EVALUATION OF TRUST POLICIES BY SIMULATION

Cosmin Mogoş and Ina Schieferdecker
ETS, TU Berlin, Berlin, Germany

Keywords: Trust modeling, Trustworthiness, Trust evaluation, Simulation.

Abstract: The evolution of the World Wide Web has created a new environment where people can interact, e.g. talk to their friends, shop online, conduct business meetings, etc. Trust and trustworthiness are central notions in human interaction; in particular, they represent important criteria for every Internet user because of the multitude of choices they are faced with when choosing whom to interact with.
This paper presents a simulation model implemented in Ptolemy II for the simulative analysis of trust policies in networked environments, called communication space (CS). The model reflects both the CS structure e.g. principals, roles, and event structure, and the interactions between the elements of CSs. Principal behavior is based on Markov chains and on criteria for selecting peers that initiate transactions. We investigate the efficiency of trust policies based on local observations and evaluation of interactions by examining a case study based on the popular auction site eBay.

1 INTRODUCTION

The adoption of Internet technologies in almost every private, economic and social sector made trust and trustability central notions of networked environments, called communication spaces (CS). The notion of communication space (CS), in opposition to isolated single communication mechanisms, has recently been proposed as a conceptional guideline for a multidisciplinary project on Human-Centric Communication at TU Berlin, which expands initial concepts on I-centric communication (Arbanowski et al., 2004). Users of CSs make use of offered services only if they perceive that they can trust the service providers and the involved technologies.

There has been extensive work done towards formalizing trust, but there is no generally accepted method of evaluating trust models. The Prisoners' Dilemma (Axelrod, 1984) was used by several authors to test their models, but such approaches do not reflect the complexity of virtual environments and are also inherently confrontational. (Schlosser et al., 2005) has proposed a framework for evaluating reputation systems that are freely configurable by the user, but because the focus is on reputation, direct experience cannot be represented by the framework. The ART testbed (Fullam et al., 2005) proposed a framework to compare multiple trust modeling algorithms in competition with each other, however because the best trust model is chosen based on an overall score, evaluating separate trust aspects is difficult.

As trust is largely built from experience, trust models have been developed to reflect this concept. Trust policies are therein used to formalize the evaluation of experiences with respect to trust. The results of trust policies are however hard to assess in networked environments where typically a large number of predominantly unknown principal identities are engaged in interactions. It is open how to select trust policies that result in a more successful choice of trusted parties to interact with.

Therefore, we developed a simulation framework within Ptolemy II (T et al., 2003) to analyze and evaluate the dynamics of trust policies in face of flexible user-to-user interactions. The simulation framework is based on the formal SECURE framework (Krukow, 2006), which builds upon event structures to model possible outcomes of interactions. In (Eilers and Nestmann, 2009), SECURE was extended with a flexible way to determine a degree of trust from given past behavior, and a basic notion of context, exemplary in the form of roles the interacting parties may occupy. (Eilers and Nestmann, 2009) constitutes the basis for our trust simulation framework.

The paper is structured as follows. Section 2 presents the simulation approach being developed and Section 3 discusses our case study and first results. A conclusion summarizes the paper and discusses ongoing work.

Mogoş C. and Schieferdecker I. (2010).
EVALUATION OF TRUST POLICIES BY SIMULATION.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents*, pages 178-183
DOI: 10.5220/0002720501780183

2 THE TRUST SIMULATION APPROACH

2.1 Trust Model

There have been several approaches towards a formalization of trust each focusing on a different aspect. Trust models can be classified using a number of criteria where three aspects are particularly relevant (Ries S., 2006): trust value domain, dimensions, and semantics. The simulation framework uses the trust structure defined in SECURE to capture these aspects. A trust structure is a triple $T = (D, \sqsubseteq, \preceq)$ where D is a set of trust values ordered by two partial orders: the trust ordering ($\preceq$) and the information ordering ($\sqsubseteq$).

Within the SECURE framework experience gathered from interaction with other principal identities is stored in a local interaction history as observed events from a mathematical structure called an event structure. In (Eilers and Nestmann, 2009) context information (e.g. role, session etc.) is added to the local interaction history and a *justification language* is introduced to derive trust from experience.

2.2 Simulation Framework

We use the SECURE trust structure and the justification language (from the extended SECURE model) to represent trust models in the simulation framework whose main focus is the evaluation of trust policies.

While an effective trust policy should adapt to different scenarios, some may be better suited than others given a fixed scenario. The simulation framework provides tools to define different scenarios in which to evaluate trust policies. A scenario has the following components: communication space, trust policies, principal behaviors, and scenario setup. The components are generic and can be be combined in different configurations to generate new scenarios.

We use the following properties to represented communication spaces within which the principals interact:

Roles. The set of roles which principals can take in a CS. Principals can have one or more roles. Each role can either produce a ”service” or require one (”client” role), e.g. in a hospital CS a doctor could provide ”consultation” and a patient could require it. The ”service” is not limited to an economical product, it could e.g. be ”advice” in the case of a family communication space.

Trust Structure. Facilitates a common understanding of trust values for the principals in a CS.

Event Structure. Describes the possible actions that can be taken by different principals.

Trust policies are defined using the justification language from the extended SECURE model and are closely related to the event structure and trust structure defined within the CS. Principal behavior is defined using finite state Markov chains (Kemeny and Snell, 1983). Fig. 3 shows the model for a reliable seller which is described in detail in Section 3. Each state of the behavior model consists of one or more actions. An action reflects the behavior of the principal when observing an event. Each action has a trigger and a set of events to be generated when the trigger is observed. One state cannot have two actions with the same trigger but every state should cover all the possible triggers.

The simulation scenario setup defines how the different components work together. It has the following parameters:

Principal Types. Selects the roles, behaviors (for each role) and policies (defined using the justification language) that a principal of the type will have. If a principal has ”client” roles than the threshold for interaction with a producer is also defined.

Principal Setup. Determines the number of principals of each type that will be simulated, it also sets the *number of interactions*, *time of first interaction*, *interaction frequency* and *behavior change frequency* for those principals

Interaction Initialization. The events that have to be triggered in order to initiate an interaction between two principals.

Event Transformations. Because communication takes place through a CS, principals do not directly observe the actions of their partner, they observe the CS responses. This parameter determines how events generated by a principal are observed by another.

Different scenarios can be defined in order to evaluate different properties of trust policies. For example the scalability of a trust policy can be verified by increasing the number of principals; the accuracy of a trust policy can be evaluated by increasing the number of malicious or incompetent principals. Also the framework can be used to evaluate one policy within different scenarios or to compare different policies within the same scenario.

2.3 Interaction Model

The previous section described how scenarios can be defined, in this section the simulation engine is out-

lined. Experience is represented in the extended SECURE model by recording observed events in a local interaction history. The simulation framework also uses events as the building blocks of principal interaction. A principal can either observe an event or generate one as a response to an observed event. If a principal A generates an event in an interaction with principal B the simulation engine translates it (based on the *Event Transformations* parameter) into an observed event for principal B. This is necessary because the two principals do not interact directly and cannot observe the exact actions of their partner; they can only observe the results. For example principal A could ship the product on time but because of a mix up with the post office the package never arrives at principal B. All that principal B can observe is that the package never arrived which may falsely lead to flagging principal A as untrustworthy, even if the fault lies within the technical system used. All the events that can be generated/observed in a communication space are defined in the *Event Structure* parameter of the CS.

Principals interact with each other based on the roles they have, when a principal requires a service it will query the simulation engine for providers of that service. Out of the available providers the one with the highest associated trust value is selected for the interaction. The selected principal must have an assigned trust value higher then the threshold defined in the *Principal Setup*. If there are more principals with the same trust value the one with which the consumer interacted the most will be selected, because the trust value is based on a larger information base. Finally, if more than one candidate remains one is selected at random. Also, if there is no information available about providers one is selected at random.

When a consumer chooses a provider it signals the choice to the simulation engine which initiates a session (interaction) between the two. By session we understand one atomic interaction between two principals. In order to initiate a session the simulation engine searches for the appropriate events (from the *Interaction Initialization* parameter) to be triggered based on the roles of the interacting principals. After the initialization events are triggered, each principal will respond to observed events according to it's behavior. A session is considered to end when neither principal has events to generate. During the interaction the simulation framework updates the local interaction histories of the two interacting principals.

Time is an important factor when determining a principal's trust, for example a positive event observed a year ago can have little importance if the most recent observed events indicate the principal is now untrustworthy. Also a few attacks (Sun et al., 2006) on trust evaluation are based on changing behavior at different points in time hoping that old untrustworthy behaviors have been forgotten. The simulation framework represents time as a natural number, this allows a time unit to reflect different amounts of time (second, week, year, etc.) depending on the scenario.

Interactions between principals can begin at any time and more than one session can be opened for each principal at a given time. The main parameters that determine when interactions take place are *time of first interaction*, *number of interactions* and *interaction frequency*. The parameters are only relevant to principals that have consumer roles, a principal begins initiating sessions at *time of first interaction* and after *number of interactions* it stops. When all principals have finished their interactions the simulation ends. By adjusting the parameters, the effects of two CS properties can be examined (Sabater, 2004): *Principal Interaction Frequency* and *Encounter Factor*. The former property determines how often principals interact within the CS, and is important because depending on the frequency it may be easier or harder for principals to derive trust from direct experience. The latter parameter reflects the probability that two principals will interact several times, this parameter is closly related to the number of principals and the *Principal Interaction Frequency*.

3 CASE STUDY AND FIRST RESULTS

The case study is based on the popular virtual marketplace eBay. Users of eBay are required to create an account in order to use the full functionality of the site, this allows their identity to be confirmed whenever they take an action. We are going to assume that the identification works perfectly, i.e. users are always who they claim to be.

Users on eBay can search for auctioned items, bid on existing auctions, or create their own auctions. In order to focus on how trust is derived from local observations, only the interactions that take place after a user wins an auction are simulated. Users can be in one of two roles: *seller* and *buyer* which trade one product type. It is considered that each *seller* has an infinite supply of products and can handle any number of simultaneous interactions. As a result when a *buyer* requires a product, she selects the *seller* she trusts the mosts and it is considered that the buyer won an auction initiated by the *seller*, so the simulation engine sends the appropriate events to initiate the interaction.

Fig. 1 illustrates the events that can be observed

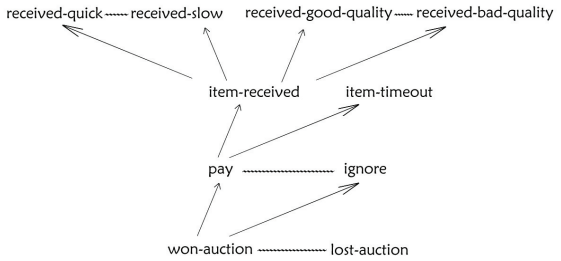

Figure 1: Buyer event structure.

by a *buyer*, the arrows represent dependencies and the wavy lines represent conflicts. The events that can be observed by the *seller* are complementary, e.g. `pay` is observed as `payment-received` by the seller. Having different event structures allows the simulation of cases when the *buyer* pays, but because of a bank error the payment does not reach the *seller*.

We chose five trust values to represent the reliability of a *seller* as perceived by a *buyer*, illustrated in Fig. 2. The figure also shows the trust ordering, the value **no-info** indicates that there is no information about a *seller*, and is considered higher than **item-will-not-arrive** since there is the possibility of encountering a trustworthy *seller*. As for the information ordering all trust values are greater than **no-info**, and are unrelated to each other.

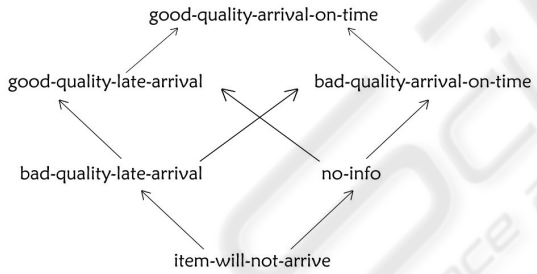

Figure 2: Client trust structure.

The following justifications were chosen to derive trust from local observations:

- **item-will-not-arrive** - always true
- **bad-quality-late-arrival** - if in all the sessions in which the buyer payed the item has arrived, and at least in one session the item was received slow and at least in one session the item was of bad quality
- **bad-quality-arrival-on-time** - if in all the sessions in which the buyer payed the item has arrived quickly but at least in one session the item was of bad quality
- **good-quality-late-arrival** - if in all the sessions in which the buyer payed the item has arrived and was of good quality but at least in one session the item was delivered late
- **good-quality-arrival-on-time** - in all sessions in which the buyer payed the item was received quickly and was of good quality

If the local interaction history is empty than **no-info** is justified.

The metric used to evaluate the policies is the rate of successful interactions, an interaction is considered successful when the outcome is the one expected by the *buyer*, if the *buyer* does not have any more *sellers* it trusts all remaining interactions are considered to be failed. In the case of a buyer with the threshold set to `good-quality-late-arrival`, an interaction is successful if the *buyer* pays and the item is received and of good quality.

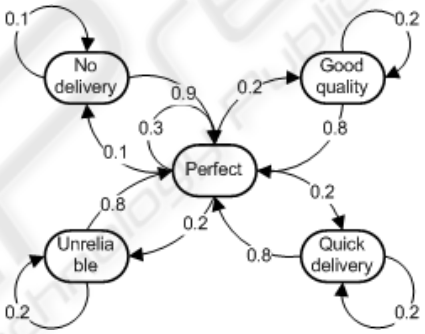

Figure 3: Reliable seller.

For the first simulation, a reliable *seller* (Fig. 3) behavior was chosen. The behavior has four states: perfect, good_quality, quick_delivery and unreliable. In the perfect state, items are delivered quickly and are of good quality, in the good_quality and quick_delivery states items are either delivered in good quality or quickly, and in the last state (unreliable) items are delivered late and are not of good quality. We chose the transitions so that the *seller* is usually in the perfect state (58%), followed by good_quality (34%), unreliable (5%) and quick_delivery (3%). In this setting, the item will be of good quality about 92% which can be considered to be a reliable seller.

The *buyer* behavior has only two states pay and not_pay, and the transitions are chosen in such a way that the main state is pay (in 90%). Since we evaluate the policies of a *buyer*, we could have chosen a perfect buyer that always pays, but the behavior of a buyer has influence on the outcome of an interaction, so limiting the behavior would also limit the amount of situations that arise. For example, if the *buyer* does not pay than it is normal to expect the *seller* not to

send the item, this is important for calculating trust because a *buyer* needs to know how to interpret the `item-timeout` event. If the *buyer* has payed and observes the `item-timeout` event than the seller is untrustwothy but if the buyer did not pay labeling the seller as untrustworty would be wrong.

The simulation was run with 500 *buyers* and 5 *sellers* using the behaviors described above, the *buyers* had a 100 interaction limit and had the threshold set to `good-quality-late-arrival`. Fig. 4 shows the average successful interaction rate with the average number of interactions per principal. While within the first interactions the rate is high, it begins to deteriorate as the average number of interactions increases. The main reason for the result is that the policies described above are too strict: at the first "bad" interaction with a seller (when the product is received in bad quality) the trust for that seller decreases instantly and becomes lower the threshold. The curve is stable within the first interactions because the buyers still have sellers not flagged as untrustworthy, and after marking a seller as untrustworthy they selected another one for which they have no information to interact with. By the end of the simulation 79% of buyers have interacted with all the sellers and 61% have not completed all the interactions because they ran out of trustworthy sellers.

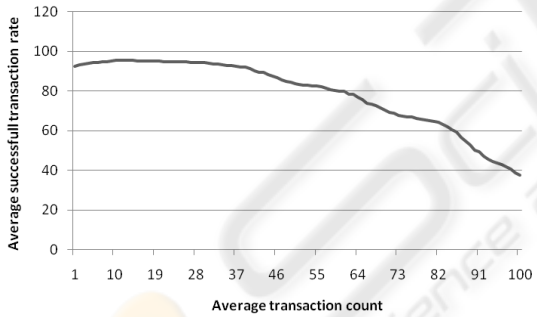

Figure 4: Successful interactions for the first run.

In order to improve the results, the justifications were modified to use quantifications, for example `qood_quality_late_arrival` would hold true if in 90% of the sessions in which the buyer payed the item has arrived and was of good quality but at least in 10% of the sessions the item was delivered late. Using the new justifications, the simulation was run using 500 buyers and 5 sellers with the same behaviours as before. The average successfull interaction rate is show in Fig. 5.

Using the more flexible policy, the average successful interaction rate remained over 90%, mainly because the number of sellers falsely flagged as untrustworthy was smaller.

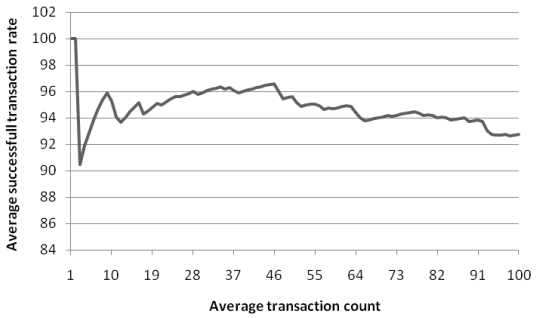

Figure 5: Successful interactions for the second run.

In the next run malicious sellers were added, they behave perfectly for 400 interactions and then for the next 400 they do not ship the products that were paid by the users. The goal is to build a good reputation and then scam buyers until the trust values buyers associate with them decreases. The setup has 10 reliable sellers, 3 malicious sellers and 100 buyers, the results are illustrated in Fig. 6. The average successful interaction rate decreases until all malicious sellers are identified and then remains constant.

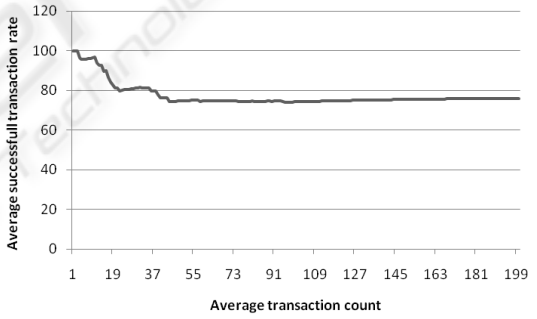

Figure 6: Successful interactions the third run.

For the last run, we introduce two roles for the sellers (which offer different products) and two roles for buyers (that require the new products). The setup contains 5 sellers that have both roles and 200 buyers split into two groups that have different local trust policies based on the flexible one presented above. All the sellers have the same behaviors, one role uses the reliable behavior described in the previous examples, and the other uses a perfect behavior (all products are shipped fast and in good quality). The first buyer group calculates trust separately for each role while the second group calculates trust by using the local history from both roles. The average interaction success rate is shown in Fig. 7, the dotted line shows the rate for the first group and the solid line for the second group.

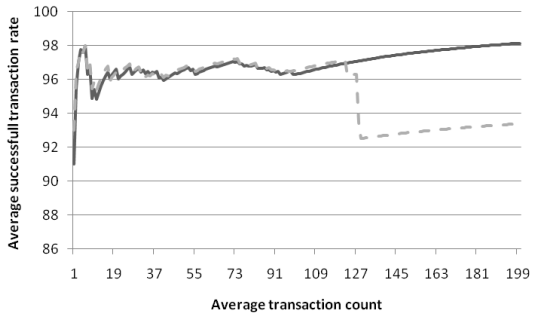

Figure 7: Successful interactions the fourth run.

The second group has very good results, maintaining a success rate over 96%, while the first one has a drop towards the end of the simulation. Because the buyers in the first group completely ignore the behaviour of a seller in other roles than the one that is required they calculate trust using limited data and as a result they have a higher rate of false positives which leads to the drop in success rate.

4 CONCLUSIONS

This paper presented a simulation framework for the evaluation of trust policies based on direct experience. The simulation is based on formal models for the representation of trust values and trust policies and can be used to compare trust policies within clear scenarios or to evaluate how one policy adapts to different scenarios. Several parameters are monitored: the local interaction histories of principals, their calculated trust values, and their behavior during the simulation; which can be used to analyze different properties of the policies. Future work will extend the framework to policies that take both direct experience and recommendations from others into consideration.

The notion of time from the extended SECURE model is supported by the simulation framework, and allows principals to interact asynchronously. Sessions can start at any time and there can be an unlimited number of sessions active for one principal at any moment. By changing simulation parameters, aspects like the *Interaction Frequency* or *Encounter Factor* of a CS can be evaluated. Future work will consider additional aspects of CSs like technical trustworthiness for the principals interactions.

While several principal behaviors can be represented by the model used in the simulation, the requirement that future states only depend on the current one may not hold true for complex malicious behaviors which might change their states based on analysis of other principals.

Many communication spaces use fully connected network topologies where any two principals can communicate directly, as such the simulation offers only this topology. However there are other topologies that may prove interesting like peer to peer. Also in order to simulate P2P networks events would most likely require parameters, e.g. `share(what)`, since having a event for each shareable resource would quickly become unmaintainable.

REFERENCES

Arbanowski, S., Ballon, P., David, K., Droegehorn, O., Eertink, H., Kellerer, W., van Kranenburg, H., Raatikainen, K., and Popescu-Zeletin, R. (2004). I-centric communications: personalization, ambient awareness, and adaptability for future mobile services. *IEEE Communications Magazine*, 42(9):63–69.

Axelrod, R. (1984). *The Evolution of Cooperation*. New York: Basic Books.

Eilers, F. and Nestmann, U. (2009). Deriving trust from experience. Submitted to the FAST International Workshop.

Fullam, K. K., Klos, T. B., Muller, G., Sabater, J., Schlosser, A., Topol, Z., Barber, K. S., Rosenschein, J. S., Vercouter, L., and Voss, M. (2005). A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 512–518, New York, NY, USA. ACM.

Kemeny, J. G. and Snell, J. L. (1983). *Finite Markov Chains*. Springer.

Krukow, K. (2006). *Towards a Theory of Trust for the Global Ubiquitous Computer*. PhD thesis, University of Aarhus, Denmark.

Ries S., Kangasharju J., M. M. (2006). A classification of trust systems. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 894–903.

Sabater, J. (2004). Toward a test-bed for trust and reputation models. In *7th International Workshop on Trust in Agent Societies*, pages 101–105.

Schlosser, A., Voss, M., and Brckner, L. (2005). On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, 9.

Sun, Y., Han, Z., Yu, W., and Liu, K. (2006). Attacks on trust evaluation in distributed networks. In *40th Annual Conference on Information Sciences and Systems (CISS)*, pages 1461–1466.

T, L. H., Hylands, C., Lee, E., Liu, J., Liu, X., Neuendorffer, S., Xiong, Y., Zhao, Y., and Zheng, H. (2003). Overview of the ptolemy project.