect a third party unwittingly? This happens quite a lot, as an emergent effect. In obligation theories (requirements, laws, and distributed permission models), the possibility for conflict is very high. Promise Theory is rather good at resolving conflicts because an agent can only conflict with itself, hence all the information to resolve the conflicts is located in the same place.

Just Make It Happen

Promise Theory seems upside down to some people. They want to think in terms of obligations. A should do B, C must do D, and so on. But apart from riling a human being's sense of dignity, that approach quickly leads to provable inconsistencies. The problem is that the source of any obligation (the obliger) is external to the agent that is being obliged. Thus if the agent is either unable or unwilling to cooperate (perhaps because it never received a message), the problem cannot be resolved without solving another distributed cooperation problem to figure out what went wrong! And so on, ad nauseum. (One begins to see the fallacy of trusting centralized push models and separate monitoring systems.)

Promise Theory assumes that an agent can only make promises about its own behaviour (because that is all it is in control of), and this cuts through the issues surrounding distribution of information, ensuring that both the information and resources needed to resolve any problem are local and available to the agent. In this way, an agent can autonomously repair a promise by taking responsibility. This is the meaning of agency.

An Exercise

Test yourself on your ability to think in terms of promises instead of desires, requirements, needs, and so on. Spend a whole day thinking about the promises made by people, places, processes, and things:

- To whom are the promises made?
- In what form are the promises made?
- Do they depend on something else to help them keep their promise?
- How do these things try to keep their promises?
- How do you assess their success?

Don't miss anything: from the bed you get out of (did it give you back pain?), to your morning exercise regimen (will it reduce fat?), the food you eat (is it fresh, tasty?), the people you work with (what are their roles?), the city you live in, all the way up to the toothbrush you end your day with.

If you can't see any promises, try asking yourself: what is the intended function? What is my relationship with these things? What value do I see in them? Finally, what additional interpretations do you add, which are not part of the promises around you? Does money mean lifestyle, recreation, savings for a rainy day?

At the end of your day, you will have a better understanding of what we call *semantics* and *intentionality* in the world, and you will be ready to apply that thinking to all kinds of situations.

With a License to Intend

Intentionality is that elusive quality that describes purpose. It is distinctly human judgement. When we intend something, it contributes meaning to our grand designs. We measure our lives by this sense of purpose. It is an intensely sensitive issue for us. Purpose is entirely in the eye of the beholder, and we are the beholders.

An Imposition Too Far

Throwing a ball to someone, without warning, is an imposition. There was no preplanned promise that advertised the intention up front; the ball was simply aimed and thrown. The imposition obviously does not imply an obligation to catch the ball. The imposee (catcher) might not be able to receive the imposition, or might be unwilling to receive it. Either way, the autonomy of the catcher is not compromised by the fact that the thrower attempts to impose upon it.

The thrower might view the purpose of the imposition as an act of kindness, for example, inviting someone to join in the fun. The recipient might simply be annoyed by the imposition, being uninterested in sport or busy with something else.

In the second part of Figure 2-1, the two players have promised to throw and catch, perhaps as part of the agreement to play a game. Was this a rule imposed on them? Perhaps, but in that case they accepted it and decided to promise compliance. If not, perhaps they made up the “rule” themselves. A rule is merely a seed for voluntary cooperation.

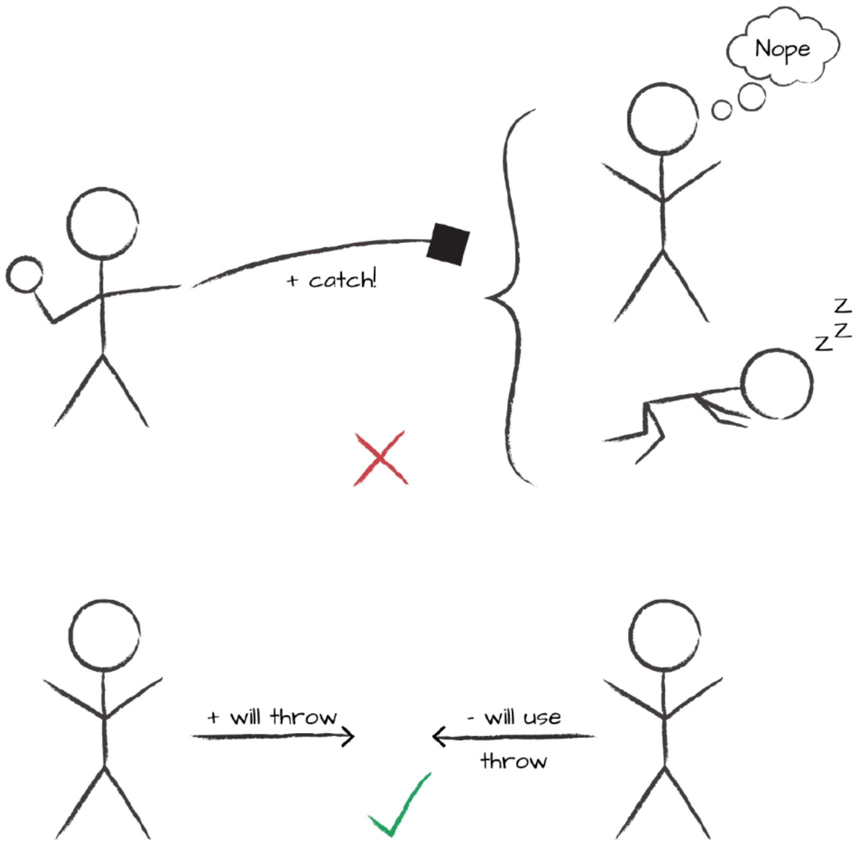


Figure 2-1. An imposition does not imply coercion. Autonomous agents still participate voluntarily. On the right, a lack of a promise to respond might be an unwillingness to respond or an inability to respond.

Reformulating Your World into Promises

Seeing the world through the lens of promises, instead of impositions, is a frame of mind, one that will allow you to see obvious strengths and weaknesses quickly when thinking about intended outcomes.

Promises are about signalling purpose through the language of intended outcomes. Western culture has come to use obligation and law as a primary point of view, rather than autonomous choice, which is more present in some Eastern cultures. There is a lawmaking culture that goes back at least as far as the Biblical story of Moses. Chances are, then, that you are not particularly used to thinking about everyday matters in terms of promises. It turns out, however, that they are absolutely everywhere.

Here are some examples of the kinds of statements we may refer to as promises:

- I promise you that I will walk the dog.
- I promise you that I fed your cat while you were away.
- We promise to accept cash payments.
- We promise to accept validated credit cards.
- I'll lock the door when I leave.
- I promise not to lock the door when I leave.
- We'll definitely wash our hands before touching the food.

These examples might be called something like *service promises*, as they are promises made by one agent about something potentially of value to another. We'll return to the idea of services promises often. They are all intended outcomes yet to be verified.

Proxies for Human Agency

Thanks to human ingenuity and propensity for transference (some might say anthropomorphism, if they can pronounce it), promises can be made by inanimate agents too. Inanimate objects frequently serve as proxies for human intent. Thus it is useful to extend the notion of promises to allow inanimate objects and other entities to make promises. Consider the following promises that might be made in the world of information technology:

- The Internet Service Provider promises to deliver broadband Internet for a fixed monthly payment.
- The security officer promises that the system will conform to security requirements.
- The support personnel promise to be available by pager 24 hours a day.
- Support staff promises to reply to queries within 24 hours.
- Road markings promise you that you may park in a space.

- An emergency exit sign promises you a way out.

These are straightforward promises that could be made more specific. The final promise could also be restated in more abstract terms, transferring the promise to an abstract entity, "the help desk":

- The company help desk promises to reply to service requests within 24 hours.
- The weather promises to be fine.

This latter example illustrates how we transfer the intentions of promises to entities that we consider to be responsible by association. It is a small step from this transference to a more general assignment of promises to individual components in a piece of technology. We abstract agency by progressive generalization:

- Martin on the front desk promised to give me a wake-up call at 7 a.m.
- The person on the front desk promised to give me a wake-up call at 7 a.m.
- The front desk promised to give me a wake-up call at 7 a.m.

Suddenly what started out as a named individual finds proxy in a piece of furniture.

In a similar way, we attach agency to all manner of tools, which may be considered to issue promises about their intended function:

- I am a doctor and I promise to try to heal you.
- I am a meat knife and promise to cut efficiently through meat.
- I am a logic gate and promise to transform a `TRUE` signal into a `FALSE` signal, and vice versa.
- I am a variable that promises to represent the value 17 of type integer.
- I am a command-line interpreter and promise to accept input and execute commands from the user.
- I am a router and promise to accept packets from a list of authorized IP addresses.

- I am a compliance monitor and promise to verify and automatically repair the state of the system based on this description of system configuration and policy.
- I am a high availability server, and I promise you service delivery with 99.9999% availability.
- I am an emergency fire service, and I promise to come to your rescue if you dial 911.

From these examples we see that the essence of promises is quite general. Indeed such promises are all around us in everyday life, both in mundane clothing as well as in technical disciplines. Statements about engineering specifications can also profitably be considered as promises, even though we might not ordinarily think of them in this way.

Practice reformulating your issues as promises.

What Are the Agencies of Promises?

Look around you and consider what things (effectively) make promises.

- A friend or colleague
- The organization you work for
- A road sign
- A pharmaceutical drug
- A table
- A nonstick pan
- A window
- A raincoat
- The floor
- The walls
- The book you are reading
- The power socket

What Issues Do We Make Promises About?

What might be the subject of a promise? Outcomes of different kinds:

- A function or service provided
- A value judgement (say about fitness for purpose)
- Access or permission to use something
- Behaviour (rules, laws)
- Timing
- Location
- Layout or configuration

What Things Can Be Promised?

Things that make sense to promise are things we know can be kept (i.e., brought to a certain state and maintained).

- States, arrangements, or configurations, like a layout
- Idempotent operations (things that happen once): delete a file, empty the trash
- Regular, steady state, or continuous change: constant speed
- An event that already happened

What Things Can't Be Promised?

We already said that the basic rule of autonomy is that an agent cannot make a promise about anyone or anything other than itself. This is a simple rule of thumb. If it did, another agent assessing the promise would be right to reject that promise as a breach of trust and devalue the promiser's reputation, as it is clear speculation.

For example, a manager might try to promise that her team will deliver a project by a deadline. If she is honest, she will make this conditional on the cooperation of the members of her team, otherwise this is effectively an imposition on her team (with or without their knowing). It might even be a lie or a deception (the dark side of promises). Without promises from each of her team members, she has no way of knowing that they will be able to deliver the project by the deadline. If this seems silly, please think again. The aim of science is for realism, not for unrealistic authority. Impositions do not bring certainty; promises that we trust are a kind of best guess.

Are there any other limitations?

Changes that are relative rather than absolute don't make much sense as promises—for example, turn left, turn upside down. Technically, these are imperatives, or nonidempotent commands. Each time we make the change, the outcome changes, so we can't make a promise about a final outcome. If we are talking about the action itself, how do we verify that it was kept? How many times should we turn? How often?

The truth is, anyone can promise anything at all, as in Monty Python's sketch "Stake Your Claim!", where contestants promise: "I claim that I can burrow through an elephant!" and "I claim that I wrote all of Shakespeare's plays, and my wife and I wrote his sonnets." These things are, formally, promises. However, they are obviously deceptions, or outright lies. If we lie and are found out, the value of our promises is reduced as a consequence.

The impartial thing to do would be to leave this as a matter for other agents to assess, but there are some basic things that nature itself promises, which allow us to draw upon a few rules of thumb.

A minimum requirement for a promise might be for there to exist some kind of causal connection between the promiser and the outcome of a promise in order to be able to keep it (i.e., in order for it to be a plausible promise).[1] So it would be fine to promise that you fed someone's cat, but less plausible to promise that you alone created the universe you were born into.

The Lifecycle of Promises

The promise lifecycle refers to the various states through which a promise passes, from proposal to the end (see Table 2-1). The lifecycle of a promise may

1 Statisticians these days are taught that causation is a dirty word because there was a time when it was common to confuse correlation with causation. Correlation is a mutual (bidirectional) property, with no arrow. Causation proper, on the other hand, refers to the perceived arrow of time, and basically says that if A precedes B, and there is an interaction between the two, then A might contribute to B, like a stepping stone. This is unproblematic.

now be viewed from either the perspective of the agent making the promise (Figure 2-2), or from the perspective of the promisee (Figure 2-3), or in fact any another agent that is external but in scope of the promise.

Table 2-1. The lifecycle of promises

Promise State	Description
proposed	A promise statement has been written down but not yet made.
issued	A promise has been published to everyone in its scope.
noticed	A published promise is noticed by an external agent.
unknown	The promise has been published, but its outcome is unknown.
kept	The promise is assessed to have been kept.
not kept	The promise is assessed to have been not kept.
broken	The promise is assessed to have been broken.
withdrawn	The promise is withdrawn by the promiser.
expired	The promise has passed some expiry date.
invalidated	The original assumptions allowing the promise to be kept have been invalidated by something beyond the promiser's control.
end	The time at which a promise ceases to exist.

Once a promise is broken or otherwise enters one of its end states (invalidated, expired, etc.), its lifecycle is ended, and further intentions about the subject must be described by new promises.

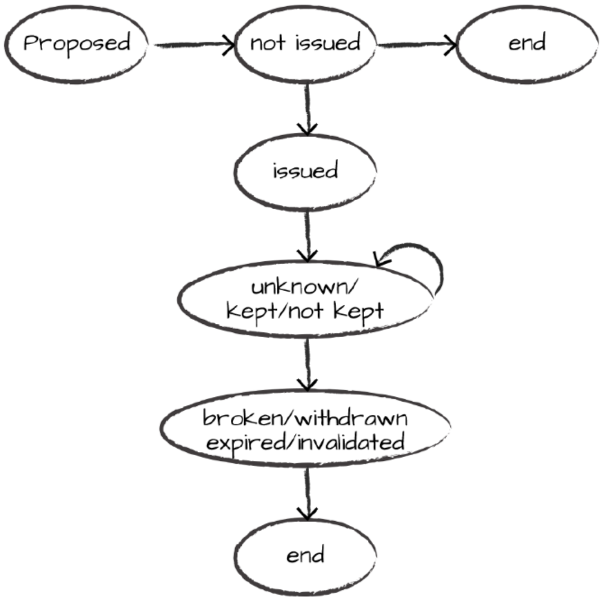


Figure 2-2. The promise lifecycle for a promiser.

From the perspective of the promisee, or other external agent in scope, we have a similar lifecycle, except that the promise is first noticed when published by the promiser (Figure 2-3).

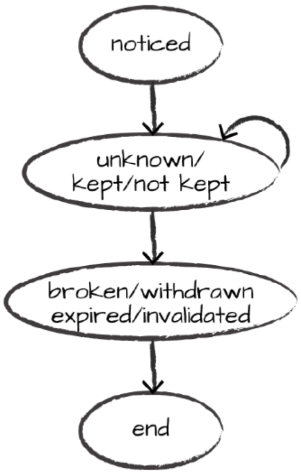


Figure 2-3. The promise lifecycle for a promisee.

Keeping Promises

What does it mean to keep a promise? What is the timeline? When is a promise kept? How many events need to occur? What needs to happen? What is the lifecycle of a promise? Probably it means that some essential state has changed or been preserved in an agent's world, or its state has been maintained or preserved. The configuration of the world measures the outcome of a promise. The result also has a value to the agent. These things are assessments to be made by any agent that knows about the promise. Agents can assess promises differently, each based on their own standards. What one agent considers as a promise kept might be rejected by another as inadequate.

Cooperation: The Polarity of Give and Take

When promises don't go in both directions, a cooperative relationship is in danger of falling apart and we should be suspicious. Why would one agent be interested in the other agent, if not mutual intent? This is a sign of potential instability. This seems initially trite and a human-only limitation, but even machinery works in this way. Physics itself has such mechanisms built into it.

In reality, promises and impositions are always seen from the subjective vantage point of one of the autonomous agents. There is not automatically any "God's-eye view" of what all agents know. We may call such a subjective view the "world" of the agent. In computer science, we would call this a distributed system. Promises have two polarities, with respect to this world: inwards or outwards from the agent.

- Promises and impositions to give something (outwards from an agent)
- Promises and impositions to receive something (inwards to an agent)

In the mathematical formulation of promises, we use positive and negative signs for these polarities, as if they were charges. It's easy to visualize the differences with examples:

- A (+) promise (to give) could be: "I promise you a rose garden," or "I promise email service on port 25."
- A (-) promise (to use/receive) could be: "I accept your offer of marriage," or "I accept your promise of data, and add you to my access control list."