Strategies and reputation - a microeconomic description of the Golem marketplace

Jan Betley
Golem Factory

February 10, 2022

1   Introduction

1.1   Reader notes

The most important statements are labelled as <u>Conclusions</u> - 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

1.3 Definitions

Agent - A decision-making entity (person, company, etc.).
Utility - The total happiness of an agent.
Utility function U - A function $U : StateOfTheWorld \to \mathbb{R}$, 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 $P : (D, w) \to [0, 1]$ 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: $P_I : (D, w) \to [0, 1]$. The expected utility $E : (I, U, D) \to \mathbb{R}$ is defined as follows:

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

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 costlessly 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. [2]

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

---

[1]https://en.wikipedia.org/wiki/Expected_utility_hypothesis
[2]Note that this assumption applies only to the "general" information about Golem that could be a public knowledge. Some examples:

- Free: bugs/vulnerabilities in software/protocol, number or agents on the market, average payments, plans of the Golem Factory.

- Not free: anything about any particular agent, anything about the future.

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:

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

(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. Every V_* is an *ex post* value, i.e. evaluated after the agreement is completed and all agreement-related effects are known.

- $V_A(a)$ is the total value of the agreement a from the POV of agent A. As per assumption 3, this is "utility expressed as money".[3], 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[4]. 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. In most cases $V_{PL} > V_{RG}$ and $V_{RL} > V_{PG}$, reasoning behind this statement can be found in footnote 6.

---

[3]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".

[4]Note that this doesn't translate directly to the agreement amount. E.g. when agreed amount is positive but so low it doesn't cover provider's costs the V_{PN} is negative.

- 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[5] of the hardware offered on the market, or the cost of writing the requestor agent).

2.2   Decomposition of the Golem Value

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

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

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

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

By placing them in the previous equation, we get:

$$V_G = \sum_{a \in all\_agreements} (V_{PN}(a) + V_{PG}(a) - V_{PL}(a) + V_{RN}(a) + V_{RG}(a) - V_{RL}(a))$$

$$- \sum_{A \in all\_agents\_using\_Golem} C_A$$

$$(5)$$

$$V_G = \sum_{a \in all\_agreements} (V_{PN}(a) + V_{PG}(a))$$

$$- \sum_{a \in all\_agreements} (V_{PL}(a) - V_{RG}(a))$$

$$- \sum_{a \in all\_agreements} (V_{RL}(a) - V_{PG}(a))$$

$$- \sum_{A \in all\_agents\_using\_Golem} C_A$$

$$(6)$$

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[6]. Also the agreements on the "untrusted" market tend to have a lower

---

[5]https://en.wikipedia.org/wiki/Opportunity_cost
[6]Two different justifications behind this statement:

- From the requestor victim POV: the rented computer power is only a component in something bigger the requestor tries to build. When they don't get this component, the total loss is the value of this bigger thing that is not working.

  Imagine e.g. a factory. When cut off the electricity, the factory owner loses all the

V_N[7]. 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:

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

This equation defines few general ways to increase the V_G:

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.

---

goods factory would have produced, and their value is much higher than the value of the electricity (e.g. because it must be enough to cover all the other non-electricity costs).

Or a more Golem-like example: imagine an online shop with an Amazon-based database. If the database goes offline the total loss is all the sales that could have taken place while the shop was not working.

- From the provider victim POV: people tend to have a strong loss aversion (https://en.wikipedia.org/wiki/Loss_aversion). The loss perceived by the provider when losing X money will be on average more than the gain perceived by the requestor who saved X.

[7]When an agent expects they might be cheated, they might implement some precautions/safeguards. They cost additional money that are spent even when trading with an honest agent, and thus lower the agreement value.

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, $D_{A_2} > 1$.[8]

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.[9]

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:

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.*

---

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

[9]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.

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 $E_{A_1}(a) = V_{A_1 N}(a) - E(V_{A_1 L}(a))$

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 *The value of the reputation system depends on how well the reputation-related information improves the "total agreement value" estimation.*

Note that the same reputation system might be valuable for some agreements and useless for other agreements, or even valuable for some agents and useless for others.

3.2 Optimal strategy dynamics

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 law: as the time passes, profitable strategies become more common.

As an example, let's assume we have some reputation-related information available to the requestors (e.g. provider benchmarks). If the number of requestors utilizing this information grows, the incentive for providers to have a high value of the benchmark increases. If number of such requestors is falling, we should expect more providers not caring about their benchmark values.

Conclusion 3.2 *The higher is the adoption of the reputation system (i.e. the more important role it plays in the agents' decisions), the more it influences agents' strategies - and thus works better.*

3.3 Market interventions

And how to increase the adoption of the reputation system? This is a sort of an egg/chicken dillema. Why be honest, when no one cares about honesty? And why care about honesty, when everyone is equally dishonest? [10]

The answer is: by market interventions. If we put on the market enough high quality honest agents, who strongly penalize dishonesty (i.e. don't want to trade with dishonest agents), we'll see the balance changing. As the new agents are high quality, everyone will want to trade with them, and this creates two incentives:

- Our new agents are honest, so there is a reason to use the honesty index as a part of the market strategy (e.g. to find them).

- Our new agents require honesty, so there is a reason to be more honest than the average.

The important thing to note here is that - as both sides change their strategies - the new balance should be preserved even after we remove the new agents from the market.[11] This works also the other way: even a very good reputation system will be vulnerable to a sufficiently strong market intervention against it.

---

[10] This sounds extreme from the POV of the "current" market, where most of the providers use the default, honest software, but this is a lower bound towards which our market will be heading without a sufficiently good reputation system.

[11] This statement might require a more careful analysys (e.g. are there any implicit assumptions about the strategies that should be stated explicitly?).

Conclusion 3.3 *Development/maintenance of the reputation system involves active market interventions that will increase its adoption.*

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[12].

1. Provide ways to estimate the agent's honesty-related behaviour:

   (a) Gather and share relevant information about providers/requestors.

   (b) Implement some ways of utilizing this information in a way that can be directly used in a market strategy, 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 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 calculate values 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 some reasonable market strategies, e.g.:

   (a) For the provider - if I expect to be paid $X\%$ of the invoice, I multiply the offer price for this requestor by $\frac{1}{X}$.[13]

---

[12]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.

[13]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 adoption of the reputation system:

- Indirectly - e.g. by setting default provider strategies in yagna, or by implementing high quality example requestor-side strategies.

- Directly - by maintaining artificial providers/requestors with appropriate strategies.