Notes on Incorporating Operational Trust Design into Automation v0.1
(Companion to code examples)

Mark Burgess

June 5, 2023

Abstract

This is not a scientific paper. It is the first part of a set of notes describing and illustrating a proof of concept for using the Promise Theory of trust to i) trace, and ii) adapt online interactions based on learning. Trust plays a role in human behaviour, and here we consider how it operates as a summary potential for human-machine interactions (including implicitly behind the scenes of large machine learning models). It accompanies a set of code stub examples that focus specifically on cases in which users interact with one another through a third party service.

The goal here is to investigate how we might use trust as a guiding potential in human-information systems. For the semantic elaboration, the specific example of users interacting with Wikipedia to read and to write contributions is used for concreteness. They should be viewed in parallel with the example code at https://github.com/markburgess/Trustability.

Contents

1 Introduction

In previous work [1–4], I proposed a two part model of trust and how it might work in human-machine systems, based on a self-consistent Promise Theory. This document is about the details of what trust does in such systems, and how we might implement to guide real time adaptation of behaviours.

The Internet and all of its services and devices augment ordinary society with a landscape of public and private spaces (websites and services) and commons (e.g. Wikis). The behaviours of agents, in the roles of client and server, are increasingly subtle and there is a melting pot of cultures and criminality alongside commerce and philanthropic activities. All this leads to the model of semantic spacetime previously described [5].

1.1 Trustworthiness as a potential landscape

The two parts of trust, discussed in [2,4], represent coarse potentials for guiding the interactions of agents. Potential trust, or trustworthiness, is a prior assessment formed by learning from possibly multiple sources of information. It summarizes whether agents relatively speaking have the intent to keep their promises, and determines whether or not new promise would be accepted from an agent, or whether existing promises should be discontinued. It forms a potential landscape which guides agents to select between alternatives based on their assessed reliability. Kinetic trust is a policy for attentiveness during an established relationship. It allows agents to save on the overhead of monitoring when relying on others. Mistrusting leads to heightened attention to detail. Trusting is the cheapest option in which an agent pays little attention to the delivery while expecting a non-urgent or causal benefit from another. For more urgent and costly fundamental reliances (high impact), agents consider the matter of the risk of promises not being kept.

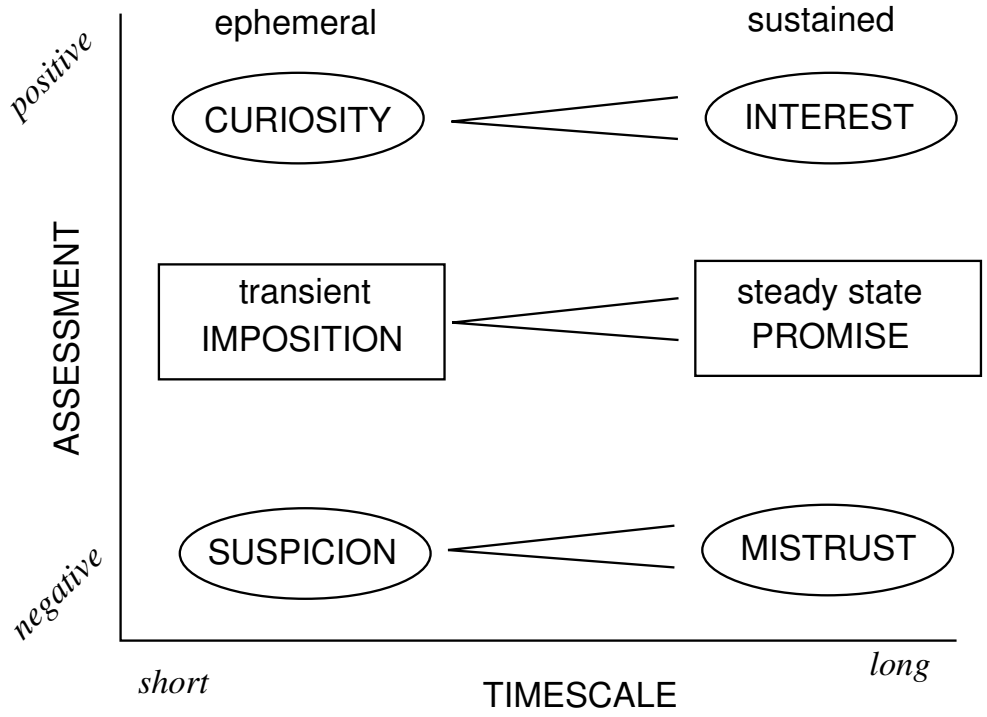

Figure 1: The general shape of the semantic layout for time and orientation for attention based processes. As the timescale for attention increases from singular transients to sustained steady state behaviour, our semantic interpretations change—and we thus use different words for essentially a rescaled version of the same phenomenon.

1.2 Stimulation begets a variable attention response

An agent can direct attention towards a process autonomously, without exterior stimulus, or it may respond to a sensory stimulus event. Trust is one of a family of attention semantics, which may actually be indistinguishable in their practical outcomes and resource allocations, but which differ in their meta information. Other states of heightened anticipation include interest, curiosity, study, hope, worry, anxiety, etc. The differences relate partly to their timescale of the process they refer to (see figure 1).

Stimulation (by source) and attention (by receiver) are processes relating to an information channel, so we are in the realm of information transfer or data pipelines pertaining to active processes. We can imagine a progression in attentiveness from noticing a single event to a sustained process of inquiry, something like this:

$$\text{ANOMALY} \mapsto \text{CURIOSITY} \mapsto \text{SUSPICION} \mapsto \text{DOUBT} \mapsto \text{MISTRUST} \tag{1}$$

In short, we crave certainty or mistrust uncertainty, but it's typically in short supply, so we seek it out or perform *searches* (including on Google). Our cognitive faculties have allowed us to make do with a rough estimator called trust, as a steady state policy to forego attentiveness (which is expensive). We say that someone or something is worthy of trust if we assess that they tend to keep their promises. If they do that, then we don't need to keep watch. It's a simple idea that works well some of the time.

There are two extreme positions for simple observables:

- *Low trust*: this means we choose to verify a lot, which has a monitoring and interpretation cost. One can adjust the time interval of sampling, like a homeostatic adaptation (increased pulse and awareness during stress, etc) to try to optimize the sampling but this tends only to work for very simply responses.
- *High trust*: this means don't pore over every moment with a magnifying glass, but create systems that will ensure continuity regardless of what happens. This usually means redundant backup or failover services ("hot spares" etc). This has its own cost and is a quite different matter to assess.

Trading the costs of these two approaches to saving cost versus winning new revenues from a promise outcomes may not be an easy choice, but it needs to be part of system design. The difficulty arises where observables are not simple. like this, and there is a complex semantic story behind the decision.

1.3 A partial proof of concept

Starting with the assessment of other agents' behaviours, we can create the beginnings of a simple mode of independent assessment for two part trust, which can guide an agent in deciding whether to be involved with another in a promise relationship. This is the potential half of trust, which can always be assessed. The determination of kinetic trust may be subject to channel constraints for Internet services so we must defer a full treatment until later.

The use cases for trust are potentially unlimited, so we need something concrete to get started with, and to limit the scope of the study. While the ideas are straightforward, there's a lot of subtlety so we look for something simple and easily recognizable, with a high impact. Let's mention a few cases before selecting one to work on.

Example 1 (Trust in Wikis) *Wikipedia has a fraught trust model. It's particularly vulnerable to tussles over editing of articles, so the custodians of the platform are cautious. It's not uncommon to find that authors have an agenda, beyond simply teaching or informing. The "talk" pages of Wikipedia contain much vitriol and personal animosity characteristic of online communication. We need to separate style from substance however; the tech world is famous for its lack of manners, but might still be acting in good faith. This is a challenge for trust.*

There are many unwritten rules that are imposed by Wikipedia post hoc and basically ad hoc, e.g. no one "close to the subject" can write about it. This may lead to people hiding and disguising their identities in order to make changes. The editors (called moderators) are extremely suspicious. Large edits are not allowed and quickly get undone. They have to be broken up into small changes, ostensibly to enable peer review. This is not always realistic, since most articles are not widely followed by an army of peer reviewers. There are rules about what kind of citations are allowed. This leads to authors mistrusting the moderators too, and slanging matches of "talk" ensue. As a public commons, it's not clear that there is anyone who can have the last word, but the moderators are powerful and often political. There are many bots involved in the monitoring (and making) of changes. The infamous "warning boxes" about integrity

and quality of articles are unevenly applied. No one would check an article about quantum physics, for instance, but an article about a topical subject would be ruthlessly micromanaged.

- *Size of a change.*
- *Identity of author.*
- *Frequency of edits.*
- *The type of language used.*
- *Bot identified issues, e.g. citations and rankings.*

As an ecosystem of bots and human contributions, Wikipedia is an interesting testing ground.

Example 2 (Trusting research–the usual problem) *The Internet preprint archive, arXiv, used to trust all authors to post papers on its site. There was no gatekeeping or selection, everything was taken. Then gradually over time the conditions changes, this trust was abused perhaps, and today there is extensive moderation or censorship under the auspices of ensuring quality. How is quality determined? Usually some cheap manpower (graduate students and others) are persuaded to vet the submissions. The problem here is that they are the ones who don't have the experience to judge the content of papers well, so criteria like host affiliation and identity are used in practice. A free license is given to "members" of the "rich club"*[1]*, and those who are outside are profiled and discriminated. This is the same problem of prejudice that afflicts gatekeeping at all levels of society. Researchers believe themselves to be rational. but rationality is only used to later justify decisions made by coarse grained criteria.*

Example 3 (Trusted Third Party services) *The question of trust for websites began with identity fraud, so TTPs like Verisign emerged which established a repository of signatures for HTTPS, verified with a credit card, for confirming identity cryptographically. This only validates the identity of the web connection, or server. There is a plausible chain of trust: credit card companies can be trusted to mistrust and validate identity etc.*

Today, content trust sites which collect user reviews and which give promise-keeping star ratings (e.g. TrustPilot). Users basically never get negative reviews, however, as these are filtered. Reviews are trivial to forge, poison, and bias. These sites are largely useless, a natural part of the arms race to establish service trustworthiness as a service.

In practice, choosing a provider is not very hard.

1. Giving the benefit of the doubt, looking for serious players.
2. More expensive often means better quality.
3. Popular providers are typically "good enough but not amazing", because quality cost cutting is their business strategy.

Example 4 (Cryptocurrency and zero trust) *BitCoin and crypto-currencies were based on a political interpretation of trust as a vector to make people identify as "victims of power" for the purpose of exploitation. Crypto activists decided that banks were untrustworthy and even argued that trust itself was harmful. The zero trust rhetoric has itself been a harmful diversion, which has held back progress in understanding for many years, and is based on a factually inaccurate portrayal.*

Distributed assessment is an outsourcing of mistrust to the higher level collective. Each individual needs to trust the integrity of the group in order to verify frequently (antitrust). As the number of agents grows, the speed of trust consensus also gets longer and more trust is required. Users are thus united in mutual trust by their mistrust of banks, replacing trust in institutions with trust in a system of distributed assessment—i.e., the software and integrity of the consensus algorithm.

The system is quite slow and therefore kinetic trust in it had to be maintained at the highest possible frequency on the timescale of the process's own clock (which is still slow for many others), because each distributed sample took a long time to complete. Over time this trustworthiness has faded for some and committed users choose to trust the platform as they see no point in watching over something beyond their control.

[1] Rich club is a hub concept from network science, which observes that well connected nodes tend to cluster.

Example 5 (FPGA chips) *FPGA are volatile software programmable chips. By programming hardware, some acceleration can be achieved by parallelism. Trust in FPGA is the same as trust in any other network service, as it behaves like a network device. Communication is through a wire protocol like UDP, so this case reduces to being essentially the same as a cloud service. The FPGA promises a certain version of software and chip manufacturer.*

Example 6 (Latest versions semantics, outsourcing dependency) *A common promise to make is to tell users they are getting the latest and the greatest. Since this is vague, it's easy to accept (coarse graining hypothesis) perhaps because it may be difficult to verify. Scanners and test kits may exists to verify the claim, e.g. like virus scanners in IT. Cloud services, databases, and software systems may make simple promises about speed or work capacity, but one is expected to trust the software and configuration.*

This is made worse in distributed systems, where we want homogeneity across a wide area, Network outages may prevent equilibration of

Each of these cases has interesting points with similar issues expressed in different ways and with differing levels of accessibility. The Wikipedia case has all the elements of spacetime process that we need, and does not require special access to work on. So, let's use it is a passive mode to see how far we can get in using the theory to develop a model based on two-component trust.

1.4 A trust user manual

We want to make sure our formalized version of trust captures the ways we use trust in human life. We assume that we can divide the process into two parts:

- The first part of our model is a determination of trustworthiness is usually made on the simplest confirmation of identity and therefore relies on a policy of preconfigured static access control to allow or exclude certain agents by name. At this point an agent decides whether or not to engage with another's promise offer.
- The second part is kinetic trust. Real-time monitoring systems can only adjust the level of mistrust once a relationship is underway. So the data structures in a monitoring channel, which could adjust our willingness to provide or accept certain services are mostly ignored, because there is no way to revoke credentials once trusted by identity alone.

In simple pragmatic terms, we have two questions:

- Should I have this conversation at all? (potential trustworthiness)
- How carefully should I manage the interaction? (kinetic attentiveness)

Once a relationship has begun, one wants to construct an automated feedback circuit that can act to maintain a steady state of behaviour and adjust its attention using a concept of trust for any agent. An agent's assessment includes both quantity and quality (measure and semantics). A rational assessment assumes we know what was promised, however monitoring systems typically don't generally know that, and thus try to look at generic and implicit measures that are only peripheral to the actual promised outcome. It leads to a guessing game (this is captured by formal game theoretic models).

We need to place a value on what has been delivered $b^{(-)}$ relative to its intersection with our own level of acceptance $b^{(-)}$. Identifying these values (provisionally) will be the motivation behind the remainder of this document.

2 The stages of trust in a promise lifecycle

We begin with a decision whether or not to interact with another agent at all, i.e. to rely on its services, delegate, or depend on in some way. When we say, I don't trust X so I won't have anything to do with X, we mean that we assess them as untrustworthy and never engage in a promise relationship.

It's helpful to relate some cases in which the different parts of trust play a role in decisions. Consider the promise interaction in figure 2 once again. The stages of a relationship for 'knowing another agent'

Assigning zero trust would imply checking the promise channel infinitely often, which is clearly not right. The allocation of kinetic trust does not take place unless there is sufficient trustworthiness to form the foundation for a relationship

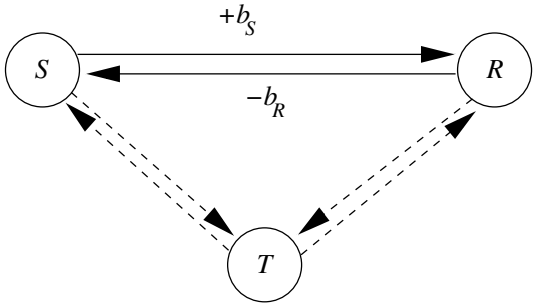

Figure 2: The prototypical three body building block for trust..

1. Trustworthiness is bootstrapped. We may inherit experiences about other unrelated promises and other agents to begin with: according to the coarse graining hypothesis we tend to profile generalities rather than specifics. This comes from a policy of trusting cheap information.

2. Starting with either an agent's initial memory state or experience, "beliefs", and hearsay reputation, an agent R bootstraps an assessment of another agent S, with trustworthiness $V(S)$, with or without prior experience of the agent.

3. The agent R may or may not form a promise relationship, accepting a promise offered by S, based on this assessment, i.e. R will only accept promises from agents with sufficient initial trustworthiness.

4. Once the promise has been accepted, R will allocated an initial kinetic (anti)trust $\overline{T}(S)$, based on its assessment of trustworthiness.

5. As R samples the promise-keeping of agent S over multiple events, it updates its assessments of $V(S)$ and $\overline{T}(S)$.

   The assessment of trustworthiness $V(S)$ is related to the belief of reliability of the agents and is not dependent on R's condition.

   The determination of a trust policy $\overline{T}(S)$ is, however, based on the condition of R and its resources for limited attention.

2.1 How to use Kinetic trust - benefit of the doubt

Example 7 (No peeking) *S intends for R to trust, i.e. 'look away' or 'mind its own business'. S wants R to forego monitoring regardless of whether R believes S to be trustworthy (it might have a large uncertainty–just don't know about S), so it wants R to give it the benefit of the doubt. This could be a way to sneak past R's security, or it could be a good natured 'relax and enjoy yourself' intention.*

Example 8 (No time wasting) *No one wants to waste others' time. We don't want to stop the traffic to cross the road, or hold up boarding to check passports, but this is part of the promise keeping. We invent rules that agents promise to follow (stop on red, keep your passport ready). Now R can trust that S will be prepared and promise follow these with a high probability, so it can afford to be less diligent in its checking process. This protocol of following rules might be of benefit to one or both of the agents, even though rules are always simplistic "the law is an ass" for doing the stupid heavy lifting (i.e. a donkey, not the American vernacular meaning).*

Example 9 (Workarounds and quick fixes) *Temporary fixes and workarounds in services or systems could be assessed as positive or negative. In a promise from S to R*

$$S \xrightarrow{+b} R, \tag{2}$$

S might make a new promise of a patch to b in order to to fix a deficiency identified by R. We could write this as

$$S \xrightarrow{+(b+\Delta b)} R, \tag{3}$$

or as two promises, where they are co-dependent:

$$S \xrightarrow{+b|fix} R, \tag{4}$$

$$S \xrightarrow{+fix|b} R. \tag{5}$$

R may assess this change as trust building, i.e. an alibi to reduce its fault finding, or it might assess it as wanting to avoid fixing the actual issue, which makes it increase its watchfulness. It indicates that the assessments of agents are not necessarily predictable, especially in relation to the intention of the other party.

Example 10 (Hedging risk) *A hedge against loss is to mitigate a bad outcome is like a workaround in the previous example. The additional promise of a hedge suggests that we've plugged a leak or covered ourselves to make the worst case outcome survivable. Then might at well go about your business because endlessly watching and "hoping" won't make it better. No need to watch too closely, we're got your back.*

The purpose of monitoring in each of these cases is often that an agent could know when to cut its losses and walk away from a promise binding.

2.2 How to use Kinetic antitrust—impositions and transients

Antitrust is often stimulated or provoked by impositions, as these are destabilizing events or transients rather than a stable steady state response.

1. Suppose S makes an imposition to R, i.e.

   $$S \xrightarrow{+b} \blacksquare\ R \tag{6}$$

   with some request, demand, or accusation which is out of the ordinary realm of expectation

   $$R \xrightarrow{-b} S. \tag{7}$$

   This may cause R to lose trust and increase its watchfulness to look for further anomalies. S might benefit from this or lose. R might be able to cut its losses by breaking off contact, but the economics of such promises depend on a larger network, even though the focus may be on a single issue. This is the downside of the coarse graining hypothesis.

2. Sleight of hand, or use of a promise infraction as a decoy or diversion. S behaves so as to induce R to increase its watchfulness about one promise, and thus reduce its watchfulness in other areas, where it could really benefit from observation. This an agent can exploit a tendency to mistrust to misdirect attention

3. Message complexity. Another way S can increase the time spent by R on verifying on verifying a promise is to increase the size or detail of the payload in promise keeping events.

4. Training is attention to knowledge: a strategy to improve one's self-confidence in promise keeping is a way in which agents can use mistrust to impose a cost. External agencies might impose a demand for training, else threaten consequences, etc, or an agent might wish to increase its own capabilities by allocating time to learning, studying, training, etc. In the latter case it's about mistrust in self, or self-confidence with the same effect.

2.3 How to use Potential trustworthiness (honour)

Trustworthiness is an assessment that suggests there is no need to check too carefully. Trustworthiness is a precursor assessment used to bootstrap a policy for mistrust. It has to be bootstrapped from not knowing about an agent or its promises, which itself requires attention. We might call that curiosity in the beginning, but it's the same operational thing. In other words, there's an iteration between trust and trustworthiness used in making assessments—not a simple linear chain of reasoning. A coarse ambient reputation is often the first information we have about trustworthiness.

1. Trust in public servants, police, doctors, etc. If S is the public servant, promising a service then R will first assess by reputation or word of mouth 'ambient feeling'. in individual encounters, trustworthiness will be ratcheted up or down by different degrees on an individual basis. For sparse contact, the initial reputational feeling will tend to dominate. This has a profound effect on society.

2. Infrastructure: road, rail, water, sanitation, etc. These are promises we all rely on. In the developed world, we tend to trust these are only notice them when they are absent. The effect of their absence is very large, however, so mistrust may temporarily increase around the time of a problem, but the general assessment of trustworthiness for these basics is hard to change. Infrastructure has a large inertia to change. Only a protracted failure, perhaps during a strike or natural disaster could alter public perception.

2.4 How to use Potential untrustworthiness—suspicion or curiosity?

When an agent is considered untrustworthy, Promise Theory's Downstream Principle tells us that it pays to have a backup or redundant alternative so that mistrust can be an effective tool for switching promise providers.

1. Our ability to detect deceptions, like fake news, misinformation, rewriting historical facts, lying and making false claims can be applied in either direction in a promise binding. The two interpretations can be subject to assessments of dynamical and semantic reliability:

$$S \xrightarrow{+\text{claim}} R \tag{8}$$

$$S \xleftarrow{-\text{believe you}} R. \tag{9}$$

2. Agents might tolerate untrustworthiness if they have no choice, i.e. offer them the benefit of the doubt for a while, especially if assessment is dogged by uncertainties.

3. An agent S could accuse a third party T of lying to R (hack its reputation) in order to discredit it. This might cause R to alter its assessment of trustworthiness.

3 Bootstrapping trustworthiness from initial seed conditions

In order to enter into a promise relationship, an agent has to make an initial assessment of the trustworthiness of its counterpart. This is a subjective process that by no means leads to a factual result.

- Does the agent's promise even align with my own thinking? If not, no need to invest more time looking. If it does, may want to inspect and come to an acceptance.

- Because we might need to form an assessment of an agent's promise long before we ever encounter the agent, or a member of a superagent group, the 'ambient field' of assessments $V(A)$ will likely shape many if not most agents' initial assessments and bootstrap trust dynamics between them. This is more likely in a steady state network society, less likely in a sparse interaction scenario.

Example 11 (Trustworth) *What do we have to go on when assessing trustworthiness in the beginning, without direct knowledge? Public information (stigmeric traces). An agent's dashing good looks or overt biases may stimulate the curiosity of the assessor, or resonate with its own biases, giving the other a position of power over the assessor. This may be how leadership emerges: by alignment with the assessors receptors, resulting in admiration and a resonance effect.*

There are both qualitative and quantitative assessments that provide different aspects of measurement. Promises acts as a coordinate 'basis' in both quantitative (real valued) and qualitative (discrete) cases. We include:

1. Assessments of agents' general 'type' and 'beliefs'

    - Likes and dislikes registered, e.g. on social media
    - Beliefs expressed and their alignment between S and R
    - Outward signals: clothing, grooming, personality traits.

- Advertising and work output (public relations).

2. Multichannel discrimination of semantics: an agent can extract a list of (sub)topics as linguistic n-grams that describes an agent's directionality (intentionality) in linguistic terms. This symbolization of input patterns is why language models are having some success presently in tools like Chat-GPT. like hash tags.

3. Data on quantitative degree of support for a particular promise type. This allows us to add a magnitude to the overlap of common interests between S and R.

3.1 Simulating human assessment, strengths and weaknesses

Human judgement is not at all like machine assessment. Humans think expansively, and there is rarely the discipline of a clean separation of concerns. This fits with the coarse graining hypothesis, which suggests that generalities will have the largest impact on assessments. So humans will mix and match even vaguely related impressions when assessing trustworthiness. For example, if someone expresses a religious or political view, this might suddenly produce a large change in the assessment of trustworthiness in some scope. If a person is associated with a country whose government misbehaves, that can influence our assessment by association. This mixing is both a *strength* and a *weakness* of human judgement in different contexts.

3.2 Use of monitoring tools

One of the frustrating aspects of monitoring is that we often have no idea what is being promised, or what we should expect, so assessing observation becomes a spurious activity that ends up being based on patterns of hopefulness, or at best 'normal' behaviour. This is both a reason to trust—after all, we may not really know what we want, or even what is being offered, thus we don't know what to look for, and we may confuse not knowing what to look for with the need to look in detail for something that we hope to find.

3.3 Lack of preparation as a basis for (mis)trust

Being prepared can easily be viewed as a prelude to an investment of trust. An agent invests in knowing enough about its counterpart to accept its promise and correctly interpret what happens during the promise lifecycle. Too much trust (neglecting to prepare) may backfire later when the promise is considered untrustworthy because expectations were misaligned.

Whose fault is it that we are unprepared to evaluate the processes in our environment? Curiously, often this unpreparedness is used as an excuse not to trust in human management (a lack of transparency or at least a lack of due diligence leaves us at a disadvantage and without a backup). In this case, I've argued that blame is a common mistrusting response: the allocation of effort to imposing on the provider when all else fails to produce a result.

An agent may make its decision to trust or mistrust based on incomplete information. The response might be to look harder in order to understand, or to back off in rejection of detail. Ill preparedness is coupled with the system 1 assessment of fear about the unknown. In an environment of fear, mistrust may lead to excessive monitoring to no avail. How agents balance their own assessments with those passed on by reputational gossiping may impact significantly on the extent to which an agent trusts and can partake in its environment. This is the lot of the impartial observer.

4 Implementation of assessment concept

We begin by instrumenting some basic code skeletons for establishing a dialogue between agents.

4.1 Stubs

A simple instrumentation has been applied to TCP, UDP, HTTP, and HTML channels between two agents S and R in the example programs. These illustrate different levels at which message data can be interpreted. A number of proof of concept code snippets, with supporting library, can be found in the repository as a first step to developing instrumentation for the different use cases. The first parts show how to instrument the dynamic timelike channel.

- `TCP_client.go` : equipping a client side communication with a latency manager.
- `TCP_server.go` : equipping a server side communication with a latency manager.
- `UDP_client.go` : equipping a client side communication with a latency manager.
- `UDP_server.go` : equipping a server side communication with a latency manager.

The second parts show how to extract sublanguage and use bioinformatic fragmentation to find codon patterns.

- `HTTP_Client.go` : deferred until better understood the significance of HTTP.
- `HTML_client.go` : extraction of sublanguage semantic content from HTML
- `ngrams.go` : scanning of pure language semantic content to establish patterns
- `TT.go` : package code for supporting functions and data structures

4.2 Technical challenge in scanning embedded text

Embedded text occurs in all protocol interactions. The protocol is one language, the payloads may embed another language—like phenotype and genotype. An HTML page is language within language.

- There is the context free structure of HTML and its many extensions, with embedded JavaScript and other encodings, and each of the tagged-and-typed regions contains free text which is the data we seek. Some of this will change over time with styling changes, despite separation of some issues with CSS.
- Only some of the HTML tagging has semantic significance to the subject, most pertains only to the process of rendering of the page in a browser.

We look for generic ways to filter out irrelevant padding and extract data that relates to the process of interacting with other authors and generating material, as well as the trustworthiness of the information itself.

4.3 Quantitative: monitoring machine and service data

These points suggest keeping variables of the following form:

- Any dynamical system is characterized by a state and a rate of change of state (process speed). So we need to keep data values that allow us to compute rates too. The last three measurements allow us to go to second derivatives and accelerations, for curvature, and maxima and minima.
- $Q[timekey]$ this pattern will give the longer term record for coping with different conditions, for the confidence assessment.
- The running average (last 3 values) allows us to measure the rate and acceleration.
- An assessment of symbolic responses amounts to an assessment of quality of service (QoS). How shall we classify such responses. These are usually simply referred to as exceptions or validation errors in software engineering.
  - Search result quality in hits and misses.
  - Recency of data returned, latency and relevance.
  - No of errors in returned data (spelling, grammar, malformation etc)
  - Result never returns (AWOL)
  - The assessment can have meanings beyond the level of service:
    Stopped, abandoned Disconnected Fallback, lagging

How shall we break down a coarse assessment into subpart assessments to increase the resolution? There are many assessments we can make about a promise relationship that would go beyond simple trust dynamics (see under qualitative assessments). These are more complex and potentially difficult to assess. That's probably why trust is useful as a coarse approximation to decision making. So we begin by focusing on this lowest level of granularity (on which the others must depend somehow) and defer other criteria until a later time.

The typical use case for a monitoring system is the "open up the bonnet and look for something that looks out of place" approach. Users generally have no idea what they are looking for. They hope to see something out of place in the forest of trees. This is increasingly unlikely in the age of multi-layer software and virtualization. There is very little correlation between behaviour and the kinds of metrics exposed by whatever layer of kernel users have access to. This is a result of extreme sharing of resources. The level of informational entropy is extremely high (by design).

We can imagine making something like a kernel resource monitor, i.e. a Process Control Block (PCB) to control and keep the state of process execution. Each service agent's interaction is itself a mini kernel for the exchange–we can call it the Service Control Block. This will allocate and assess resource usage in a promise channel. The lesson of Promise Theory is that we must place dynamics and semantics on an equal footing as far as possible. The assessment of the quality of a promise keeping event by an agent will involve more than quantitative comparisons. We may still be able to use machine learning, but with language information and graphical relationships too. Note that this doesn't have to be a network connection. Any interaction between agents (two users, machine and user, etc) will have such a context. We could define it as a part of user experience where humans are involved, for instance.

This process or service control block can monitor the behaviour of the promise process, and by implication the agent on the other end of the promise. We want the memory to be in a form that can easily be coarse grained over multiple behaviours and associated with the channel, so we should keep common identifiers like the IP address and name of the host on which the process is based.

For promises originating from a superagent, such as client-server interactions, a channel is open to public clients then it will most likely fluctuate according to the well-known weekly pattern, so we should memorize by the periodic time key to retain whatever context could be important for measuring reliability and trustworthiness. On top of such variations, there will be variations by time of year too, so we should allow the possibility of making quick adaptation and use different "lenses" for promise and agent assessment.

4.4 Quantitative observability

We need to instrument events on the transport layer and on the content layer. We might not use all these details in every case, but they are simple and systematic to compute. Imagine every service interaction online as a series of events, sampled by the receiver (promisee).

```
type Event struct {
Key    string     `json:"_key"`    // name

Q      float64    `json:"q"`       // some quanitative value for promise body b
Q_av   float64    `json:"q_av"`    // running time average of q, Bayesian style
Q_var  float64    `json:"q_var"`   // running variance of q

T      int64      `json:"lastT"`   // the assessor's timestamp of the previous sample update

Dt_av  float64    `json:"dT"`      // the running average time interval between samples
Dt_var float64    `json:"dT_var"`  // the running variance of time intervals

Units  string     `json:"units"`   // annotation
}
```

These values are enough to assess whether or not a promise is kept (by looking at q) and whether or not events arrived late to too infrequently relative to the sampling rate Dt and its Nyquist frequency.

4.5 Usage by observation context

The temporal aspect of trust is easily captured with a simple context aspect parenthesis. Each agent watches over its side of the connection and assesses the other party. We can create a simple context parenthesis to measure the round trip time for a service event between agents:

```
ctx:= TT.PromiseContext_Begin(g,"tcp_service") // periodigram?

KeepPromise(ctx, ...)

TT.PromiseContext_End(g,ctx)
```

The implementation of this results in a summary like this:

```
Promise duration b (ms) 0.220159 = 0.220159
Running average 50/50 0.0002091255
Change in promise since last sample 22067
Promise derivative b/s 9.142311739322543e-06

Time since last sample (s) phase 2.413722112
Time signal uncertainty dtau (s) group 24.081327160729053
Running average sampling interval 4.191426140640625
```

In addition to this, we need to perform a semantic assessment of the payload. Does it look like the agent was doing an honest job, or does this look like a hack, etc? The function

```
TT.AssessPromiseOutcome(g,"tcp_server",e,AssessResult(string(received)))
```

does this. Note that it calls a user-provided callback to assess the received data. Each client must provide its own classification of the quality of service by returning one of the following constants

```
const ASSESS_EXCELLENT = 1.0
const ASSESS_PAR = 0.5
const ASSESS_WEAK = 0.25
const ASSESS_SUBPAR = 0.0
```

4.6 Scaling and dimensionless variables

The difficulty with arbitrary time series is that the magnitudes of changes are not anchored in a model or expectation. Simply learning values is meaningless unless we can compare the results to an established scale. There are two alternatives here:

- One assumes a steady state process, possibly with calculable trends (e.g. as used in Machine Learning),
- One defines normal by policy (e.g. some dial threshold decision as used in machine monitoring).

The first approach is risky and is generally what gets us into trouble. The second is simplistic but can't be questioned. It's every agent's "sovereign right".

Generally speaking, because promises can be changing, we need to use the current promise as the scale. It is the expression of policy referred to above. So we look for dimensionless ratios to make sense of the scales.

$$q/q(\pi), \Delta q/q(\pi), \Delta t/\Delta t_{\text{sample}}, etc \tag{10}$$

If we don't control the intent, such as when we are on the receiving end of an imposition, then we can only try to learn the other party's intent by assuming a steady state and using statistics as the benchmark. Statistics assumes a steady state, because it's based on long term repeatability. Note however, that this may be questionable and may lead to illusory interpretations and increased uncertainty especially when considering anomalies.

- Whenever we rely on learned statistics to measure trust, we have to ask why we would trust the statistics (data don't lie, but the way we use them may be wrong).
- When we rely on policy levels we know that was defined to be intended.

For trust, we are trying to answer whether or not we should accept what we see from the source process concerned, because we might want to choose a different source in future. Which source is doing a better job at keeping its promise?

Any expression of comparative scales can be extremely sensitive to small changes. The sigmoid function is a common way to bound and regulate the sensitivity in fairly ad hoc ways, but this also requires some fine tuning, which is concerning for the generality of the method of assessment. This will no doubt be an issue we have to return to in the future.

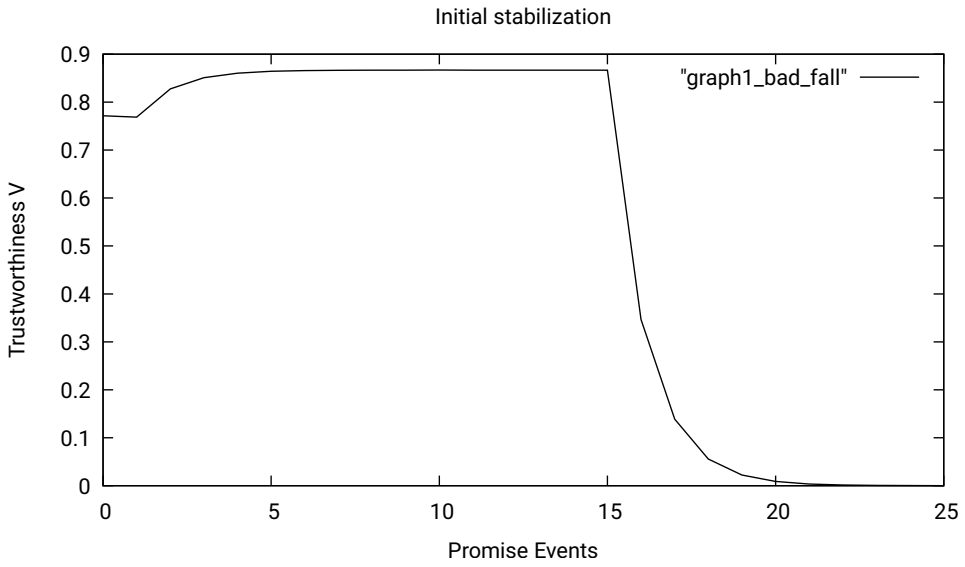

Figure 3: Agent assesses normal promise keeping up to the discontinuity, then a step function in semantic (quality) assessment leads to a sustained period of very bad behaviour makes the trustworthiness fall off to a minimum value.

4.7 Qualitative: language models and symbolic matching

Qualitative assessment is particularly important for bootstrapping a relationship, since we have no regular contact to calibrate quantitative values. Qualitative assessments are based on past learning, or first impressions if there is no past to rely on. For our Wikipedia example case, interactions use text so we need to learn key phrases. The method used in [5] uses essentially a bioinformatic approach to text analysis: breaking units of text down into small pieces on the scale of words (called n-grams) and counting their occurrences. The most common n-grams are padding (spaces and joining words).

Language models are complex hierarchical systems of expression on a generic level, but also ordered sequences of intentional structure on a short term. So we have both long and short term memories involved in this 'cognition'.

We can build lists and relational graph structures to memorize the fragments and their possible compositionality into Semantic Spacetime patterns.

- Sentences become agents. They express their content as an atomic unit of narrative.
- The contents of fragments (which have the status of symbols, i.e. an atomic instance of a proper name) are expressed by each fragment.
- By counting sentences as units of 'proper time', a finite buffer size aggregates sentences into coarse grains of narrative progress called 'legs'. Sentences that score above a certain threshold for acceptance become aggregated into grains, and promise to be part of a superagent called a hub

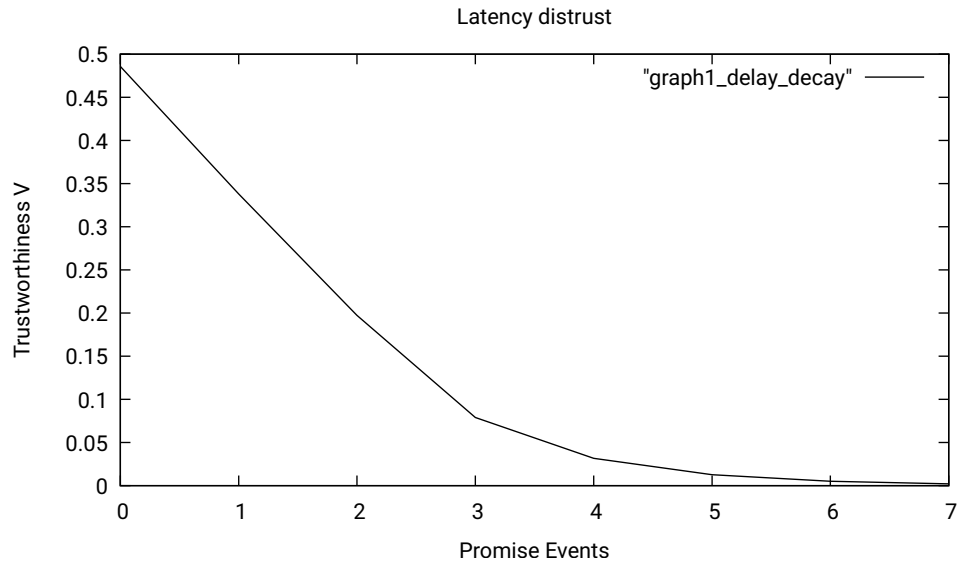

Figure 4: Instead of a change in semantic assessment, there is a progressively larger delay in delivery of the promised outcome.

(denoted H_i). A hub therefore contains sentences, and each sentence expresses multiple ϕ_n fragment attributes. Each hub therefore expresses the sum of those attributes too—which summarizes a *context*.

- Sentences express microscopic ordered combinations of words and phrases ϕ_n. The meaningful sentences S_t 'follow' each other in the proper time order of the narrative (labelled t). Hubs follow one another too when derived from the sentence event order.
- Fragments ϕ_n are contained by larger sequences, which are 'contained' by sentence agents S_t, which are contained by hubs in their respective legs.
- The function of smallest fragments is to match with similar patterns in other sentence agents. The function of longer fragments is to encode uniqueness. Beyond $n = 3$, fragments rarely recur [6]. These ϕ_n become the bodies of ($\pm$) promises to offer and accept information, much as molecular sequences allow binding between cells or polymers.

As a technical issue, we also have to extract text from different embeddings. Languages are embedded in one another, e.g. English within HTML, books, or text messages, all within a hierarchy of taxonomic naming.

We should not imagine that we can reduce human-machine behaviour to a simple utility model, but we can use statistical occurrence as a rough measure of sampling frequency. From there we get a value for trust and a kind of importance ranking [7].

We keep simple associative arrays (maps) for n-grams. These act as sparse histograms. We use a prefix STM for short term memory and LTM for long term memory, to indicate the timescales of usage. Long term structures can be cached in persistent memory.

```
// ****************************************************************************
// The ranking vectors for structural objects in a narrative
// LHS = type (semantic, metric) and RHS = importance / relative meaning
// ****************************************************************************

// n-phrase clusters by sentence are semantic units (no relevant order) - these are memory
// implicated in selection at the "smart sensor" level, i.e. innate adaptation
// about what is retained from the incomning 'signal'

var LTM_NGRAMS_IN_SENTENCE [MAXCLUSTERS]map[int][]string

// inverse: in which sentences did the ngrams appear? Sequence of integer times by ngram
```

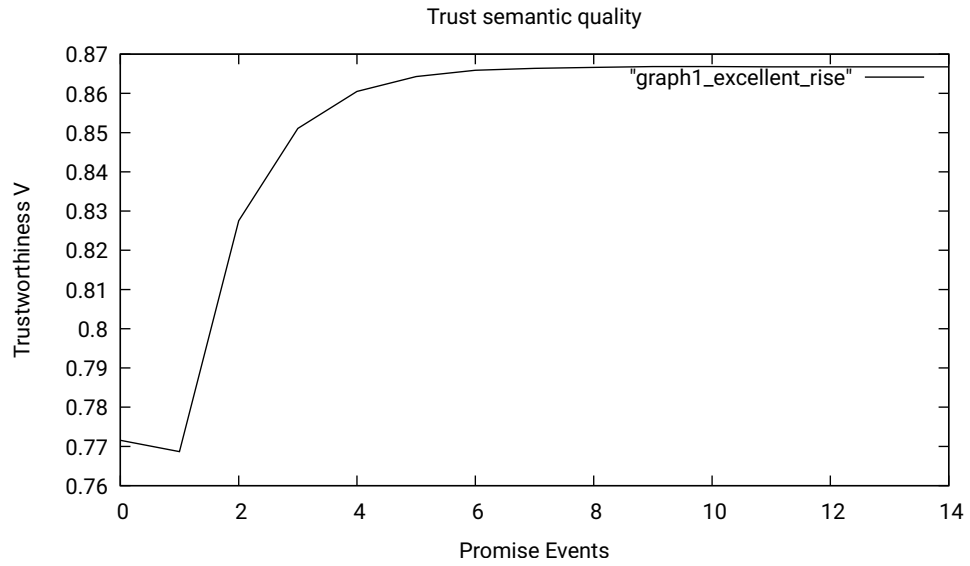

Figure 5: After a period of low assessed trustworthiness, the value recovers quickly to a maximum value after a return to an excellent assessment.

```
var LTM_EVERY_NGRAM_OCCURRENCE [MAXCLUSTERS]map[string][]int

var HISTO_AUTO_CORRE_NGRAM [MAXCLUSTERS]map[int]int  // [sentence_distance]count

// Short term memory is used to cache the ngram scores
var STM_NGRAM_RANK [MAXCLUSTERS]map[string]float64
var LTM_NGRAM_RANK [MAXCLUSTERS]map[string]float64
```

This results in n-gram rankings of the form:

```
ngram  burgess 60
ngram  mark 26
ngram  physics 18
ngram  has 9
ngram  identifiers 13
ngram  knowledge 7
ngram  articles 18
ngram  theory 21
ngram  cite 7
ngram  computer 24
ngram  theoretical 11
ngram  semantic 9
ngram  computer science 10
ngram  identifiers articles 12
ngram  mark burgess 9
ngram  the idea 6
ngram  semantic spacetime 7
ngram  burgess mark 17
ngram  articles with 16
ngram  promise theory 15
ngram  cite journal 6
ngram  system administration 9
ngram  network and system 6
ngram  identifiers articles with 12
ngram  network and system administration 6
ngram  cite journal requires journal help 3
ngram  cite journal cite journal requires 3
```

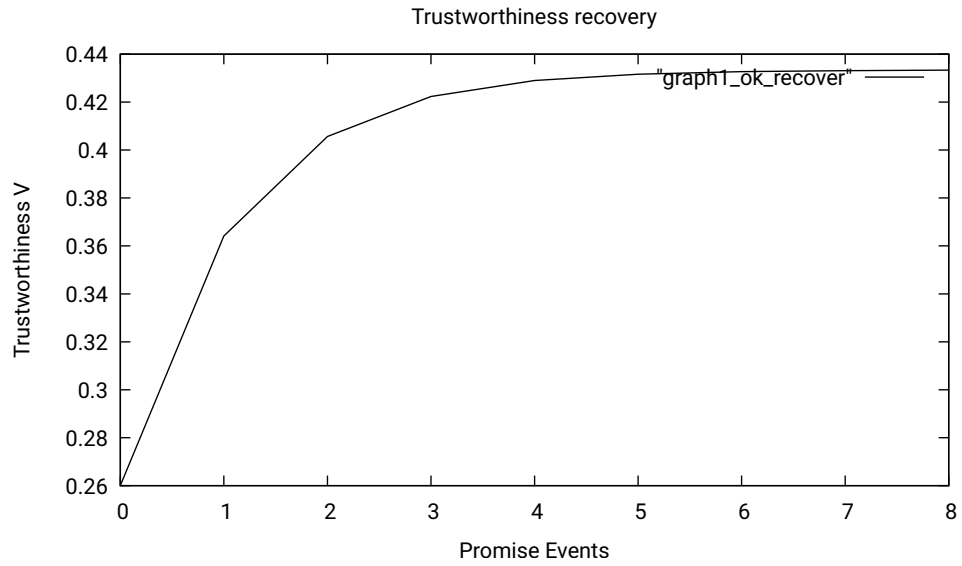

Figure 6: A return to normal promise keeping "good enough" leads to a middling assessment of trustworthiness.

...

From these fragments a semantic assessment can be developed as in [6, 7].

4.8 Machine states and error messages

In IT, different cases are represented by types or exceptions. This is a software engineering way for classifying intent, which derived from the prevalence of taxonomies for knowledge management during the early years of software language design. Types are simplistic because taxonomy was designed to handle far less scalable cases than one experiences in reality; also it represents essentially the intent of the designer, not of the user. To build trust the intent of the user and the designer should align to begin with, and then the resulting promises should be kept with a high fidelity.

The type model deals with the concept of an "error" (now called "exception") to characterize some impediment to progress. A simple rigid type list makes for a simple response. This is the benefit of logic. These are typically designed for software developers rather than for users.

In human interactions and in machine learning states of complexity, we might be lucky to define procedural states and protocols for signalling intent. One trades the brittleness of a type system for the uncertainty of intent in a learned corpus.

Example 12 (Application) *In Wikipedia, several of these exist, thanks to the widespread use of automation to scale the task. Special boxes are added to pages, and messages on chat are labelled "undo" etc. These can be recognized without too much trouble.*

- When a system call registers an error, it fails to keep a promise. The program that depends on it will therefore also fail to keep its promise.
- The risk of poor error handling, crashes, etc is that the program fails to keep its promised function.
- Packet loss in monitoring information delivery, or outcome loss in that outcome fails to arrive on time.
- Reputation can be a probability of keeping promises shared to a trusted third party or peer gossiping protocol. This is hearsay, so there is also a question of trust in the source's intent.

  We can model reputation as a cached value that we do with as we like. In practice, our confidence in this as an assessment has to be compared to self-confidence in making the assessment. These are the uncertainties to balance.

Timescales are crucial wherever processes are involved.

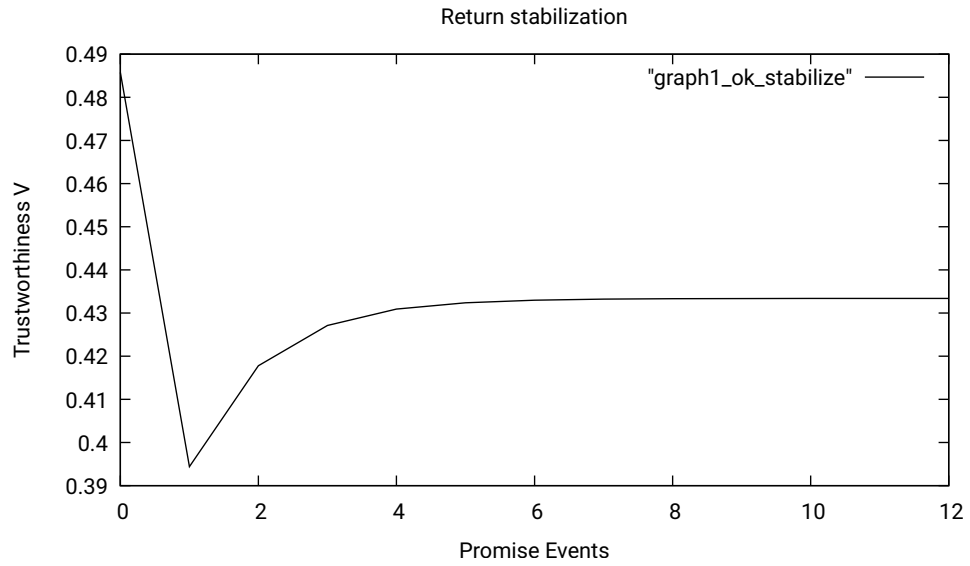

Figure 7: In the opposite direction a good rating settles to middling assessment on return to normal promise keeping.

4.9 How to update trust assessments

How do we change our minds? We don't exactly know how to define causal updates in terms of the updated values for T, V yet. Only after testing out a model on data will be get some idea about what scales and sensitivities are useful in practice. It seems unlikely that there will be any universal set of such scales. Promise Theory tells us that each agent's experiences will be quite unique. The only reference scales are those belonging to each agent's process scales for time and space. This is the Semantic Spacetime hypothesis [5].

For example,

- An agent may begin with positive curiosity, stimulated by some novel or aligned event, then move to long term interest and then choose to invest in a learning epoch of its relationship behaviour.
- An agent may begin with suspicion, stimulated by some novel or misaligned event, then move to long term mistrust or simply label the agent as untrustworthy.

What scale (time and 'space' or data quantity) do we want to use for trustworthiness and confidence? Possible ranges are $[0, 1]$, $[0, \infty]$, or $[-1, 1]$, $[-\infty, \infty]$

We now want to use these values to assess trust potential V for intent to keep the promise, and to allocate $\Delta\tau$ for the sampling interval. According to the theory, potential can be negative or positive. The more an agent is untrustworthy, the deeper the potential well (leading to an acceleration of $\overline{T}$). We keep to the convention that trustworthiness is more positive and untrustworthy or risky processes are more negative.

In a sequence of assessments, our reliability ρ tally and promise-keeping assessment α may be adjusted in arbitrary ways.

$$\rho \mapsto C \pm \sum_e \alpha_{\text{kept}}(\pi_e) \tag{11}$$

i.e. an acceleration of the rate of kinetic activity when approaching this.

We form a total reliability by quantitative AND qualitative assessments

$$\rho = \alpha_{\text{frequency}}(\pi) \times \alpha_{\text{content}}(\pi) \tag{12}$$

and from this we limit the scale of variation over the interval $[0, 1]$ using a classic sigmoid form:

$$V = \frac{1}{1 + e^{-k(\rho - \rho_0)/\rho_0}} \tag{13}$$

where ρ_0 is the policy determined baseline for the promise, and $k = 3$ is a dimensionless scaling constant which is determined mainly for aesthetic convenience in determining the range of values over which V is effectively spread. We now depend only on the relative rate of promise keeping through the dimensionless ratio ρ_0.

Having determined this value once, we need a memory function with a tunable forgetting rate to adapt in proper time. This is most useful in steady state promise keeping, and acts as a simple smoothing operation for sporadic impulses. We take a convex mixture

$$V_{n+1} = \beta\, V_n + (1 - \beta)\, \delta V \tag{14}$$

with $\beta = 2/5$ (initially) to give a small weight to new values for more rapid adaptation. These constants are chosen arbitrarily at this stage in order to capture the desired behaviour. Some of them might plausibly be determined by other constraints later.

The sum in this expression is related to what we've calculated on the fly with $Q_{\rm av}$. So if this value is within acceptable levels (over its averaging timescale) then we assess promise kept. The averaging rate is tunable. Ultimately, we want to set the sampling rate approximately as the square root of the negative trustworthiness potential.

In the case where one can choose the monitoring rate (at a rate less than the event frequency)

$$\Delta\tau(\pi) \sim \sqrt{|\Delta V(\pi)|}. \tag{15}$$

In many cases, interactions are not regular to we are limited to observing each event as singleton opportunistically.

4.10 Long term learning (out of band)

The utility of a currency to exchange guidance about interactions comes from the memory function of currency. By accumulating information from multiple sources (different bodies of knowledge) and exchanging it between specific contexts, the value of currency becomes far greater as a network exchange. This may be part of the function of trust as a proto reasoning method in an evolutionary sense.

This is true of trust, through reputation, for instance. Reputation is just assessments of trust passed on as hearsay. The weather forecast is useful as a policy information even though it isn't based on current data in context, because the data are widely applicable. We can gauge what the maximum temperature or rainfall *might be*, based on long experience. Although this is not directly likely or unlikely, it helps us to gauge confidence in changes that occur in realtime. For example, the learning that "in ten years it has never snowed on June 4 in Oslo", is helpful in being confident that we can move the snow plough from busy waiting to long term storage, i.e. trust the weather and be confident that we don't need the plough. If some flakes of snow appear in the sky, the confidence would be reduced, but one wouldn't assume that a few flakes would lead to a blizzard.

So we can use "similar experiences" by coarse graining to identify the limits of reason about what happens in realtime. Then we use the fluctuating potentials of kinetic and potential trust, confidence, and risk (informed but past learning) to shape allocation of resources for well-intended outcomes.

5 Extending trust to confidence and other attentiveness models

As already mentioned, trust is not alone as an attention potential. Confidence goes beyond trust in assessing whether an agent not only intends to try but is also likely capable of keeping its promise. We can imagine trying to reason about confidence by constructing an edifice of necessary and sufficient conditions to rely on as preconditions for success. In practice, this is both expensive and fragile, because these kinds of logical dependence arguments are brittle. Instead, we use data estimators based on coarse learning—to what extent do the promise we rely on disappoint across a range of activities (general trustworthiness).

To compute confidence, we need a specific insight into the context and meaning of a process. Only the participants and designers of the process can undertake this. But trust is so general that it can be computed for just about any process, based on the simplest of assessments. Perhaps this is all we need?

In a world where agents live in relative safety, i.e. in which the impact of undesirable and non-cooperative behaviours is small, trust is a great cost saver. But in a non-linear world of large and sudden surprises, such as a technologically enabled, highly connected, and heavily weaponized world, then trust may not be worth it. This is what we need to figure out.

	ESTIMATOR	*Semantics*	*Stake*
before the event	TRUST	stability,continuity,predictability	horizon in time
	CONFIDENCE	sufficiency, reachability (QoS)	horizon in outcome
after the event	RISK	restorability, resilience	horizon for recovery

Figure 8: The semantics of trust, confidence, and risk.

6 Summary

Critical dependency must play a role in the valuation of assessments about trustworthiness. If we depend for our survival on the catching of a fish, we will tend to mistrust the outcome of a promise that a fish will be caught, regardless of how we assess the trustworthiness of the fisherman. We maintain "and active interest" in the matter. However, mistrusting the fisherman by looking over his shoulder doesn't make him feel appreciated and will tend to lead to mutual mistrust. Finding the balance here in a potential arms race of oversight between agents is the problem to be solved in the human-machine society.

These considerations may serve as a causal guide; however, since trust is a policy determination, deciding trust is itself an ad hoc policy issue. We can't calculate a simple answer, but we can offer statistical guidance. The variability of context will pose a challenge here, since variability opposes accuracy. Stability enables certainty.

- Before a promise is made, we might look to reputation of agent for this or other promises.
- We might assess with what confidence the agent can keep its promise proposal, by considering information about resources, prior track record, etc
- During the running, we update assessment and decide whether trust and confidence in a successful outcome (promise kept for promisee, not necessarily "you", maybe as a third party).
  - Trust is out belief that the intention is there.
  - Confidence is our belief that the desired outcome will be delivered.

6.1 Relation to monitoring and the Dunbar limit

The Dunbar hypothesis suggests that there is a limit to our human cognitive capacity for monitoring relationships, as a finite agent—albeit a very powerful one. Technology can be used to outsource some of this, if we trust its judgement, but that is only a new form of trust. Trust is the antidote to trying to learn too much about too many, or becoming too close to others. Technology allows us to overcome this limitation artificially to validate behaviour, but our understanding of the results is still limited by our human cognitive faculties. Machine learning of behavioural patterns can act as a way to amplify our capacity for simplistic judgement—as simplistic or nuanced as we care to invest in the resources of programming. The programming costs are time and effort too, so we have to see them as a part of the cost or investment in mistrust.

Reliance on monitoring as a crutch to sense-making is presumably a confidence building measure, when we don't know what we're looking for—because, in the absence of clear promises, we may distrust systems to deliver what we believe they ought to promise. There's a paradox here, because we often mistrust because we don't actually know what we're looking for. We're afraid of missing whatever it is, if we don't keep looking very hard, so we keep sampling as quickly as we can. Most of that effort is wasted, like asking "are we there yet?" On the other hand, not looking as fast as you can (like the security camera example) is exploitable.

How much are we willing to pay (with certainty) to capture a rare unicorn event (probabilistically). This is the trust question.

An agent with a good understanding of its situation, and with sufficient information can do a reasonable job of modelling and computing its chances. Confidence calculations have to build on knowledge of system and context. The finance world does this, for example. In the IT industry, however, we are

nowhere near this level of modelling (or interest), because we have very few resource shortages. We wallow in excess power for a little more money.

7 Open questions

The relative level of T and V need to be fixed or calibrated to make sense in a quantitative way. This is true in physics too. For example, we might identify the shift from mistrust to curiosity or interest when trustworthiness increases beyond a certain level, like a confining potential. So interest is a bound state, while mistrust is a free process with some influence towards different promises.

7.1 What do we need from trust in the human-machine world?

Something is clearly missing from our idea of trust. In the adversarial world of computer security, there's a pervasive belief that we trust too much. On the other hand, the service economy (including the cloud) wants us to trust more. Outsource! Delegate! The coherence that comes from close interaction is disintegrating with mobile autonomy.

Database platforms claim to promise consistency and availability over wide areas, but these are not promises that can be made independently of application semantics. So we assess these to be—what? Over optimistic, deceptions? Under the influence of powerful companies, clients tend to fall in line with what is being offered. They have little choice, which is a reason to trust. This brings a certain coherence which is useful and cost saving.

- Passive commons: low level sparse consumption coexists easily, but is inefficient and therefore expensive in terms of return on investment.
- Attention seekers: advertisers, crowd sourcers, monitoring companies, etc. rely on being watched closely.
- Attention shirkers: hackers, long term disaster recovery archives, data privacy information.

Many companies and users employ cliches without actually understanding what they are giving or wanting. "Data consistency" and "scalability" are common phrases, but different users mean different things by them. Ultimately a service platform can only offer a few basic spacetime processes reliably.

- Equilibration of information and offerings is an advantage for stability, but too much stability is stagnation.
- Platforms may be read dominated (Wikipedia) or write dominated (banking transactions).

  If the rate of change of writes is greater than the rate of equilibration between mirrors, then agents at different locations will observe different values concurrently.

  If the rate of writes is greater than the rate of reads, clients will see only a subset of changes. Whether this is right or wrong can only be given meaning in the context of an application. Reading and writing of data are too low level to have any intrinsic meaning.
- Facility for separating contexts into virtual channels.
- Allow policy to manage contention issues.
- Perk up at transient behaviours, where steady states are learned along with trends (working week, etc). In most cases, a transient should probably be a short sustained anomaly not just a single unusual sample. Determined hackers will always find a way around these search criteria, but we can see how far this takes us.

Based on discussions with a small number of different users, I'm going to conclude that the best kind of platform to trust in is one that transparently enables the hosting of reliable spacetime information processes. We include a simple approximation to estimating rates and times, which can be used to estimate a kind of trustworthiness, with plugins for assessing data semantics. The user of the platform has to be entirely responsible for the way in which data are used. There is no meaning to data consistency without a thermodynamic equilibrium. That requires a minimum level of interaction or effective mistrust.

Acknowledgment: This work is supported by NLnet project Trust Semantic Learning and Monitoring.

References

[1] J.A. Bergstra and M. Burgess. Local and global trust based on the concept of promises. Technical report, arXiv.org/abs/0912.4637 [cs.MA], 2006.

[2] M. Burgess. Notes on trust as a causal basis for social science. *SSRN Archive, available at http://dx.doi.org/10.2139/ssrn.4252501 (DOI: 10.2139/ssrn.4252501)*, August 2022.

[3] M. Burgess. Notes on trust literature, bridging the perspectives of social philosophy and technology. *Personal notes available on Researchgate (DOI: 10.13140/RG.2.2.29476.35208/1)*, February 2023.

[4] M. Burgess. Trust and trustability: An idealized operational theory of economic attentiveness. *preprint paper (DOI: 10.13140/RG.2.2.26862.28480/1)*, April 2023.

[5] M. Burgess. A spacetime approach to generalized cognitive reasoning in multi-scale learning. *arXiv:1702.04638*, 2017.

[6] M. Burgess. Testing the quantitative spacetime hypothesis using artificial narrative comprehension (i): Bootstrapping meaning from episodic narrative view as a feature landscape. *preprint*, 2020.

[7] M. Burgess. Testing the quantitative spacetime hypothesis using artificial narrative comprehension (ii): Establishing the geometry of invariant concepts, themes, and namespaces from narrative. *preprint*, 2020.