Strategies and reputation - a microeconomic description of the Golem marketplace

Jan Betley
Golem Factory

January 28, 2022

1 Introduction

1.1 Reader notes

The most important statements are labelled as Conclusions - consider skimming through them if you're not interested in the reasoning.

This document is quite formal, but please don't expect clean, bulletproof implications only. The purpose of this document is to present the topic and conclusions while maintaining resonable level of brievity.

1.2 Abstract

This document provides a simple microeconomic model of the Golem marketplace. Following questions are answered within the model:

- What is the exact problem we hope to solve with the reputation system?
- What do we exactly mean by the "reputation"?
- What are the other, non-reputation approaches to the main problem?
- How to measure the quality of our solutions? I.e. how to know if we succeeded?
- How to split the "reputation" into separate subproblems?

The topic is approached from the highest point of view - no specific solutions are proposed, only general classes of solutions are discussed. With an exception of a few details/examples this document describes just a "general" marketplace (replace "provider/requestor" with "seller/buyer").

1.3 Definitions

Agent - A decision-making entity (person, company, etc.).
Utility - The total happiness of an agent.
Utility function U - A function $U : StateOfTheWorld- > \mathbb{R}$, defined for a particular agent, that represents their utility in a given situation.
Expected utility - For every possible decision an agent can make (D) and every possible state of the world w, there is some probability $P : (D, w) \rightarrow [0, 1]$ that given decision will lead to the given state of the world. Agents don't have the full knowledge about P, they know only some information I and an estimation of P based on this information: $P_I : (D, w) \rightarrow [0, 1]$. The expected utility $E : (I, U, D)- > \mathbb{R}$ is defined as follows:

$$E(I, U, D) = \sum_{w \in all\_possible\_world\_states} U(w) * P_I(D, w) \tag{1}$$

We'll be also using a shorter notation, $E(I, U, D) = E_A(D)$ to describe "expected utility of the decision D, for agent A, who has information I and utility function U".

In other words, for agent A and decision D, $E_A(D)$ is "how happy agent A expects to be if they do D".

Golem - The observable result of all the work done by the Golem Factory - the Golem protocol, available software, state of the market etc.

1.4 Additional assumptions

1. The Expected Utility Hypothesis[1]: every agent tries to maximize their expected utility.

2. Every agent can costlessy access and analyze all of the publicly available information about Golem. Thanks to this assumption we can remove I from the equations - we no longer care about "what agent knows", but only about "what really is". This is a major simplification that might not be a good approximation of the "real world" Golem market, but we accept it for the sake of the brevity of this document. Also "truth about Golem is known" is an ideal world we would like to live in.

3. The expected utility and money are interchangeable, that is - for every agent A and a pair of decisions $(D_1, D_2) : E_A(D_1) < E_A(D_2)$, there is an amount of money X that can be given to the agent so that $E_A(D_1) + U_A(X) = E_A(D_2)$. This way we can treat utilities as if they were money, and thus compare them between different agents.

4. Lets define the Golem Value as:

[1] https://en.wikipedia.org/wiki/Expected_utility_hypothesis

$$V_G = \sum_{A \in all\_agents\_using\_Golem} E_A(USE\_GOLEM) \tag{2}$$

(Note that - because of assumption 1 - $E_A(USE\_GOLEM)$ is always positive, decision "use Golem" is only made by agents who expect to profit from it).

We assume that maximization of V_G is one of the goals of the Golem Factory - and from the point of view of this document, the only goal.

2 The "reputation" and its purpose

2.1 Notation

We'll be using a bunch of different symbols, but they follow a clear common pattern, so please don't be frightened.

- $V_A(a)$ is the total value of the agreement a from the POV of agent A. As per assumption 3, this is "utility expressed as money".[2], i.e.:
  - For the provider it is the utility of the money received decreased by the utility cost of the hardware/electricity/etc.
  - For the requestor it is the utility of the resources obtained decreased by the utility cost of the money spent
- $V_P(a)/V_R(a)$ are agreement values from the POV of (respectively) provider/requestor.
- $V_{PN}(a)/V_{RN}(a)$ are nominal (i.e. negotiated) values of the agreement. They equal $V_P(a)/V_R(a)$ if neither side breached the agreement a.
- $V_{AL}(a)$ is the value lost by agent A because of the other side breaching the agreement a.
- $V_{AG}(a)$ is the value gained by agent A when they break the agreement a.
- V_{PL}, V_{RL}, V_{PG}, V_{RG} are V_{AL}/V_{AG} from the POV of provider/requestor.
- C_A is the cost of the participation in the Golem market for agent A that is not related to any particular agreement (e.g. the opportunity cost[3] of the hardware offered on the market, or the cost of writing the requestor agent).

[2] I.e. $V_A(a) = X$ means "when agent A takes part in the agreement a, their hapiness changes as if they were given X money".

[3] https://en.wikipedia.org/wiki/Opportunity_cost

2.2 Dissolving the Golem Value

Utilizing the above notation, we can rewrite the V_G equation as:

$$V_G = \sum_{a \in all\_agreements} (V_P(a) + V_R(a)) - \sum_{A \in all\_agents\_using\_Golem} C_A \tag{3}$$

The following equation and it's counterpart from the requestor POV are true by definition:

$$V_P(a) = V_{PN}(a) + V_{PG}(a) - V_{PL}(a) \tag{4}$$

By placing them in the previous equation, we get:

$$\begin{aligned} V_G = \sum_{a \in all\_agreements} (V_{PN}(a) + V_{PG}(a) - V_{PL}(a) + V_{RN}(a) + V_{RG}(a) - V_{RL}(a)) \\ - \sum_{A \in all\_agents\_using\_Golem} C_A \end{aligned} \tag{5}$$

$$\begin{aligned} V_G = &\sum_{a \in all\_agreements} (V_{PN}(a) + V_{PG}(a)) \\ &- \sum_{a \in all\_agreements} (V_{PL}(a) - V_{RG}(a)) \\ &- \sum_{a \in all\_agreements} (V_{RL}(a) - V_{PG}(a)) \\ &- \sum_{A \in all\_agents\_using\_Golem} C_A \end{aligned} \tag{6}$$

Important note here is that both $\sum_{a \in all\_agreements} (V_{PL}(a) - V_{RG}(a))$ and $\sum_{a \in all\_agreements} (V_{RL}(a) - V_{PG}(a))$ are positive: when someone breaks the agreement, the harm done to the victim is usually greater then the offenders gain[4]. Let's thus define one more symbol:

- $V_L(a) = (V_{PL}(a) - V_{RG}(a)) + (V_{RL}(a) - V_{PG}(a))$ - the total value lost because of agents breaking the terms of agreement a.

and use it to rewrite the main equation one last time:

$$\begin{aligned} V_G = &\sum_{a \in all\_agreements} (V_{PN}(a) + V_{RN}(a) - V_L(a)) \\ &- \sum_{A \in all\_agents\_using\_Golem} C_A \end{aligned} \tag{7}$$

This equation defines few general ways to increase the V_G:

[4] TODO: examples/proof/justification?

1. Increase the number of agreements
2. Increase the average nominal value of the agreement (e.g. by providing additional capabilities, like internet connectivity or GPU access)
3. Decrease the average value lost because of breaches of the agreements ($V_L(a)$)
4. Decrease the cost of the participation in the Golem market (e.g. by creating better SDKs)

Note that these ways interact with each other: e.g. if we improve 2 by implementing some features that will be hard to use, we'll also worsen 4. Or: the better is the average nominal value of the agreement, the more agreements we'll expect to have. Or - what is important from the POV of this document - if we implement complex safeguards against cheating, they might have negative impact on all the other points.

Keeping that in mind, the rest of the document is aimed at the third direction from the above list.

2.3 Honest strategy

Fo an agreement a between agents A_1 and A_2 let's define the "dishonesty index" of agent A_2 as:

$$D_{A_2}(a) = \frac{V_{A_1L}(a)}{V_{A_1N}(a)} \tag{8}$$

That is: if A_1 received everything that was agreed, $D_{A_2}(a) = 0$. If all agreement-related costs of A_1 are covered, but they got nothing more, $D_{A_2}(a) = 1$. If A_1 lost on the agreement, $D_{A_2} > 1$.[5]

The value of $V_{A_1L}(a)$ (so also the dishonesty index) depends on the decision made by agent A_1. In an usual case the "visible decision" will by caused by the algorithm implemented in the provider/requestor agent, but this is just an effect of the agent's "real decision" about the implementation.[6]

The agent's core decision-making process is called a "strategy". The better agent fulfills their part of the agreements, the more "honest" is the strategy. So, in other words: when agent A signs an agreement a with an agent using a dishonest strategy, the expected value of $V_{AL}(a)$ is high. When the other side is fully honest, $V_{AL}(a) = 0$. Note that agent's intentions don't matter here - there's no difference if an agent breaks an agreement on purpose or accidentally.

Agent always tries to maximize their own utility (the "expected utility maximization" assumption), in other words: agent always selects the most profitable strategy. Let's now paraphrase the third goal from the previous section as:

[5] Values below 0 are also possible, e.g. paying more than agreed is also "against the agreement".

[6] In an extreme case we can imagine a human operator who directly interacts with the Golem protocol. Or just imagine someone who turns off the currently rented hardware.

Conclusion 2.1 *The final purpose of the reputation system on Golem is to make honest strategies more profitable than dishonest strategies.*

2.4 Non-reputation solutions

Let's consider few directions towards the goal defined in the previous section:

- Make dishonest strategies not available at all, e.g. ensure that debit note acceptance forces payment.
- Make dishonest strategies hard to implement, e.g. hide/obfuscate some important components of Golem.
- Add some mechanics that directly penalize dishonesty, e.g. require deposits and confiscate them when dishonesty is proven.

It's important to note that - while they have the same purpose as the reputation - they have nothing else to do with anything we call the "reputation".

Conclusion 2.2 *Golem Factory works on the reputation system because we believe it's the best way to achieve goal defined in Conclusion 2.1.*

2.5 Defining reputation

$V_{whatever}$ is a determined, known value. When trading on the Golem market, some values are known from the start (e.g. $V_{AN}(a)$), and other only post factum (e.g. $V_{AL}(a)$, so also $V_A(a)$). When making decisions under uncertainity, we're using expected values - they will be written as $E(...)$, e.g. $E(V_A(a))$.

Imagine an honest agent A_1 who considers signing an agreement a with agent A_2. The decision algorithm can be rougly summarized as:

1. Calculate the expected value of the decision **not** to sign the agreement, $E_{A_1}(\neg a)$
2. Calculate the expected value of the decision to sign the agreement $E_{A_1}(a) = V_{A_1N}(a) - E(V_{A_1L}(a))$
3. Sign the agreement if $E_{A_1}(a) > E_{A_1}(\neg a)$

This leads us to a simple observation: the higher dishonesty we expect, the better E_N we need to sign the agreement, and thus to:

Conclusion 2.3 *"Reputation system" is an attempt to solve the problem defined in the conclusion 2.1 in the following way:*

- *Make some additional information available to the market participants*
- *This information can be used to estimate the "dishonesty index" of an agent, and thus improves the accuracy of the total agreement value estimation*

- *The more accurate is the total agreement value estimation, the less profitable it is to trade with dishonest agents*
- *The less profitable it is to trade with dishonest agents, the fewer/worse agreements they have*
- *The fewer/worse agreements dishonest agents have, the less profitable are dishonest strategies*

3 Summary & discussion

3.1 Agreement value estimation

The mechanics described in the last conclusion work only when we can estimate the total value of the agreement. It's worth noting that such measure is useful also for other purposes, such as estimating the total income from a provider node, or just determining if Golem is worth using at all.

Conclusion 3.1 *Any reputation system requires a reasonable way of estimating the "expected dishonesty index" of an agent, or more generally - the total agreement value.*

3.2 Optimal strategies

Different agents have different utilities and thus different strategies. Also, the market changes because of agents leaving/entering it. We should not expect there to be a single best strategy, but rather a constant mix of strategies governed only by a single rule: as the time passes, profitable strategies become more common.

E.g. consider a market where the optimal strategy for the provider is to be less honest than the current average market honesty. We should expect the average honesty to fall, and if this is still true for the lower average honesty, our reputation system might be in a death spiral towards the zero-honesty of all providers. Note that this situation is hard to recover from: if all providers are equally dishonest, there's no incentive for requestors to use the dishonesty-penalizing strategies, so there will never be any incentive for a provider to increase their honesty.

But this works also the other way. If at a given moment the optimal strategy is to be **more** honest than the average, we'll see the average honesty growing. In fact, that could be our goal, but the weaker version is enough[7]:

Conclusion 3.2 *Reputation will work only on a market where the honesty of the optimal strategy always exceeds some treshold.*

[7] Perfect honesty will not be achieved by the means of reputation without the perfect agreement value estimation (Conclusion 3.1), and thus we should settle for a lower goal. Also perfect honesty (at least the way we defined it, that is - without a reference to the intentions) is not really possible because of random events.

3.3 Market balance

How to ensure the condition from Conclusion 3.2 is satisfied? The general answer is: by influencing supply/demand balance. E.g. let's say we have a market where:

- supply and demand are reasonably balanced
- agent honesty varies, so honesty-based strategies are common

and we want to increase the average honesty of the requestors. There are two ways we can approach this:

- By influencing the supply:
    1. Add to the market high-quality (e.g. cheap) providers who strongly penalize dishonesty.
    2. Requestors want to trade with the new providers, so they have an incentive to become more honest.
    3. As the requestors' strategy has changed, the new balance will be preserved even after we remove the providers added in 1.
- By influencing the demand:
    1. Add to the market high-quality (e.g. accepting high prices) honest requestors.
    2. There are more honest requestors available, so providers trade less with the dishonest requestors.
    3. Buying becomes more expensive for the dishonest requestors, so they have an incentive to become more honest.
    4. As the requestor' strategy has changed, the new balance will be preserved even after we remove the requestors added in 1.

On the other hand, to decrease the requestors' honesty below some treshold, it's enough e.g. to add to the market enough providers who don't care about requestors' honesty at all.

As the market changes, so will the average honesty - it's our role to care about its value[8]:

Conclusion 3.3 *Condition described in Conclusion 3.2 is not a phenomenon we should expect to naturally emerge when the information required for Conclusion 3.1 is available. It must be actively shaped by the market interventions, and will be always vulnerable to a sufficiently strong intervention against it.*

[8] Note that the bigger is the market, the more stable are the market equilibriums. We should hope that one day it will be big enough to make all the changes negligible.

3.4 Plan for the Golem Factory

NOTE: this section should **not** be treated as something final, but rather as a feed for thoughts/discussions[9].

1. Implement a way to calculate the "expected total agreement value".
   (a) Gather and share relevant information about providers/requestors
   (b) Implement some ways to estimate the total agreement value from the available information. Note that this might be a very rough estimation, e.g.:
      i. For the provider - "(When signing the agreement) I expect to be paid $X\%$ of the final invoice".
      ii. For the requestor - "I expect there is $X\%$ chance provider will break the agreement before finishing", or "I expect provider to have $X\%$ performance of the average provider with the same offer, doing the same task".
   (c) Make this modular/clean enough - we should make it easy for the Golem market participants to implement their own estimation methods, better suiting their needs, or utilizing the other information they have access to.
2. Implement the evaluation of the "total agreement value" estimations. The market balance will be changing, dishonest agents have an incentive to worsen the estimations - we must know if our reputation system is working well enough.
   (a) Spawn our own providers and requestors, make them estimate agreement values (using methods from the previous point).
   (b) Gather data ($estimation\_method, estimated\_value, final\_value$).
   (c) Control the quality of the estimation methods, improve them when it degrades.
3. Implement market strategies that reasonably utilize the calculated "expected total agreement value". Example strategies:
   (a) For the provider - if I expect to be paid $X\%$ of the invoice, I multiply the offer price for this requestor by $\frac{1}{X}$.[10]

[9] Four elements of the reputation system are specified here. For a better grip of their purpose, consider following exercise:

- Imagine any working reputation system you know (e.g. stars on Amazon).
- Find counterparts of this four elements in the other system.
- Consider the consequences if they went missing.

[10] Actually, this particular strategy is not good enough because it's vulnerable to the Pascal's mugging (https://en.wikipedia.org/wiki/Pascal's_mugging) counterstrategy. E.g. imagine we estimate the requestor will pay only 1% of the invoice, but the offer us a billion GLMs - this doesn't look like a good deal at all.

(b) For the requestor - if I expect there to be a $X\%$ chance the provider will break the agreement before finishing computations, I multiply their offer score by $1 - X$.

4. Take care about the market balance by maintaining artificial providers/requestors with appropriate strategies.