Rule-Based Trust Among Agents Using Defeasible Logic

Nick Bassiliades

Intelligent Systems group,
Software Engineering, Web and Intelligent Systems Lab,
Dept. Informatics, Aristotle University of Thessaloniki,
Greece

RuleML Webinar
Mar 31, 2017
TALK OVERVIEW

- Introduction on Trust / Reputation Models
  - Centralized approaches
  - Distributed approaches
  - Hybrid approaches

- Rule-based Trust / Reputation Models
  - HARM
  - DISARM

- Summary / Conclusions
TRUST AND REPUTATION

- Agents are supposed to act in open and risky environments (e.g. Web) with limited or no human intervention
- Making the appropriate **decision** about who to **trust** in order to **interact** with is necessary but challenging
- **Trust** and **reputation** are key elements in the design and implementation of multi-agent systems
- **Trust** is expectation or belief that a party will act benignly and cooperatively with the trusting party
- **Reputation** is the opinion of the public towards an agent, based on past experiences of interacting with the agent
- Reputation is used to **quantify** trust
**TRUST / REPUTATION MODELS**

- **Interaction trust**: agent’s own direct experience from past interactions (aka **reliability**)
  - Requires a long time to reach a satisfying estimation level (cold start)

- **Witness reputation**: reports of witnesses about an agent’s behavior, provided by other agents
  - Does not guarantee reliable estimation
    - Are self-interested agents willing to share information?
    - How much can you trust the informer?
IMPLEMENTING TRUST / REPUTATION MODELS

- **Centralized** approach:
  - One or more centralized trust authorities keep agent interaction references (ratings) and give trust estimations
    - Convenient for witness reputation models (e.g. eBay, SPORAS, etc.)
  + Simpler to implement; better and faster trust estimations
  - Less reliable; Unrealistic: hard to enforce central controlling authorities in open environments

- **Decentralized (distributed)** approach:
  - Each agent keeps its own interaction references with other agents and must estimate on its own the trust upon another agent
    - Convenient for interaction trust models
  + Robustness: no single point of failure; more realistic
  - Need more complex interaction protocols
OTHER TRUST / REPUTATION MODELS

- **Hybrid models**: Combination of Interaction Trust and Witness Reputation
  - Regret / Social Regret, FIRE, RRAF / TRR, CRM
  - T-REX / HARM / DISARM

- **Certified reputation**: third-party references provided by the agent itself
  - Distributed approach for witness reputation

**Centralized / Distributed**

Underlined -> rule-based
Agent A wants a service from agent B
Agent A asks agent C if agent B can be trusted
Agent C trusts agent B and replies yes to A
Agent A now trusts B and asks B to perform the service on A’s behalf

A = **truster** / beneficiary,
C = **trustor** / broker / consultant,
B = **trustee**
Agent A wants a service from agent B

Agent A judges if B is to be trusted from personal experience

Agent A trusts B and asks B to perform the service on A’s behalf

A = trustor / trusting / beneficiary, B = trustee
Agent A wants a service from agent B
Agent A asks agent B for proof of trust
Agent B provides some agents R that can guarantee that B can be trusted
Agent A now trusts B and asks B to perform the service on A’s behalf

A = trustor / truster / beneficiary,
B = trustee,
R = referee
RULE-BASED REPUTATION MODELS
RULE-BASED TRUST / REPUTATION MODELS

- **Centralized**
  - HARM
    - Hybrid
    - Knowledge-based
    - Temporal Defeasible Logic

- **Distributed**
  - DISARM
    - Hybrid
    - Knowledge-based, Defeasible Logic
    - Social relationships
HARM OVERVIEW

- **Centralized hybrid reputation model**
  - Combine Interaction Trust and Witness Reputation

- Rule-based approach
  - Temporal defeasible logic
  - Non-monotonic reasoning

- Ratings have a time offset
  - Indicates when ratings become active to be considered for trust assessment

- Intuitive method for assessing trust
  - Related to traditional human reasoning
(TEMPORAL) DEFEASIBLE LOGIC

- Temporal defeasible logic (TDL) is an extension of defeasible logic (DL).
- DL is a kind of non-monotonic reasoning
- Why defeasible logic?
  - Rule-based, deterministic (without disjunction)
  - Enhanced representational capabilities
  - Classical negation used in rule heads and bodies
  - Negation-as-failure can be emulated
  - Rules may support conflicting conclusions
  - Skeptical: conflicting rules do not fire
  - Priorities on rules resolve conflicts among rules
  - Low computational complexity
DEFEASIBLE LOGIC

- **Facts:** e.g. student(Sofia)
- **Strict Rules:** e.g. student(X) → person(X)
- **Defeasible Rules:** e.g.
  
  \[ r: \text{person}(X) \Rightarrow \text{works}(X) \]
  
  \[ r': \text{student}(X) \Rightarrow \neg \text{works}(X) \]
- **Priority Relation** between rules, e.g. \( r' > r \)
- **Proof theory example:**
  - A literal \( q \) is defeasibly provable if:
    - supported by a rule whose premises are all defeasibly provable AND
    - \( \neg q \) is not definitely provable AND
    - each attacking rule is non-applicable or defeated by a superior counter-attacking rule
Temporal Defeasible Logic

- **Temporal literals:**
  - Expiring temporal literals $l:t$
    - Literal $l$ is valid for $t$ time instances
  - Persistent temporal literals $l@t$
    - Literal $l$ is active after $t$ time instances have passed and is valid thereafter

- **Temporal rules:** $a_1:t_1 \ldots a_n:t_n \Rightarrow^d b:t_b$
  - $d$ is the delay between the cause and the effect

- **Example:**
  
  $(r1) \Rightarrow a@1$
  - Literal $a$ is created due to $r1$.

  $(r2) a@1 \Rightarrow^7 b:3$
  - It becomes active at time offset 1.
  - It causes the head of $r2$ to be fired at time 8.
  - The result $b$ lasts only until time 10.
  - Thereafter, only the fact $a$ remains.
HARM — AGENT EVALUATED ABILITIES

- Validity
  - An agent is valid if it is both sincere and credible
    - Sincere: believes what it says
    - Credible: what it believes is true in the world

- Completeness
  - An agent is complete if it is both cooperative and vigilant
    - Cooperative: says what it believes
    - Vigilant: believes what is true in the world

- Correctness
  - An agent is correct if its provided service is correct with respect to a specification

- Response time
  - Time that an agent needs to complete the transaction
HARM INTERACTION MODEL

- Central ratings repository: **Trustor**
  - A special agent responsible for collecting, storing, retrieving ratings and calculating trust values through defeasible reasoning
  - Considered certified/reliable

- Interacting agents
  - **Truster / Beneficiary**: an agent that wants to interact with another agent that offers a service
  - **Trustee**: the agent that offers the service

- **Role of Trustor**
  - *Before* the interaction, **Truster asks** from **Trustor** calculation of Trustees trust value
  - *After* the interaction, **Truster submits** rating for Trustee’s performance to **Trustor**
HARM RATING MECHANISM (I)

Truster

interact?

Trustee

Calculates reputation through defeasible reasoning, stored ratings and agent’s weights

Trustor

Gives personalized weights for each rating criteria

asks reputation

final reputation value
HARM RATING MECHANISM (II)

Trustor

interact

Trustee

Evaluation criteria
• Validity
• Completeness
• Correctness
• Response time

Weights
• Confidence
• Transaction value

Truster

evaluation report
HARM - RATINGS

- Agent A establishes interaction with agent B:
  - (A) **Truster** is the evaluating agent
  - (B) **Trustee** is the evaluated agent

- Truster’s rating value has 8 coefficients:
  - 2 IDs: **Truster, Trustee**
  - 4 abilities: Validity, Completeness, Correctness, Response time
  - 2 weights (how much attention agent should pay on each rating?):
    - **Confidence**: how confident the agent is for the rating
      - Ratings of confident trusters are more likely to be right
    - **Transaction value**: how important the transaction was for the agent
      - Trusters are more likely to report truthful ratings on important transactions

- Example (defeasible RuleML / d-POSL syntax):

\[
\text{rating(id→1, truter→A, trustee→B, validity→5, completeness→6, correctness→6, resp_time→8, confidence→0.8, transaction_val→0.9).}
\]
HARM — EXPERIENCE TYPES

- Direct Experience ($PR_{AX}$)
- Indirect Experience
  - reports provided by strangers ($SR_{AX}$)
  - reports provided by known agents (e.g. friends) due to previous interactions ($KR_{AX}$)
- Final reputation value
  - of an agent $X$, required by an agent $A$

$$R_{AX} = \{PR_{AX}, KR_{AX}, SR_{AX}\}$$
- One or more rating categories may be missing
  - E.g. a newcomer has no personal experience

- A user is much more likely to believe statements from a trusted acquaintance than from a stranger.
  - Personal opinion (AX) is more valuable than strangers’ opinion (SX) and known partners (KX).

- Superiority relationships among rating categories

```
AX, KX, SX

AX, KX

AX, SX

KX, SX

AX

KX

SX
```
HARM — Final Reputation Value

- $R_{AX}$ is a function that combines each available category
  - personal opinion (AX)
  - strangers’ opinion (SX)
  - previously trusted partners (KX)

- HARM allows agents to define weights of ratings’ coefficients
  - Personal preferences

$$R_{AX} = \mathcal{S}(PR_{AX}, KR_{AX}, SR_{AX})$$

$$R_{AX} = \mathcal{S} \left[ \frac{AVG \left( w_i \times \log \left( pr_{AX}^{\text{coefficient}} \right) \right)}{\sum_{i=1}^{4} w_i}, \frac{AVG \left( w_i \times \log \left( kr_{AX}^{\text{coefficient}} \right) \right)}{\sum_{i=1}^{4} w_i}, \frac{AVG \left( w_i \times \log \left( sr_{AX}^{\text{coefficient}} \right) \right)}{\sum_{i=1}^{4} w_i} \right]$$

coefficient = \{validity, completeness, correctness, response \_ time\}
HARM- WHICH RATINGS “COUNT”?

\( r_1: \text{count} \text{ rating}(\text{rating}\rightarrow \text{idx}, \text{truster}\rightarrow \text{a}, \text{trustee}\rightarrow \text{x}) := \)

\( \text{confidence} \text{ threshold}(\text{conf}), \text{transaction} \text{ value} \text{ threshold}(\text{tran}), \)

\( \text{rating}(\text{id}\rightarrow \text{idx}, \text{confidence}\rightarrow \text{confx}, \text{transaction} \text{ val}\rightarrow \text{tranx}), \)

\( \text{?confx} \geq \text{?conf}, \text{?tranx} \geq \text{?tran}. \)

\( r_2: \text{count} \text{ rating}(\ldots) := \)

\( \ldots \)

\( \text{?confx} \geq \text{?conf}. \)

\( r_3: \text{count} \text{ rating}(\ldots) := \)

\( \ldots \)

\( \text{?tranx} \geq \text{?tran}. \)

\( r_1 > r_2 > r_3 \)

- if both confidence and transaction importance are high, then rating will be used for estimation
- if transaction value is lower than the threshold, but confidence is high, then use rating
- if there are only ratings with high transaction value, then they should be used
- In any other case, omit the rating
HARM - CONFLICTING LITERALS

- All the previous rules conclude positive literals.

- These literals are conflicting each other, for the same pair of agents (truster and trustee)
  - We want in the presence e.g. of personal experience to omit strangers’ ratings.
  - That’s why there is also a superiority relationship between the rules.

- The conflict set is formally determined as follows:

\[
C[\text{count\_rating(truster}\rightarrow?a, \text{trustee}\rightarrow?x)] = \\
\{ \neg \text{count\_rating(truster}\rightarrow?a, \text{trustee}\rightarrow?x) \} \cup \\
\{ \text{count\_rating(truster}\rightarrow?a_1, \text{trustee}\rightarrow?x_1) \mid ?a \neq ?a_1 \land ?x \neq ?x_1 \}
\]
known(agent1→?a, agent2→?y) :-

count_pr(agent→?a, truster→?a, trustee→?x, rating→?id) :-
    count_rating(rating→?id, truster→?a, trustee→?x).

count_kr(agent→?a, truster→?k, trustee→?x, rating→?id) :-
    known(agent1→?a, agent2→?k),
    count_rating(rating→?id, truster→?k, trustee→?x).

count_sr(agent→?a, truster→?s, trustee→?x, rating→?id) :-
    count_rating(rating→?id, truster→?s, trustee→?x),
    not(known(agent1→?a, agent2→?s)).
Final step is to decide whose experience will “count”: direct, indirect (witness), or both.

The decision for $R_{AX}$ is based on a relationship theory

- e.g. **Theory #1**: All categories count equally.

\[
\begin{align*}
\text{r}_8 : \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?id\_rating_{AX}) & := \\
& \text{count}_{\text{pr}}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?id\_rating_{AX}). \\
\text{r}_9 : \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?id\_rating_{KX}) & := \\
& \text{count}_{\text{kr}}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?id\_rating_{KX}). \\
\text{r}_{10} : \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?id\_rating_{SX}) & := \\
& \text{count}_{\text{sr}}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?id\_rating_{SX}).
\end{align*}
\]
SELECTING EXPERIENCES
ALL CATEGORIES COUNT EQUALLY

AX, KX, SX

AX, KX

AX, SX

KX, SX

AX

KX

SX
SELECTING EXPERIENCES - THEORY #2
PERSONAL EXPERIENCE IS PREFERRED TO FRIENDS’ OPINION TO STRANGERS’ OPINION

\[ r_8 : \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id_rating}_{\text{AX}}) := \text{count_pr}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id_rating}_{\text{AX}}). \]

\[ r_9 : \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id_rating}_{\text{KX}}) := \text{count_kr}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id_rating}_{\text{KX}}). \]

\[ r_{10} : \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id_rating}_{\text{SX}}) := \text{count_sr}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id_rating}_{\text{SX}}). \]

\[ r_8 > r_9 > r_{10} \]
SELECTING EXPERIENCES

PERSONAL EXPERIENCE IS PREFERRED TO FRIENDS’ OPINION TO STRANGERS’ OPINION

\[ AX, KX, SX \]

\[ AX, KX \]
\[ AX, SX \]
\[ KX, SX \]

\[ AX \]
\[ KX \]
\[ SX \]
SELECTING EXPERIENCES - THEORY #3
PERSONAL EXPERIENCE AND FRIENDS' OPINION IS PREFERRED TO STRANGERS' OPINION

\[ r_8: \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id\_rating}_{AX}) := \text{count}_{pr}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id\_rating}_{AX}). \]

\[ r_9: \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id\_rating}_{KX}) := \text{count}_{kr}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id\_rating}_{KX}). \]

\[ r_{10}: \text{participate}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id\_rating}_{SX}) := \text{count}_{sr}(\text{agent} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{rating} \rightarrow ?\text{id\_rating}_{SX}). \]

\[ r_8 > r_{10}, r_9 > r_{10} \]
SELECTING EXPERIENCES
PERSONAL EXPERIENCE AND FRIENDS’ OPINION IS PREFERRED TO STRANGERS’ OPINION

AX, KX, SX

AX, KX

AX, SX

KX, SX

AX

KX

SX
Agents may change their behavior / objectives at any time
  - Evolution of trust over time should be taken into account
  - Only the latest ratings participate in the reputation estimation

In the temporal extension of HARM:
  - each rating is a persistent temporal literal of TDL
  - each rule conclusion is an expiring temporal literal of TDL

Truster’s rating is active after \texttt{time\_offset} time instances have passed and is valid thereafter

\[
\text{rating}(\text{id} \rightarrow \text{val}_1, \text{truster} \rightarrow \text{val}_2, \text{trustee} \rightarrow \text{val}_3, \text{validity} \rightarrow \text{val}_4, \\
\text{completeness} \rightarrow \text{val}_5, \text{correctness} \rightarrow \text{val}_6, \text{resp\_time} \rightarrow \text{val}_7, \\
\text{confidence} \rightarrow \text{val}_8, \text{transaction\_val} \rightarrow \text{value}_9) @ \text{time\_offset}.
\]
HARM - TEMPORAL DEFEASIBLE LOGIC EXTENSION

- Rules are modified accordingly:
  - each rating is active after \( t \) time instances have passed
  - each conclusion has a \textit{duration} that it holds
  - each rule has a \textit{delay} between the cause and the effect

\[
\text{count_rating}(\text{rating} \rightarrow ?\text{idx}, \text{truster} \rightarrow ?a, \text{trustee} \rightarrow ?x) : \text{duration} := \text{delay} \\
\text{confidence_threshold}(?\text{conf}), \\
\text{transaction_value_threshold}(?\text{tran}), \\
\text{rating}(\text{id} \rightarrow ?\text{idx}, \text{confidence} \rightarrow ?\text{conf}_x, \text{transaction_value} \rightarrow ?\text{tran}_x) @ t, \\
?\text{conf}_x \geq ?\text{conf}, ?\text{tran}_x \geq ?\text{tran}.
\]
**DISARM OVERVIEW**

- **Distributed** extension of HARM

- **Distributed hybrid** reputation model
  - Combines Interaction Trust and Witness Reputation
  - Ratings are located through agent’s social relationships

- Rule-based approach
  - Defeasible logic
  - Non-monotonic reasoning

- Time is directly used in:
  - Decision making rules about recency of ratings
  - Calculation of reputation estimation (similar to T-REX)

- Intuitive method for assessing trust
  - Related to traditional human reasoning
Social relationships of trust among agents
- If an agent is satisfied with a partner it is more likely to interact again in the future
- If dissatisfied it will not interact again

Each agent maintains 2 relationship lists:
- **White-list**: Trusted agents
- **Black-list**: Non-trusted agents
- All other agents are indifferent (**neutral zone**)

Each agent decides which agents are added / removed from each list, using rules

Personal **social network**
Truster’s rating value has 11 coefficients:
- 2 IDs: Truster, Trustee
- 4 abilities: Validity, Completeness, Correctness, Response time
- 2 weights: Confidence, Transaction value

- Timestamp
- Cooperation: willingness to do what is asked for
  - Important in distributed social environments
- Outcome feeling: (dis)satisfaction for the transaction outcome
  - Degree of request fulfillment

Example (defeasible RuleML / d-POSL syntax):

```
rating (id→1, truster→A, trustee→X, t→140630105632, resp_time→9, validity→7, completeness→6, correctness→6, cooperation→8, outcome_feeling→7, confidence→0.9, transaction_val→0.8)
```
Propagating request

1. Request ratings

3. Receive ratings

5. Choose agent x

6. Receive service

4. Evaluate Reputation
   (DISARM rules + ratings)

7. Rate agent X

Agent A
Truster

Agent X
Trustee

WL agents providing ratings
good_behavior(time → ?t, truster→ ?a, trustee→ ?x, reason → all) :-
    resp_time_thrshld(?resp), valid_thrshld(?val), ..., trans_val_thrshld(?trval),
    rating(id→?idx, time → ?t, truster→ ?a, trustee→ ?x, resp_time→?resp_x,
        validity→?val_x, transaction_val→?trval_x, completeness→?com_x,
        correctness→?cor_x, cooperation→?coop_x, outcome_feeling→?outf_x),
bad_behavior(time → ?t, truster→ ?a, trustee→ ?x, reason → response_time) :-
    rating(id→?idx, time → ?t, truster→ ?a, trustee→ ?x, resp_time→?resp_x),
    resp_time_thrshld(?resp), ?resp_x>?resp.

- Any combination of parameters can be used with any defeasible theory.
DISARM - DECIDING WHO TO TRUST

- Has been good twice for the same reason

\[
\text{add}\_\text{whitelist}(\text{trustee} \rightarrow ?x, \text{time} \rightarrow ?t2) := \\
\text{good}\_\text{behavior}(\text{time} \rightarrow ?t1, \text{truster} \rightarrow ?\text{self}, \text{trustee} \rightarrow ?x, \text{reason} \rightarrow ?r), \\
\text{good}\_\text{behavior}(\text{time} \rightarrow ?t2, \text{truster} \rightarrow ?\text{self}, \text{trustee} \rightarrow ?x, \text{reason} \rightarrow ?r), \\
?t2 > ?t1.
\]

- Has been bad thrice for the same reason

\[
\text{add}\_\text{blacklist}(\text{trustee} \rightarrow ?x, \text{time} \rightarrow ?t3) := \\
\text{bad}\_\text{behavior}(\text{time} \rightarrow ?t1, \text{truster} \rightarrow ?\text{self}, \text{trustee} \rightarrow ?x, \text{reason} \rightarrow ?r), \\
\text{bad}\_\text{behavior}(\text{time} \rightarrow ?t2, \text{truster} \rightarrow ?\text{self}, \text{trustee} \rightarrow ?x, \text{reason} \rightarrow ?r), \\
\text{bad}\_\text{behavior}(\text{time} \rightarrow ?t3, \text{truster} \rightarrow ?\text{self}, \text{trustee} \rightarrow ?x, \text{reason} \rightarrow ?r), \\
\]
blacklist(trustee→ ?x, time → ?t) :=
  ¬whitelist(trustee→ ?x, time → ?t1),

¬blacklist(trustee→ ?x, time → ?t2) :=
  blacklist(trustee→ ?x, time → ?t1),
  add_whitelist(trustee→ ?x, time → ?t2),
  ?t2 > ?t1.

whitelist(trustee→ ?x, time → ?t) :=
  ¬blacklist(trustee→ ?x, time → ?t1),

Add to the blacklist

Remove from the blacklist

Add to the whitelist

...
**DISARM - LOCATING RATINGS**

- Ask for ratings about an agent sending request messages
- To whom and how?
  - To everybody
  - To direct “neighbors” of the agent’s “social network”
  - To indirect “neighbors” of the “social network” though message propagation for a predefined number of hops (Time-to-Live - P2P)
- “Neighbors” are the agents in the whitelist
- Original request:

  send_message(sender→?self, receiver→?r, 
  
  msg →request_reputation/about→?x,ttl→?t) := 
  
  ttl_limit(?t), whitelist(?r), locate_ratings/about→?x).
Upon receiving request, return rating to the sender

```
send_message(sender→?self, receiver→?s,
msg →rating(id→ id_x, truster→ ?self, trustee→ ?x, ...)) :=

receive_message(sender→?s, receiver→?self,
msg →request_rating(about→?x)),

rating(id→?id_x, truster→ ?self, trustee→ ?x, ...).
```

If time-to-live has not expired propagate request to all friends

```
send_message(sender→?s, receiver→?r,
msg →request_reputation(about→?x, ttl→?t)):=

receive_message(sender→?s, receiver→?self,
msg →request_rating(about→?x, ttl→?t)),

?t >0, WL(?r), ?t1 is ?t - 1.
```
DISARM - RATING CATEGORIES

- Direct Experience ($PR_X$)
- Indirect Experience (reports provided by other agents):
  - “Friends” ($WR_X$) – agents in the whitelist
  - Known agents from previous interactions ($KR_X$)
  - Complete strangers ($SR_X$)

- Final reputation value
  - $R_X = \{PR_X, WR_X, KR_X, SR_X\}$

New compared to HARM
DISARM - SELECTING RATINGS

- According to user’s preferences

$$\text{eligible\_rating}(\text{rating} \rightarrow ?id_x, \text{truter} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{reason} \rightarrow \text{cnf\_imp}) :=$$
$$\text{conf\_thrshld}(?\text{conf}), \text{trans\_val\_thrshld}(?\text{tr}),$$
$$\text{rating}(\text{id} \rightarrow ?id_x, \text{truter} \rightarrow ?a, \text{trustee} \rightarrow ?x, \text{conf} \rightarrow ?\text{conf}_x, \text{trans\_val} \rightarrow ?\text{tr}_x),$$
$$?\text{conf}_x \geq ?\text{conf}, ?\text{tr}_x \geq ?\text{tr}.$$  

- According to temporal restrictions

$$\text{count\_rating}(\text{rating} \rightarrow ?id_x, \text{truter} \rightarrow ?a, \text{trustee} \rightarrow ?x) :=$$
$$\text{time\_from\_thrshld}(?\text{ftime}), \text{time\_to\_thrshld}(?\text{ttime}),$$
$$\text{rating}(\text{id} \rightarrow ?id_x, t \rightarrow ?t_x, \text{truter} \rightarrow ?a, \text{trustee} \rightarrow ?x),$$
$$?\text{ftime} \leq ?t_x \leq ?\text{ttime}.$$
count_wr (rating $\rightarrow$?id$_x$, trustee$\rightarrow$?x) :-
  eligible_rating(rating $\rightarrow$ ?id$_x$, cat$\rightarrow$?c, truster$\rightarrow$?k, trustee$\rightarrow$ ?x),
  count_rating(rating$\rightarrow$?id$_x$, truster$\rightarrow$?k, trustee$\rightarrow$ ?x),
  known(agent$\rightarrow$?k),
  whitelist (trustee $\rightarrow$?k).

count_kr (rating $\rightarrow$?id$_x$, trustee$\rightarrow$?x) :-
  eligible_rating(rating$\rightarrow$?id$_x$, cat$\rightarrow$?c, truster$\rightarrow$?k, trustee$\rightarrow$ ?x),
  count_rating(rating$\rightarrow$?id$_x$, truster$\rightarrow$?k, trustee$\rightarrow$ ?x),
  known(agent$\rightarrow$?k),
  not(whitelist(trustee $\rightarrow$?k)),
  not(blacklist (trustee $\rightarrow$?k)).
When ratings provided by an agent are outside the standard deviation of all received ratings, the agent might behave dishonestly:

\[
\text{bad_assessment} \left(\text{time} \rightarrow ?t, \text{truster} \rightarrow ?y, \text{trustee} \rightarrow ?x\right) :\]

\[
\text{standard_deviation_value}(?t, ?y, ?x, \text{stdev}_y),
\]

\[
\text{standard_deviation_value}(?t, _, ?x, \text{stdev}),
\]

\[
\text{stdev}_y > \text{stdev}.
\]

When two bad assessments for the same agent were given in a certain time window, trust is lost:

\[
\text{remove_whitelist}(\text{agent} \rightarrow ?y, \text{time} \rightarrow ?t2) :\]

\[
\text{whitelist}(\text{truster} \rightarrow ?y),
\]

\[
\text{time_window}(?\text{wtime}),
\]

\[
\text{bad_assessment}(\text{time} \rightarrow ?t1, \text{truster} \rightarrow ?y, \text{trustee} \rightarrow ?x),
\]

\[
\text{bad_assessment}(\text{time} \rightarrow ?t2, \text{truster} \rightarrow ?y, \text{trustee} \rightarrow ?x),
\]

\[
?t2 \leq ?t1 + ?\text{wtime}.
\]
CONCLUDING...
TRUST / REPUTATION MODELS FOR MULTIAGENT SYSTEMS

- **Interaction Trust** (personal experience) vs. **Witness Reputation** (Experience of others)
  - Hybrid models

- **Centralized** (easy to locate ratings) vs. **Distributed** (more robust)

- Rule-based trust / reputation models
  - **HARM** *(centralized, hybrid, knowledge-based, temporal defeasible logic)*
  - **DISARM** *(distributed, hybrid, knowledge-based, defeasible logic, time decay, social relationships, manages dishonesty)*
CONCLUSIONS

- **Centralized models**
  - Achieve higher performance because they have access to more information
  - Simple interaction protocols, easy to locate ratings
  - Both interaction trust and witness reputation can be easily implemented
    - Single-point-of-failure
    - Cannot scale well (bottleneck, storage & computational complexity)
    - Central authority hard to enforce in open multiagent systems

- **Distributed models**
  - Less accurate trust predictions, due to limited information
  - Complex interaction protocols, difficult to locate ratings
  - More appropriate for interaction trust
  + Robust – no single-point-of-failure
  + Can scale well (no bottlenecks, less complexity)
  + More realistic in open multiagent systems
CONCLUSIONS

- **Interaction trust**
  + More trustful
  - Requires a long time to reach a satisfying estimation level

- **Witness reputation**
  - Does not guarantee reliable estimation
  + Estimation is available from the beginning of entering a community

- **Hybrid models**
  + Combine interaction trust and witness reputation
  - Combined trust metrics are usually only based on arbitrary / experimentally-optimized weights
CONCLUSIONS — PRESENTED MODELS

- **Centralized models**
  - Cannot scale well (bottleneck, storage & computational complexity)
  + **HARM** reduces computational complexity by reducing considered ratings, through rating selection based on user’s domain-specific knowledge

- **Distributed models**
  - Less accurate trust predictions, due to limited information
  - Complex interaction protocols, difficult to locate ratings
  + **DISARM** finds ratings through agent social relationships and increases accuracy by using only known-to-be-trustful agents

- **Hybrid models**
  - Combined trust metrics are usually only based on arbitrary weights
  + **HARM & DISARM** employ a knowledge-based highly-customizable (both to user prefs & time) approach, using non-monotonic defeasible reasoning
ACKNOWLEDGMENTS

- The work described in this talk has been performed in cooperation with Dr. Kalliopi Kravari
  - Former PhD student, currently postdoctorate affiliate

- Other contributors:
  - Dr. Efstratios Kontopoulos (former PhD student, co-author)
  - Dr. Antonios Bikakis (Lecturer, University College London, PhD examiner)
RELEVANT PUBLICATIONS


Thank you!

Any questions?
A FEW WORDS ABOUT US...

- Aristotle University of Thessaloniki, Greece
  - Largest University in Greece and South-East Europe
  - Since 1925, 41 Departments, ~2K faculty, ~80K students

- Dept. of Informatics
  - Since 1992, 28 faculty, 5 research labs, ~1100 undergraduate students, ~200 MSc students, ~80 PhD students, ~120 PhD graduates, >3500 pubs

- Software Engineering, Web and Intelligent Systems Lab
  - 7 faculty, 20 PhD students, 9 Post-doctorate affiliates

- Intelligent Systems group (http://intelligence.csd.auth.gr)
  - 4 faculty, 7 PhD students, 17 PhD graduates
  - Research on Artificial Intelligence, Machine Learning / Data Mining, Knowledge Representation & Reasoning / Semantic Web, Planning, Multi-Agent Systems
  - 430 publications, 35 projects
EVALUATION OF TRUST / REPUTATION MODELS
Simulation in the EMERALD* multi-agent system

Service provider agents
- All provide the same service

Service consumer agents
- Choose provider with the higher reputation value

Performance metric: Utility Gain

---

Number of simulations: 500
Number of providers: 100

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Good providers</td>
<td>10</td>
</tr>
<tr>
<td>Ordinary providers</td>
<td>40</td>
</tr>
<tr>
<td>Intermittent providers</td>
<td>5</td>
</tr>
<tr>
<td>Bad providers</td>
<td>45</td>
</tr>
</tbody>
</table>

DISARM VS. HARM VS. STATE-OF-THE-ART

Mean Utility Gain

Mean Utility Gain (Mean UG) over time for different systems:

- DISARM
- CRM
- NONE
- FIRE
- HARM
- Certified Reputation
- Social Regret

RuleML Webinar, Mar 31, 2017
Better performance when alone, due to more social relationships
DISARM VS. HARM VS. STATE-OF-THE-ART

Storage Space

Memory Space (%) vs. Time

-N. Bassiliades - Rule-Based Trust Among Agents Using Defeasible Logic

RuleML Webinar, Mar 31, 2017
EVALUATING DISHONESTY HANDLING

![Graph showing Mean UG vs Dishonest density, comparing DISARM (without standard deviation) and DISARM (standard deviation included).]