Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

TODO rename

Márton Donát Nagy

Master's Thesis

TODO rename

Master's Thesis

Supervised by Prof. Dr. phil. nat. Sebastian Steinhorst
Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

Advisor Adam Advisor

Co-Advisor Corinna Coadvisor

Author Márton Donát Nagy
Musterstr. 42
42424 Musterstadt

Submitted on July 15, 2021

Declaration of Authorship

I, Márton Donát Nagy, declare that this thesis titled "TODO rename" and the work presented in it are my own unaided work, and that I have acknowledged all direct or indirect sources as references.
This thesis was not previously presented to another examination board and has not been published.

Signed:

Date:

Abstract

This thesis is about ... This thesis shows that ...

Contents

1

Introduction

TECHNOLOGY is on the move and this topic is important because it will change the world.

1.1 Problem Statement

As a long term goal we would like to have ... The problem is that ... still does not work. So we will investigate the questions

- whether A
- or whether B

1.2 Scenario

The scenario and assumptions and the background of this thesis. We assume ..., we will focus on ... and we will exclude ...
on a more broader sense, the work is intended to be general enough that it could be applied for other iot/cps type applications as well

1.3 Task Definition

We will do

- try A
- try B

1.4 Structure of This Document

First, ...

1.5 Terminology

describe user, peer, agent, principal, node etc namings in research used according to domain of the paper
network
underlying value, trueness, ground truth,
reputation score, estimate, global rating
rating, opinion, review
Content, transaction, service, product, file, claim
advisor, witness, second hand reputation
internal vs agent-exposed: in the program, internal is how reputations, reviews, claims etc are stored and refers to [0,1] real agent-exposed is how the agent sees everything, typically [1,9] and discrete integers in the code _i vs _ae denotes which type a variable stores
score: review value, claim value, reputation etc. referred generally

2

State of the Art

This chapter gives an overview of ...

2.1 Domains of Trust and Reputation Research

discuss fields like WSN, p2p filesharing, MAS, mobile sensor systems, flying ad hoc networks, vehicular ad hoc networks, ecommerce, recommendation systems online stuff like imdb or hotel ratings....

2.2 Evaluation and Comparison

2.2.1 Theoretical Categorization

2.2.2 Simulation Frameworks

[1] does an ad hoc comparison of accuracy improvement / unfairness filtering methods. Maybe mention

Asd Example

2.3 Improvement Techniques for Reputation Systems

Various papers propose methods to make reputation systems more robust against exploitations and improve the accuracy of their reputation calculation. Works with this goal are most commonly in the e-commerce and online recommendation systems domains. A smaller amount also explore other domains.

2.3.1 Categorization

accuracy (better function) vs attack countering they can overlap
categories based on domain mentioned, but not the main driver of categorization here, because looking for general ideas to implement in IoT etc as cross domain better if domain dependent or independent
based on general idea primary categories for this collection
endo vs exogenous mention that it gained some traction in research Whitby et al. group methods to identify and remove unfair ratings into two categories: endogenous and exogenous approaches [2]. Endogenous refers to methods where only the rating values are analyzed for statistical anomalies. Exogenous methods consider factors other than the ratings themselves for detecting unfair ratings. Such external factors are for example the reputation of the rater, in which case the assumption is that low-reputation users are more likely to give unfair ratings.
Endogenous could be expanded or generalized to say only the ratings, reviews, feedbacks, including their value and accompanying text, any data and metadata are analyzed.
[1] uses apart from the endo-exo categorization two other categories: public-private and global-local. These are adopted specifically for e-commerce-type scenarios where local trust (past experiences) and local advisor opinions both play a role.
Public and private differentiated between how advisor opinions are judged for trustworthiness. in private methods, agents evaluate an advisor's opinion based on whether past opinions received by the same advisor turned out to be correct or not. Data for evaluation is limited to the specific advisor's interactions with the specific agent in the past. in public methods, agents judge advisors based on the advisor's ratings across the whole system in the past. Here, all previous opinions of the advisor and their subsequent results are taken into account, not just that which were provided for the agent currently doing the evaluation.
local refers to methods where looking for received unfair ratings of an agent is based on all the ratings that agent received, but not other agents. (E.g. this seller typically gets 4..5 so this 1-rating is suspicious) In a sense, opinions are isolated between agents. in global methods, advisor (interchangeable with rating, feedback, review etc...) trustworthiness is judged using ratings on all the agents in the system.
Although defined for e-commerce advisor-opinion scenario, these categories can be understood in a more general sense. In this case, sellers are to be understood as agents providing content (data, service, product, sensor reading...) and receiving a rating. Vice versa, buyers are receiving the content and providing the rating. Advisor refers to any third party contributing to the final aggregated reputation (rating, trust level) of a potential transaction partner. Opinion is any rating, review or feedback in a more broader sense.
The public-private and local-global distinction can be understood as two sides of the

transaction. In public-private, the rater is judging,
They also identify 4 capabilities a complete improvement approach should have: -majority: work if majority is unfair / malicious -flooding: function under sudden onset of ratings in short time -lack of experience: still work when agents don't have any connections or private XP -varying: agent behavior change should not pose a problem
reactive or proactive. The reactive solutions intend to identify the unfair feedbacks and the proactive solutions propose incentive to the buyers to encourage them to report fair feedbacks [3]
Accuracy of ratings is a crucial part of any reputation system. This is true whether transactions or directly the users are rated. Many attacks exploit exactly this vulnerable point of TRs. Slandering, promoting and other attack strategies, see attacks chapter...
Interpreting accuracy of ratings is difficult in systems where people rate based on their subjective experiences, or where there is a degree of uncertainty. E.g. e-commerce rep systems Since ratings in real-life situations are not accurately computable analogous term is honesty of users. some researchers (cite?) use a scale based on honesty to categorize users from purely malicious to purely honest. In this case this refers to given feedback, as other dimensions of "inhonesty" is also possible, like malice in transaction content. There was a paper that categorized malice into these two ways, cite mayb. Also this discussion mayb not here.
Accuracy's further interpretation is a sort of "trueness" value. This implies that an objectively true and absolutely accurate rating of a user's experience exists. This is then different from the feedback the user gives into the system. The user may or may not be aware of this "true rating." If not aware, maybe bias, subjectively very unpleasant experience. Some call this inherent bias. E.g. e-com the product is perfect but shipping time was bad and gives a 1-star rating. (There is a philosophycal aspect to this, as this is saying "you experienced this a 1/5 but we say it really was a 4/5" and what right or objective measure exists to permit this.) If the user is aware of the "true quality," giving a doctored rating out of malicious intent is still a possibility. In any case, the result is that the given rating differs from a hidden, "objectively true" rating. This situation is called an "unfair" rating by Josang.
Some other metrics measuring accuracy exist in evaluation frameworks. See chapter metrics.
(continuation of above) machine populated reputation systems where ratings are given by algorithms evaluating data, are more fitting to calculations of numerical accuracy

2.3.2 Statistical Reputation Systems

The Whitby et al. work with this line of thought in their paper in [2]. they extend the BRS from a previous paper They argue that unfair and fair ratings have different statistical patterns. Consequently, it is possible to identify and filter unfair ratings with

statistical methods.

TRAVOS is a bayesian trust and reputation method based on the beta probability distribution [4]. It uses both experience-based local trust and global reputation, depending on whether past experience with a particular transaction partner is available or not. If a potential transaction partner is not already well known, the system relies on global reputation reported by third party advisors. Deception from advisors is countered by evaluating the perceived accuracy of their past opinions. Specifically, first a probability is calculated for how likely it is the third party advisor gives an accurate opinion. This is done on the basis of previous opinions given, and the eventual outcomes of those transactions. Then as a second step, the opinions likely to be inaccurate are altered so as to have a smaller impact on the final calculated reputation. This is done through decreasing the parameters of the distribution which represents the advisor's opinion. An opinion modified in this manner will influence the expected value of the final reputation distribution to a lesser degree.

TRAVOS and the Beta Reputation System are very similar. They are both bayesian statistical methods using the beta probability distribution. However, the Beta Reputation System does not differentiate between direct observarions and advisor recommendations. Additionally, the two methods handle inaccurate (unfair, malicious...) ratings differently. TRAVOS is an exogenous approach, scrutinizing each individual advisor based on the perceived accuracy of their past opinions. BRS is an endogenous approach, taking a single rating and judging it based on how far it is from the mainstream opinion.

There is the question of what to do with opinions caught in the filter. Completely disregard, lower weight, or temporarily disregard...

Majority opinion methods rely on the assumption that the majority of the users are honest and accurate.

The Dirichlet Reputation System is worth mentioning as a multinomial generalisation of the Beta Reputation System [5]. Using the dirichlet distribution instead of the beta distribution allows more than two discrete rating levels. It is a method well grounded in bayesian statistics, and is built around similar concepts to the BRS. Specifically, aging of ratings is also present. A similar extension for filtering non-majority outlier opinions like in the BRS was not found for the Dirichlet Reputation System.

The Personalized Approach introduces in [6] ... in e-com, private opinion and global reputation of advisor both used, agents can weight these two, has aging, private: agent take opinion giver, see how they rate commonly rated fourth parties, public reputation based on how consistent the advisor's rating of other parties is

[7] Weighted Majority Algorithm advisors given a weight to sum their opinions, if incorrect based on future transaction, the weight is decreased, constantly wrong advisors are thus filtered eventually

The authors of [6] and [1] take a survey of previous improvement methods, introduce a new method (personalized approach) and compare them, including experimental

measurements. The methods compared are the Beta Reputation System, TRAVOS, Personalized Approach, Bayesian Network Approach and Weighted Majority Algorithm.
[8] "based on the assumption that a dishonest recommendation is one that is inconsistent with other recommendations and has a low probability of occurrence in the recommendation set. Based on this assumption, a new dissimilarity function for detecting deviations in a recommendation set is defined"
[9] in MANETs, dissimilarity function based
[10] uses randomly selected samples of all reviews to calculate global reputation. (e-com)
[11] uses a five-step process to make the reputation system more secure against unfair ratings: Evaluation based filtering, Time domain unfair rating detector, suspicious user correlation analysis, trust analysis based on Dempster-Shafer theory and malicious user identification and reputation recovery.
[12] also surveys methods against unfair ratings. Considered approaches are filtering based on majority opinion (Whitby et al), entropy-based filtering, reputation trees, (agents reputation is their rating's weight) controlled anonymity and agent clustering, clustering ratings and checking IP address, using trust rating in MANETs and WMA and belief function.
[13]
"Transactions are evaluated by both the source peer and the target peer in [9]. A source peer's feedback is considered consistent if it agrees with the target peer's self-evaluation. Assuming most of the peers are trustworthy and honest, most inconsistencies would be in the case that the source peer reports a trustworthy transaction as untrustworthy (badmouthing) or an untrustworthy transaction as trustworthy (collusion to boost the target peer's reputation). Thus, a source peer is suspected to be providing false feedbacks if the proportion of inconsistent feedbacks exceeds certain threshold, T. Feedbacks from inconsistent peers are no longer accepted. " WARNING, this is a quote of consistency based
"The reputation system in [15] uses a measurement called credibility factor. The credibility factor increases if a recommender provides a recommendation that matches the actual result of the transaction. From the credibility, discredibility factor can be derived. A recommender whose discredibility factor is higher than its credibility factor will be filtered out." WARNING quote of credibility based
[14] for e-com, introduces two extension ideas to TRs to combat unfair ratings: controlled anonymity to avoid unfairly low ratings and negative discrimination cluster filtering to reduce the effect of unfairly high ratings and positive discrimination
[15][16] work with the idea of normalizing the rating scale each agent is typically using. This approach comes from collaborative filtering systems.
[17] "finds similar agents and consider them as reputable source of information. Further, the agents compare their trust values with trust values of similar agents"
explicit inclusion of private experience as a system component possibility of maintaining

a trust network and getting recommendations (opinions) from them as advisors
both of these are done in a system primarily used by men. e.g. in e-commerce, once the reliability and quality of a provider is known through first-hand experience, reviews and ratings are not scrutinized as they would be by first-time buyers. Social proof mechanisms like getting recommendations and opinions from friends or trusted contacts is also very common without it needing to be part of any reputation system.
This is different in reputation systems made for and used by algorithmic agents. whether decisions are made by other machine agents or the system (algorithm) itself like in a traffic management scenario, it will only ever consider factors it was programmed to. There is a case for explicitly integrating these human mechanisms or similar methods inspired by them. The potential upside is better reputation estimate accuracy, or at the least better decision outcomes. On the other hand, the increased complexity provides new attack vectors and thus potentially enables new exploits.
Mechanisms like a list of trusted "friends," trust networks, third party advisors providing private opinions come from a trust-centric approach to the decision making problem. Three separate categories give themselves from this distinction: pure reputation systems, reputation systems infused with trust components and pure trust systems. Only the first two is of interest within the scope of this thesis.
With this, the categorization of local-global and public-private is also easier to define in a broader context.
Provide incentives to prevent or mitigate unfair ratings: [3] [18]
[19] recovers the temporary damage of sellers who were given a lower rating by a buyer by mistake, and the buyer willingly corrects it. The seller's reputation is temporarily inflated to revert reputation damage.
[20] aims to improve unfair rating detection by analyzing selected factors of the environment the detection is deployed in and selecting the most suitable approach from a multitude of methods.
learning approaches: [21] [22] Bayesian Network-Based Trust Model, p2p, reinforcement learning [23]
approaches combining first hand trust observations with reputation in some way: [24] for manets, everyone maintains a first-hand reputation (i.e. trust) and a second-hand (global) reputation of others. First-hand reputation is exchanged by nodes occasionally, and data is combined to reach a better accuracy. The system employs aging and occasional re-evaluation to enable reputation redemption. [25]
regarding recommender systems: [26]

2.4 Attacks and Exploits

3

Approach

3.1 Improving Reputation Estimates

3.1.1 Categorization

other cateogry: what is rated
agents content links could call this rating dimension aka who/what is rated exactly can represent as graph (vertices, edges, directed boxes as content) discuss special cases, e.g. in e-com, rating the very same physical products between different sellers, an extra rating dimension is neede for service quality (packaging, mayb its white/gray label, shipping time, customer service interaction's quality etc... this is different than the product rating and also diff from the agent rating itself also accuracy rating

3.1.2 next todo

there is a case to be made about applying improvement techniques depending on the environment of the reputation system. e.g.
rating aging makes sense only if behavior are known to change over time. if for sure constant (like machines which receive no updates whatsoever) then it has no point
frequency of feedback rating given after transaction.. means only something maybe if its not mandated, which in a fully machine network might as well just be
outlier filtering makes sense only if you know that significant (extreme) differences in transaction experiences are unlikely to the point that such claim becomes suspicious. Once highly subjective experiences come into play, it is wrong to say a nonmajority opinion is unfair. Or in a highly dynamic agent population, where behavior changes often and rapidly, extremity filtering is contraproductive since it introduces reputation lag.
differenciate based on how many ratings contributed to a final reputation score, i.e. confidence. like if a new entrant gets a default of 4, and another has 4 based on a 1000

ratings, this difference should be noticable and agents should be able to make decisions based on that

3.1.3 Outlier Filtering based on Majority Opinion

3.1.4 Aging Ratings

3.1.5 Normalizing Rating Range

3.1.6 Weighing based on Rater Reputation

3.1.7 Anonymous ratings

outlier filtering based on something: statistical, entropy, dissimilarity function, similarity dis/similarity is also a kind of stereotype-based TR, also collaborative filtering ties in here normalizing rating range weighing based on rater reputation anonymous ratings adding local trust use random subsets to calculate reputation Weighted Majority Algorithm simplifying rating space (some believe that +- or only + or - helps eliminate attacks) feedback consistency-like solution – in intersection, car estimates ETA, later rates its own ETA, and see if others rate the same ETA consistently – inspired by one menitoned in [13]
pair ETAs with uncertainty (generally: stake) if I can say along with the claim how sure i am in the claim or maybe once when I commission a car, I can say how accurate the sensor reading is, so this accuracy rating remains static for a long time then if this uncertainty is larger, I have more room to manipulate the ETAs within plausible deniability, but on the other hand the other cars can say my claim is more worthless because I have a large uncertainty range..
also in this part: author reviews

3.2 Pyrepsys Evaluation Framework

check how to implement these can use existing eval frameworks? or need own one in python if latter, what components a diagram can be good also for the presentation
The Pyrepsys Reputation System was designed in order to allow simulation, comparison and evaluation of selected reputation schemes with various environment conditions. This chapter gives a conceptual introduction for Pyrepsys, including main design goals, simulation flow and core features. For a description of the concrete implementation of Pyrepsys, see Chapter 4.
Overall requirements and unique approaches warranted the inception of a new reputation simulation framework. These are among others:

- domain-independence

- the use of one or more reputation imporvement methods
- ability to represent claims
- possibility of second-order rating schemes like stakes
- full simulation of agents
- flexible review values (to support custom integer ranges)
- extendability and ease of modification for fast prototyping

3.2.1 Design Goals

ASD

Flexibility and extendability are among the main design objectives of pyrepsys. Variables can be altered in scenario configuration files. The size and composition of the agent population is customizable. New agent behaviors, metrics, reputation calculation and imporvement methods can be easily added.
Pyrepsys is a one-stop solution for simulating, comparing and evaluating reputation schemes. Simulation of scenarios is performed on a round-by-round basis. Shared configuration among scenarios, simulating batches of scenarios in succession and controlled random number generation help compare different scenarios. Evaluation is aided by metrics. These can automatically generate graphs and export data from simulations.
Miscallaneous supportive functionalities are also discussed in this chapter. A scenario configuration creator can be used to make a batch of scenarios with variations along selected configuration parameters. Extensive automated self-testing can verify the integrity of the simulation framework if any changes are made. Finally, profiling and benchmarking tests are provided to identify simulation bottlenecks and measure performance.

3.2.2 Scenarios

A scenario is the description of a complete simulation environment including reputation schemes and agents with various behaviors. All parameters that are followed during the simulation are part of the scenario. Some of the most important are listed below.

- how reputation values are calculated
- which reputation improvement methods are applied
- the size of the agent population
- how agents behave during simulation
- how long the simulation goes (i.e. how many rounds)
- the possible rating and reputation values (e.g. binary, integers 1 to 5)
- what seed is used to generate random numbers

- what graph and data exports metrics should be created

In this way, a scenario encapsulates all that is in a single simulation. This helps organizing combinations of simulation parameters and comparing them. For example, when the goal is to see what effect malicious agents have, one could draft separate scenarios with varying percentage of malicious agents ranging from 0% to 100% in each.
In the Pyrepsys implementation, it is possible to describe scenarios without a full specification. This means not all needed parameters are listed, just some selected ones. In this case, the missing parameters are taken from a default scenario configuration. This is useful when the effect of one or more parameter changes are needed, since those can be simply listed alone in their scenario configuration. This is described along the implementation in chapter 4. For all conceptual discussions, a scenario refers to all the simulation environment fully specified, regardless of where they come from in the implementation.

3.2.3 Simulation Flow

Simulation in Pyrepsys is built around scenarios. Each invocation consists of simulating one or more scenarios after each other. At the beginning, the default scenario configuration is fetched. Then Pyrepsys reads and applies the individual configuration for each scenario. The scenario is simulated after these preparations. Once all scenarios are finished, the results processor module exports collected data and draws graphs based on the selected metrics. The simulation part's schematic flow is shown in figure 2.
Scenario simulation is performed on a round basis. The number of rounds is part of the scenario configuration. Each round consists of four distinct phases:

1. claiming
2. rating of new claims
3. applying reputation improvements
4. reputation calculation

CLAIMING

Claiming is the part where agents can make new claims. During this phase, all agents get an opportunity to make one new claim. Whether an agent ends up publishing a claim or not depends on multiple factors.
First of these is the agent's claiming probability. This represents the likelihood of them attempting a claim if given an opportunity. A more consumer-type agent would have a lower probability and thus claim rarely if ever. On the other hand, producer-type agents would tend toward higher probabilities, and claim more often. The random check whether agents attempt a claim or not is performed on the main random chain.

If the agent passes this check, a claiming process begins. First, a claim is generated containing a random ground truth. The ground truth is not directly accessible for the agent. It can however be measured, which is the second step of the claiming process. Measurement results in the measured claim score, which represents an approximation of the claim's true quality. How good this approximation is, depends on the agent's ability to assess claims. Further details of measuring claims is described in section 3.2.7.
With the measured truth, the agent executes its distortion strategy. It essentially applies one of the distortion methods found in section 3.4. The resulting value is the distorted claim quality.
At this point the agent checks whether the distorted quality falls in the agent's claim range. Claim range serves as a minimum and maximum limit as to what claims an agent is willing to publish. If the distorted claim quality falls outside these limits, the claim is discarded and claiming is aborted. In this case, the agent will not claim in this round.
If the limits are not violated, the new claim will be published. As a last step, the agent creates an author review. Author reviews are special reviews appended to each claim by the claiming agent, the author. The review value is always the distorted claim quality from the previous step.
- idea of author review - what each thing represents - accuracy is the agents competency "expertness" / unwilling inaccuracy - distortion is honesty / willing inaccuracy - author review is a declaration on what the claimer thinks his claim is worth
TODO continute from here
claim range

3.2.4 Claims

describe claims and what means what

3.2.5 Reviews

describe reviews and what means what

3.2.6 Random Number Generation

reproducability: rng main chain sub chains for cases where - dont know if they need a random - they need and dont know how many - they need and dont know what kind (distribution, range etc) see notes

3.2.7 Measuring Claims

measuring claims

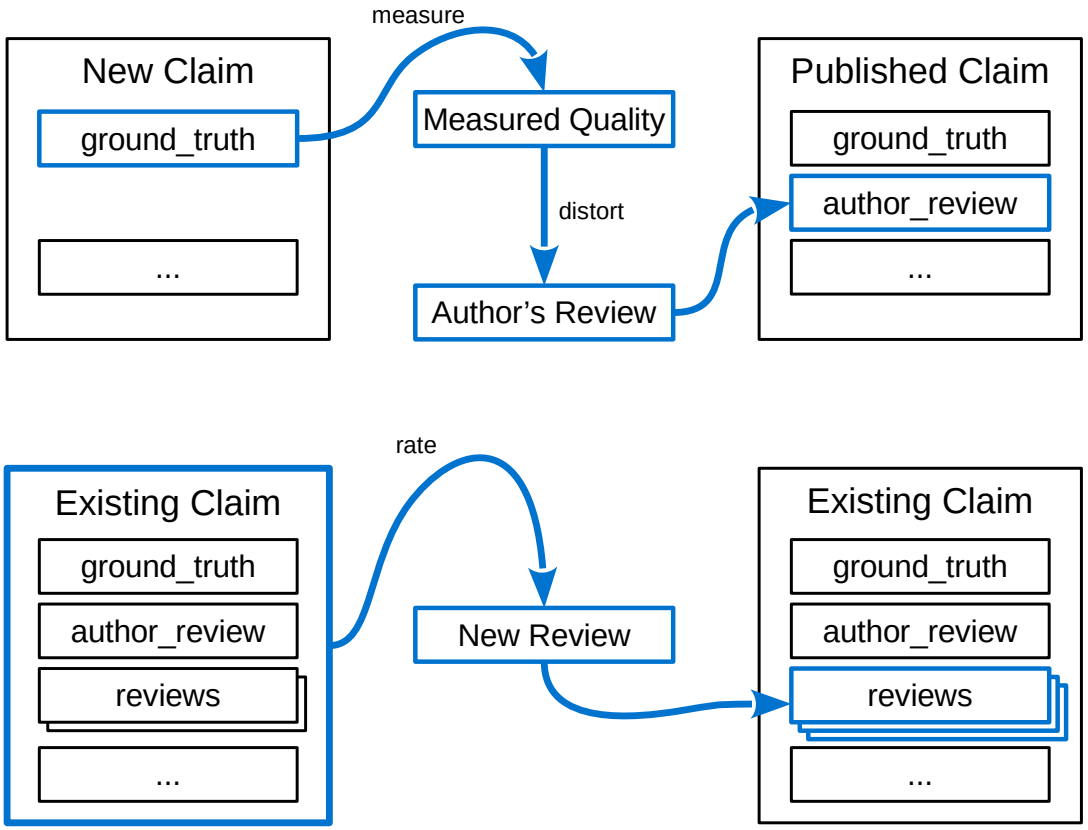

Figure 1: Schematic representation of claiming (above) and rating (below) processes with the involved claims and reviews. *Claiming* (above) spawns a new claim without any reviews, containing a random ground truth. The claimer agent measures this hidden quality and distorts it, resulting in a value representing the clamer's opinion of his own claim. This is attached as an author review to the claim to be published. *Rating* (below) takes an existing published claim. The rater agent takes all data relevant to the claim and produces a review. This is then appended to the other reviews on the same claim.

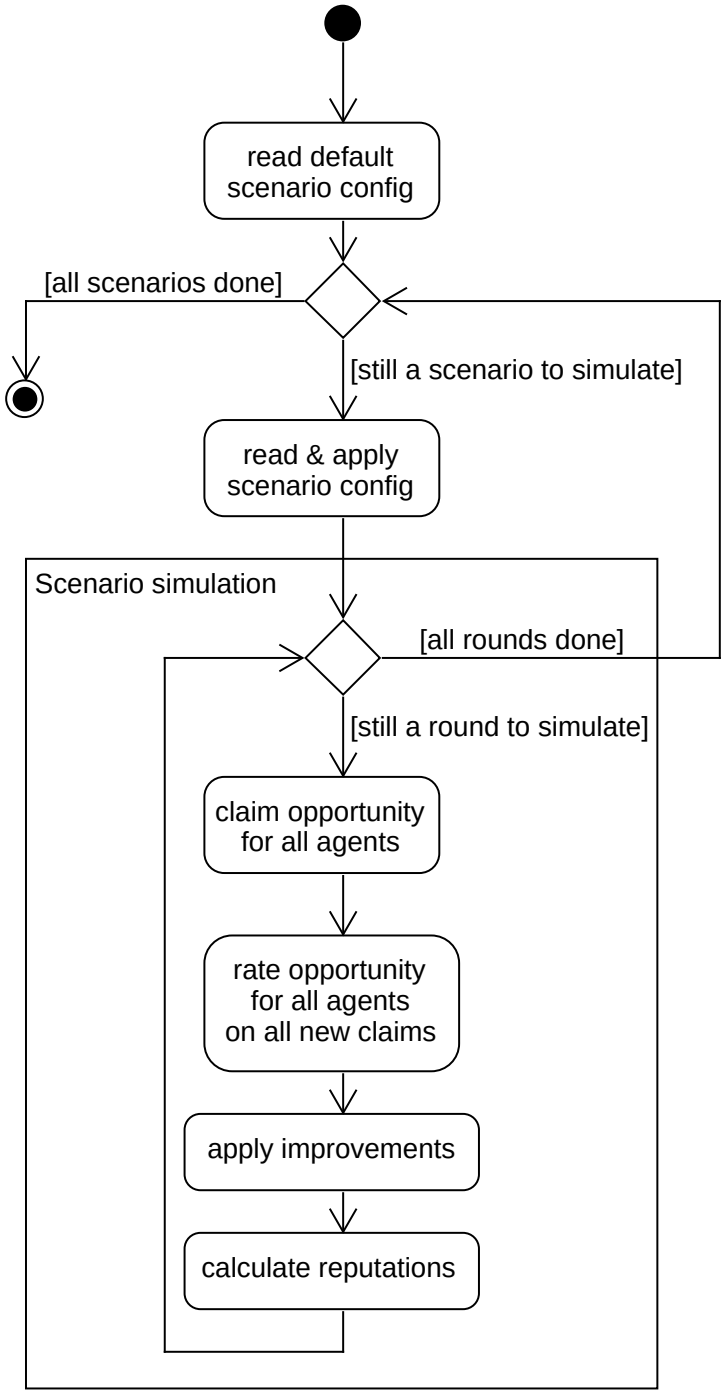

Figure 2: Simulation flow in Pyrepsys. A single simulation unit is the scenario. Each scenario is simulated independently after one another. Rounds make up the simulation within a scenario. Each round consists of four distinct phases: claiming, rating, improvement and reputation calculation. The number of rounds is specified in the configuration.

3.3 Reputation Calculation

BasedOnAvgDifferenceOfClaimsAndReviews ReputationAverageStrategy
why these two: - simple - thus easy to know what is happening - avg is a common-sense approach that a common user would expect when seeing a star-system (even tho that is often not the case in the rep calc, just implied) - BasedOnAvgDifferenceOfClaimsAndReviews uniquely uses the author-review system of claim rating the difference in their interpretation ReputationAverageStrategy: a rating becomes a quality of the claimer other one: review rates what this claim should have been rated by the author ability to easily incorporate weights

3.4 Distortion Strategies

[7] has the four static in the graph.. static: "pessimistic" rater (rates in the lower range) bad mouthing "optimistic" rater (rates in the higher range) ballot stuffing inverted rating (rates good as bad and vice versa) random rater (rates random all the time)
user tendentially rating extreme ... conservative random error rater (occasionally makes a significantly inaccurate rating, most ratings honest to a degree of noise, i.e. not perfect!) big inaccuracy e.g. erroneous measurement,sensor
Seller sells most products with high quality, but some with low quality for benefit. The honest bad reputation report from victim may be submerged, or be mistakenly regarded as unfair rating. [27]
dynamic: build up honest reputation and attack after that single out an agent and distort their rating only, honest otherwise mayb combine negative and positive attacks in collusion scenario, where attackers uprate each other and downrate their target mayb make difference bw. targeted attack on someone or group and just wrecking havoc on all individual vs collusion scenarios

3.5 Rating Strategies

4

Implementation

Simulations are done with a custom-made python simulation framework named pyrepsys. This chapter gives the implementation-specific details of pyrepsys. For a conceptual description, see Section 3.2.

Pyrepsys is a one-stop solution for simulating, comparing and evaluating reputation schemes. Simulation of scenarios is performed on a round-by-round basis. Shared configuration among scenarios, simulating batches of scenarios in succession and controlled random number generation help compare different scenarios. Evaluation is aided by metrics. These can automatically generate graphs and export data from simulations.

Flexibility and extendability are among the main design objectives of pyrepsys. Variables can be altered in scenario configuration files. The size and composition of the agent population is customizable. New agent behaviors, metrics, reputation calculation and imporvement methods can be easily added.

Miscallaneous supportive functionalities are also discussed in this chapter. A scenario configuration creator can be used to make a batch of scenarios with variations along selected configuration parameters. Extensive automated self-testing can verify the integrity of the simulation framework if any changes are made. Finally, profiling and benchmarking tests are provided to identify simulation bottlenecks and measure performance.

Additional references for using Pyrepsys are provided outside this chapter. For a description of the pyrepsys command-line interface, refer to Appendix B. A complete list and explanation of possible scenario configuration parameters are found in Appendix A.

TODO: prerequisites, dependencies, tested versions, environment

4.1 Simulation Flow

general simulation flow +flow diag claiming process rating process measuring claims improvement handling process reputation calculation (for completeness' sake)

extendable strategies: distort, rate, reputation + metric (adding new ones too?)

4.2 Data Handling

data structures: agents, claims, reviews (classes) + metrics (as they will) internal vs ae data storage precision/resolution domains handling

4.3 Random Number Generation

4.4 Results Processing

After the simulation is finished, Pyrepsys processes the collected data and exports it as graphs or tables. Collection during sim events
artifacts metrics graph data reproc event based subscribe-notify + uml

4.5 Configuration

configuration (process) where are configs are stored (responsibilities) (config.get, local caches, local configs) reading in 2-leveled: default and active config config updated callbacks base behaviors extended entries / entries extended with settings CHECK: all big config options covered
mention: scenario creator, link appendix

4.6 Other Facilities

Automated Self-Testing performance other facilities (logging, error handling, helpers, paths?)

5

Evaluation

5.1 Metrics

- reputation - avg tot claim inaccuracy - also other metrics from my notebook

5.2 Setup

simulation setup – used settings, scenarios describe different simulation bundles (of scenarios) and their rationale why they are interesting and what's expected

5.3 Results

5.4 Discussion

6

Conclusion

We successfully ...

6.1 Outlook

But we still need to ...

A

Scenario Configuration Options

TODO comprehensive list of all scenario parameters

B

Command Line Interface and Invocation

TODO extensive CLI description

C

Scenario Creator

TODO SC desc

Glossary

author review TODO. 13, 14

claim TODO. 11, 12

claim range TODO. 13

claiming probability A percent likelihood determining whether an agent initiates a claiming process if given an opportunity. Specified in scenarios for each agent.. 12

distorted claim quality TODO a value produced by the distort strategy from the measured claim q. 13

distortion strategy TODO. 13

ground truth TODO. 13, 14

main random chain TODO. 12

measured claim score TODO. 13

review TODO. 13, 14

scenario Description of a complete simulation environment for Pyrepsys. Specifies the reputation scheme, improvement methods, agents, possible review values etc. See section 3.2.2 for details.. 11, 12, 15

Bibliography

[1] J. Zhang, M. Sensoy, and R. Cohen, "A Detailed Comparison of Probabilistic Approaches for Coping with Unfair Ratings in Trust and Reputation Systems," in *2008 Sixth Annual Conference on Privacy, Security and Trust*, (Fredericton, Canada), pp. 189–200, IEEE, Oct. 2008. cited on p. 3, 4, 6

[2] A. Whitby, A. Jøsang, and J. Indulska, "Filtering Out Unfair Ratings in Bayesian Reputation Systems," in *Proceedings of the Workshop on Trust in Agent So- cieties, at the Autonomous Agents and Multi Agent Systems Conference*, p. 12, 2014. cited on p. 4, 5

[3] S. Thakur, "A reputation management mechanism that incorporates accountability in online ratings," *Electronic Commerce Research*, vol. 19, pp. 23–57, Mar. 2019. cited on p. 5, 8

[4] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck, "TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources," *Autonomous Agents and Multi-Agent Systems*, vol. 12, pp. 183–198, Mar. 2006. cited on p. 6

[5] A. Josang and J. Haller, "Dirichlet Reputation Systems," in *The Second International Conference on Availability, Reliability and Security (ARES'07)*, (Vienna, Austria), pp. 112–119, IEEE, 2007. cited on p. 6

[6] J. Zhang and R. Cohen, "A Personalized Approach to Address Unfair Ratings in Multiagent Reputation Systems," p. 10, 2006. cited on p. 6

[7] B. Yu and M. P. Singh, "Detecting Deception in Reputation Management," p. 8, 2003. cited on p. 6, 16

[8] N. Iltaf, A. Ghafoor, and U. Zia, "A mechanism for detecting dishonest recommendation in indirect trust computation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, p. 189, Dec. 2013. cited on p. 7

[9] Zakirullah, M. H. Islam, and A. A. Khan, "Detection of dishonest trust recommendations in mobile ad hoc networks," in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, (Hefei, China), pp. 1–7, IEEE, July 2014. cited on p. 7

[10] M. Rezvani and M. Rezvani, "A Randomized Reputation System in the Presence of Unfair Ratings," *ACM Transactions on Management Information Systems*, vol. 11, pp. 1–16, Apr. 2020. cited on p. 7

[11] A. Baby, A. Kumaresan, and K. Vijayakumar, "A Secure Online Reputation Defense System from Unfair Ratings using Anomaly Detections," *International Journal of Computer Applications*, vol. 93, pp. 17–21, May 2014. cited on p. 7

[12] L. Ngo, "A survey of unfair rating problem and detection methods in reputation management systems," p. 12, 2007. cited on p. 7

[13] F. Azzedin, "Identifying Honest Recommenders in Reputation Systems," p. 7, 2010. cited on p. 7, 10

[14] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd ACM conference on Electronic commerce - EC '00*, (Minneapolis, Minnesota, United States), pp. 150–157, ACM Press, 2000. cited on p. 7

[15] D. Margaris and C. Vassilakis, "Improving Collaborative Filtering's Rating Prediction Quality by Considering Shifts in Rating Practices," in *2017 IEEE 19th Conference on Business Informatics (CBI)*, (Thessaloniki, Greece), pp. 158–166, IEEE, July 2017. cited on p. 7

[16] D. Margaris and C. Vassilakis, "Improving Collaborative Filtering's Rating Prediction Accuracy by Considering Users' Rating Variability," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, (Athens), pp. 1022–1027, IEEE, Aug. 2018. cited on p. 7

[17] E. Zupancic and D. Trcek, "QADE: A Novel Trust and Reputation Model for Handling False Trust Values in E-Commerce Environments with Subjectivity Consideration," *Technological and Economic Development of Economy*, vol. 23, pp. 81–110, Jan. 2015. cited on p. 7

[18] R. Jurca and B. Faltings, "Minimum payments that reward honest reputation feedback," in *Proceedings of the 7th ACM conference on Electronic commerce - EC '06*, (Ann Arbor, Michigan, USA), pp. 190–199, ACM Press, 2006. cited on p. 8

[19] S. Liu, C. Miao, Y. Liu, H. Fang, H. Yu, J. Zhang, Y. Chai, and C. Leung, "A Reputation Revision Mechanism to Mitigate the Negative Effects of Misreported Ratings," in *Proceedings of the 17th International Conference on Electronic Commerce 2015 - ICEC '15*, (Seoul, Republic of Korea), pp. 1–8, ACM Press, 2015. cited on p. 8

[20] C. Wan, J. Zhang, and A. A. Irissappane, "A Context-Aware Framework for Detecting Unfair Ratings in an Unknown Real Environment," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, (Macau, China), pp. 563–567, IEEE, Dec. 2012. cited on p. 8

[21] A. Khoshkbarchi and H. R. Shahriari, "Coping with unfair ratings in reputation systems based on learning approach," *Enterprise Information Systems*, vol. 11, pp. 1481–1499, Nov. 2017. cited on p. 8

[22] Y. Wang and J. Vassileva, "Bayesian network-based trust model," in *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, (Halifax, NS, Canada), pp. 372–378, IEEE Comput. Soc, 2003. cited on p. 8

[23] A. Yazidi, B. J. Oommen, and M. Goodwin, "On Solving the Problem of Identifying Unreliable Sensors Without a Knowledge of the Ground Truth: The Case of Stochastic Environments," *IEEE Transactions on Cybernetics*, vol. 47, pp. 1604–1617, July 2017. cited on p. 8

[24] J.-Y. L. Boudec and S. Buchegger, "A Robust Reputation System for Mobile Ad-hoc Networks," tech. rep., 2003. cited on p. 8

[25] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "On Handling Inaccurate Witness Reports," p. 15, 2005. cited on p. 8

[26] A. Jøsang, G. Guo, M. S. Pini, F. Santini, and Y. Xu, "Combining Recommender and Reputation Systems to Produce Better Online Advice," in *Modeling Decisions for Artificial Intelligence* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, V. Torra, Y. Narukawa, G. Navarro-Arribas, and D. Megías, eds.), vol. 8234, pp. 126–138, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Series Title: Lecture Notes in Computer Science. cited on p. 8

[27] Ping Xu, Ji Gao, and Hang Guo, "Rating Reputaion: A Necessary Consideration in Reputation Mechanism," in *2005 International Conference on Machine Learning and Cybernetics*, (Guangzhou, China), pp. 182–187, IEEE, 2005. cited on p. 16