10. Summary Sheet: A Trust User Guide

Mark Burgess

December 5, 2023

Abstract

This work is part of the study of trust sponsored by NLnet. This document is a concise final summary and "user guide" for trust.

Contents

1 Introduction

This is not a scientific paper. Think of it as cheat-sheet for the foregoing results of the project, and a way of using trust in all its forms and contexts without further explanation. See earlier installments and chapters for details.

2 Prerequisites

Many of us try to separate trust from a particular behaviour of promise, often to point a causal finger of blame of accountability. This is loose thinking, insufficient for making sense of trust. The first step in any trust relationship is to establish the parameters of what is offered, accepted, and expected by agents we interact with.

Next, we decide whether or not to engage with others. Finally we assess the reliability of an agent's behaviour in relation to out expectations and adjust the level of attention we invest in making sure our expectations are met.

3 How trust works for promises

Consider two agents S and R, where S is the 'sender' or servant of a promise to a recipient R.

1. S makes a promise π to R, with some body text b.

$$\pi^{(+)} : S \xrightarrow{+b_S} R \tag{1}$$

2. R assesses the trustworthiness $\alpha_R(\pi(S))$ of S to keep its promise π based on whatever information it can acquire:
   - Past experience of S keeping π
   - Past experience of S keeping other promises
   - Reputation of S from other agents.
   - Speculation about whether S or any other agent could keep a promise π.
3. If the assessment $\alpha_R(\pi(S))$ is favourable according to R, then R may accept the promise in some desired amount b_R

$$\pi^{(-)} : R \xrightarrow{-b_R} S, \tag{2}$$

   instigating a promise relationship with a degree of alignment $b_S \cap b_R$.
4. R keeps a running assessment of promise keeping reliability about $\pi^{(+)}(S, R)$ and denotes this the trustworthiness or trustworthiness potential for $\pi^{(+)}(S, R)$, denoted $V(pi)$.
5. R may use the information V as a currency to be used for the assessment of other promises by S or any other agent. R decides autonomously according to its own internal criteria.
6. As a matter of definition, R will assign a certain level of attention (i.e. kinetic activity associated with further assessment of its receipts). According to the dimensions of process velocity v the kinetic trust $V \mapsto T = \frac{1}{2}mv^2$, so that the rate of watching is up to the square root of the rate of the expected promise keeping V.
7. The kinetic trust budget of an agent tends to limit the size of collaborative groups it's involved in, according to the Dunbar number hierarchy.
8. If the alignment of promise keeping wavers or reliability wavers, R can reassess its acceptance and may withdraw the promise, thus ending the relationship.

4 How trust works for impositions

Impositions are unexpected transient attempts to induce acceptance. They include pushes and accusations. The lifecycle is the same for promises, except that an imposition will tend to yield a less favourable estimate of trustworthiness of S by R.

Consider two agents S and R, where S is the 'sender' or servant of a promise to a recipient R.

1. S makes an imposition ι to R, with some body text b.

$$\iota : S \xrightarrow{+b} \blacksquare\ R \tag{3}$$

2. R assesses the trustworthiness $\alpha_R(\iota(S))$ of S to keep its promise π based on whatever information it can acquire:
   - Past experience of S keeping π
   - Past experience of S keeping other promises
   - Reputation of S from other agents.
   - Speculation about whether S or any other agent could keep a promise π.

5 Common idioms interpreted

We are loose in our use of language about trust. We can be more precise in terms of the definitions used in Promise Theory.

1. When someone says 'I don't trust S', they mean: I don't assess S to be trustworthy in the present context.
2. When someone says: 'I don't trust your promise (to do that)', they mean: they don't accept the promise, they assess the promised intentions will not be kept, either a deception/lie or unrealisable. The promiser is not honest about the outcome. This is about promise alignment of intent. In this case, there is not a promise relationship to monitor yet, because the promise offer (+) has not been accepted.
3. When someone says 'I'm going to trust you', they mean: I will accept your promise (or even your impositions to some extent) and I will not watch too closely.
4. When someone says 'trust but verify', they mean: I will accept your promise and I will watch too closely.
5. Bertolt Brecht's quote: "I don't trust him, he's my friend"is deliciously ambiguous in its meaning. It could mean:
   - I assess him to be untrustworthy, because I know him so well.
   - I don't need to even think about whether or not he is trustworthy because I know him so well that I choose to (kinetically) trust him completely.
   - Because he is my friend, our relationship goes beyond trust or mistrust to something "higher". This is a more mystical and speculative assertion, as we don't know what might be of higher purpose than trust. In practice, the effect is the same as the second case.

6 The group scaling formula

The formula for group scaling, which predicts the Dunbar group size laws:

$$\psi(\nu) = \frac{4}{\sqrt{\pi}} \frac{\nu^{\frac{1}{2}}\, e^{-\nu}}{\langle N \rangle_{\rm ep}}, \qquad \nu = \frac{2(N-1)}{\langle N \rangle_{\rm ep}}\ ,\ (N > 1). \tag{4}$$

where N is the probable size of a group of humans and $\langle N \rangle_{\rm ep}$ is the size at which conflict is maximized, is to be interpreted as follows. The promise structure is that of a hub and spoke topology: there is a single seed that attracts others to a group, hence an $(N-1):1$ role split. The single promise focus that is mistrusted by the $N-1$ followers and thus repeatedly assessed at a rate increasing like the square root of the assessed potential for the thing mistrusted. Once the size of the crowd that comes to look grows past $\langle N \rangle_{\rm ep}$, the mutual presence of others begins to disrupt the group leading to a maximum size which is below the size at which conflict is maximal.

7 Kinetic trust: attentiveness

There is a continuous spectrum of attentiveness from curiosity to mistrust. The differences in semantics are subtle, not as clear as we might think. The degree of attentiveness is independent of our assessment of intent. We tend to use the term mistrust only for assumed intent to harm, but the result of excessive curiosity is easily perceived as mistrust by others.

The implications for machine learning are interesting. Unless we are up to date in training, users may have reason to mistrust ML systems. If ML trains too much, people might feel spied upon, just as people react negatively to surveillance (particularly in Western countries).

Attention mechanisms are used in many human contexts and in many technologies.

- Any service is based on attentiveness.
- Any cognition learning process is based on observation.

Remark 1 (Attention models in deep learning) *The term "attention" is also used in deep learning as an expression of adaptive weighting of words in text prediction for "transformer" contextualization models. The idea is somewhat similar. In the language of trust, we would say that the model assesses the rigid meanings of words frompretraining to be relatively untrustworthy so one has a policy of kinetic attention for reevaluating the results continuously.*

8 Software methods for trust

8.1 White paper on trustable software

The 2017 white paper by Codethink "Towards Trustable Software, a systematic approach to establishing trust in software" is compatible with the findings of this project. It proposes two parts that correspond as follows:

Trtustability (potential reliabilty):

- We know where the code comes from.
- The code does what it promises.
- We know how the code was built.
- We can reproduce from source code.
- We can update the software without breakage or regression.

Auditabilty (kinetic attention):

- History of Requirements, scope
- History of Solution architecture
- History of tests, results and validation
- History of development (traceability)
- History of Maintenance changes

8.2 Analytic Assessments

We can assess promise keeping for transactions by embedding machine learning. First we open an analytics context with database in the background:

```
var dbname string = "SemanticSpacetime"
var url string = "http://localhost:8529"
var user string = "root"
var pwd string = "mark"

g := TT.OpenAnalytics(dbname,url,user,pwd)
```

This does nothing more than defining a handle for establishing database access.

8.3 IT Transactions and Client-Server Assessments

In technology, many interactions take the form of transaction processing. We can wrap transactions atomically and use internal clocks as the arbiter of process rates. The results learn average times for typical services at typical times of the working week. The working week is always the dominant temporal statistical pattern in data, if there is any.

A particularly simple API can be established for this.

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

        // Do transaction

e := TT.PromiseContext_End(g,ctx)

// Do we know what was promised? Or how to express it?

promised_upper_bound := 1.6 // response time in seconds
trust_interval := 1.0       // monitor interval in seconds

V := TT.AssessPromiseOutcome(g,
                  e,
                  MyAssessResult(string(received)),
                  promised_upper_bound,
                  trust_interval)
```

Collecting data like this is no guarantee of being able to make sense of normal and anomalous results, but it gives us a starting point for sense-making.

8.4 Heuristic signalling

The simplest way to assess trustworthiness autonomously is to decide the criteria (i.e. policy) for mistrust for the autonomous process and then use the signalling model in the TT library:

```
  // import "TT"


TT.InitializeContext()

        if state == busy {
                TT.ContextAdd("busy")
                }

        if mistrust > 1 {

                TT.ContextAdd("explicit_undo")
                TT.ContextAdd("state_of_contention")
                TT.ContextAdd("state_of_uncertainty_about_article")
        }

        running_context := TT.ContextSet()

        for signal := range running_context {

                fmt.Println("Context: ",running_context[signal])
        }

        confidence := TT.Context("explicit_undo && busy | just_don't_like_you")
```

8.5 Text sampling

For the subsampling of text based on an estimate of the work factor, novelty, or "intentionality", we first have to set a length scale for the typical rate of meaning production. I call this the "leg window". Then we use the potential/kinetic formula to estimate the rate of sampling with the leg: selecting the top n sentence samples, where:

```
n = int(0.5 + math.Sqrt(float64(LEG_WINDOW) * scale_free_trust))
```

8.6 Limiting access

For the service transaction sampling in section 8.2, we can use the dynamic locking described in [1] as a rate limiting mechanism, similar to neuron dead-times. Normally we either sample every transaction or none. Using this mechanism, we limit access to the service itself as a protection against 'spamming' or denial of service attacks.

8.7 Cost of work of attention—general guidelines

We don't need to create a library to measure the work of attention. A rule of thumb should suffice to implement the idea in other contexts. Essentially we are trying to measure the amount of time something takes to complete. As in complexity classes P, NP, PSPACE, etc, we can always convert different measures into effective times. The length of a string is proportional to some function of the time to process it.

The work intent associated with a pattern falls off with reuse after a time. The most used patterns are the least significant in the end. They lose their significance by being common and mass reproducible. They recur because they are "cheap".

The scale of attention is important when measuring intent. High level, coarse monitoring may still be appropriate when micro-management attentiveness is considered unwelcome.

If we agree that the purpose of trust is to limit the cost of this work of attention, then basically any measure of the amount of work, which is consistently applied, should suffice as a potential V associated with some intentional process, or promised outcome.

9 Originality and boundaries is a densely connected world

As we become more connected, and pay attention to the things we choose, its harder to separate ourselves from our environments. How can we define individuality, originality? How does this change copyright convention? We already see these issues playing out, especially with machine learning tools that explicitly mine information from the public view. Knietic trust becomes hard to maintain when everything is vying for our attention, and if we insist that everything be "verified". This means that:

- Privacy loses its meaning.
- Originality loses its meaning.

To restore trust in the usual sense, we need to manage boundaries for these new challenges of population density.

10 Trust and AI

The role of trust in AI related matters iss quite broad. Recently researchers have begun to publish the most basic considerations on this topic.

Role of trustworthiness:

1. Reliability and stability (promise keeping) of training sources.
2. Functional correctness of "AI" tools (fitness for purpose).
3. Reputational assessment from public discussion–AI as a concept.

Role of kinetic mistrust/attention:

1. The decision to watch the space, follow AI developments, and use the latest.

2. Attention mechanisms in deep learning: switching between context sets, inhibitor signals etc.
3. Self-censuring, assessing own correctness.
4. Enagaging in the public debate: ethics, self-interest, human welfare, etc.

Many of these also apply to general automation systems.

11 Role of trust in society

As we trust society, and take it for granted, we disengage. The Dunbar group work supports this hypothesis quantitatively. Disengagement tends to cause group instability.

The issue for trust in the 21st century is: how do we want to live? If we want to preserve the coherence of our society (as we understand it today), then the conclusions from the Promise Theory of trust might be summarized like this:

- Don't live by imposing obligations or activities on others (demand their effort).
- Don't throw accusations or blame (contention wasting time).
- Always offer something to others (a reason not to question rather than ignore you).
- Always be willing to accept what others offer (give the benefit of the doubt after suitable attention has been invested).

In other words: pay attention to one another, and don't take too much for granted. Conversely, if we are watching each other too much and we use or depend on what we see, there can be too strong a binding between agents. Then it becomes difficult to argue that the agents are independent and can make individual contributions. This has significance for copyright and ownership, as well as the sense of privacy. In normal times, "market forces" tend to find the balance between these poles, but if we hand more of our affairs over to machinery it becomes harder to say that they reliably represent our interests.

These are rules of thumb for maintaining stable groups. If we can see beneath the apparently moral facade, they are simply about saving effort by fostering alignment of intent. What is intent? It's the direction of activity, whether chosen or emergent (if you could even tell the difference).

References

[1] Mark Burgess and Demosthenes Skipitaris. Adaptive locks for frequently scheduled tasks with unpredictable runtimes. In *11th Systems Administration Conference (LISA 97)*, San Diego, CA, October 1997. USENIX Association.