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

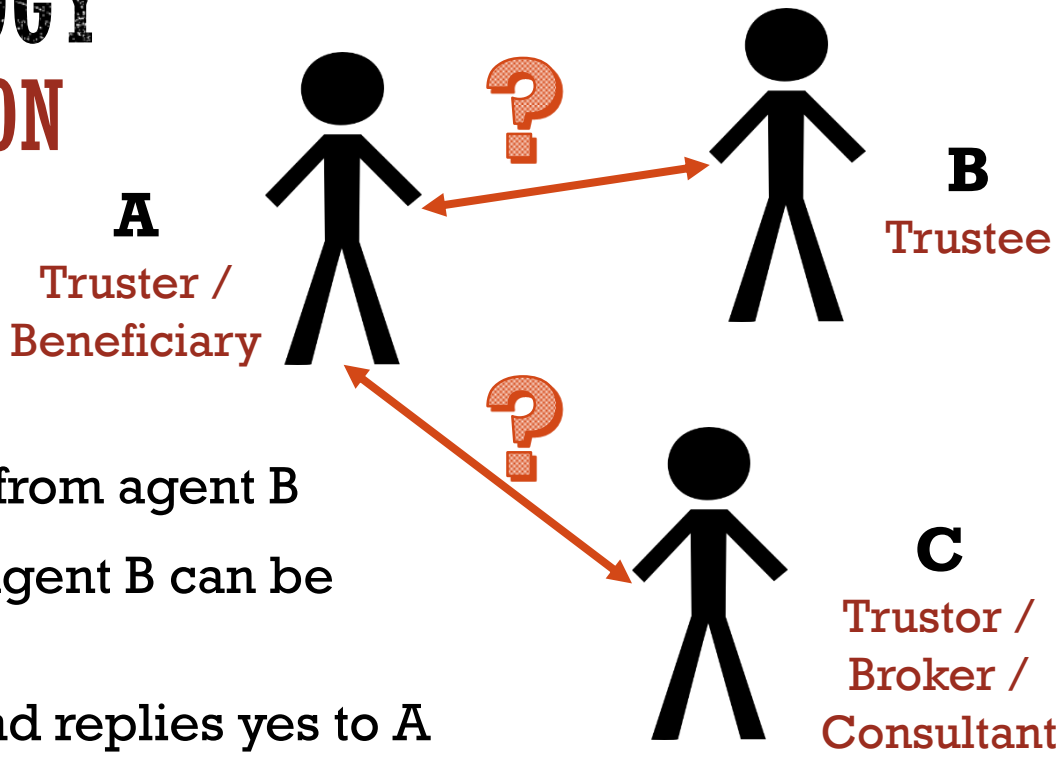
OTHER TRUST / REPUTATION MODELS

- 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

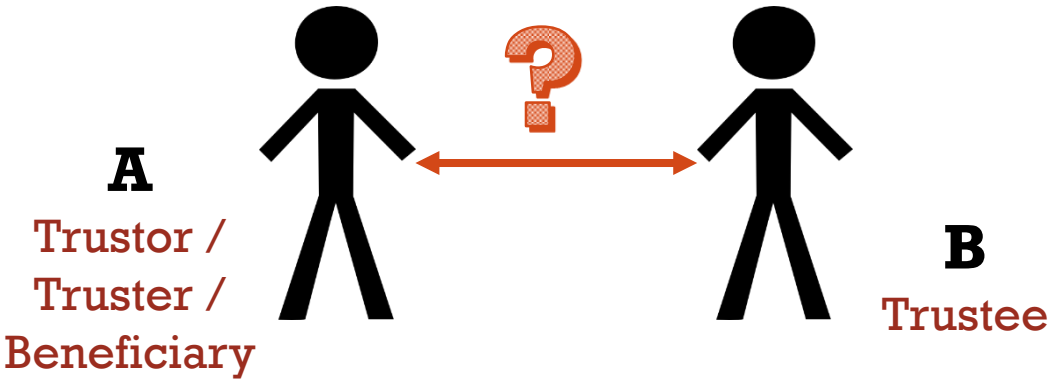
SOME TERMINOLOGY
WITNESS REPUTATION



- 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

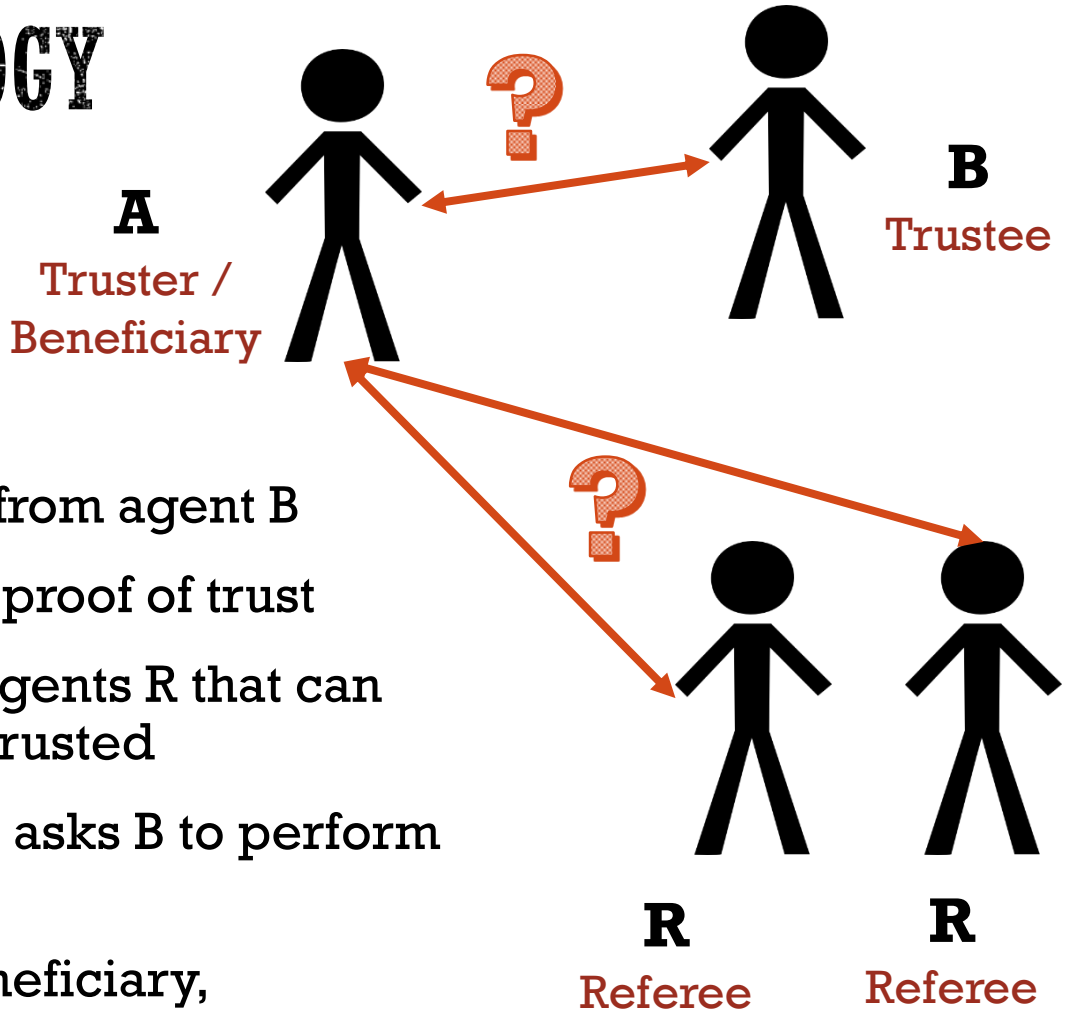
SOME TERMINOLOGY

INTERACTION TRUST



- 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 / truster / beneficiary,
  B = trustee

SOME TERMINOLOGY
REFERENCES



- 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

Kravari, K., & Bassiliades, N. (2012). HARM: A Hybrid Rule-based Agent Reputation Model Based on Temporal Defeasible Logic. 6th International Symposium on Rules: Research Based and Industry Focused (RuleML-2012). Springer, LNCS 7438: 193-207.

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) $\rightarrow$ person(X)
- Defeasible Rules: e.g. r: person(X) $\Rightarrow$ works(X)

  r': student(X) $\Rightarrow \neg$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) => a@1

(r2) a@1=>^{7} b:3

Literal a is created due to r1.

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.

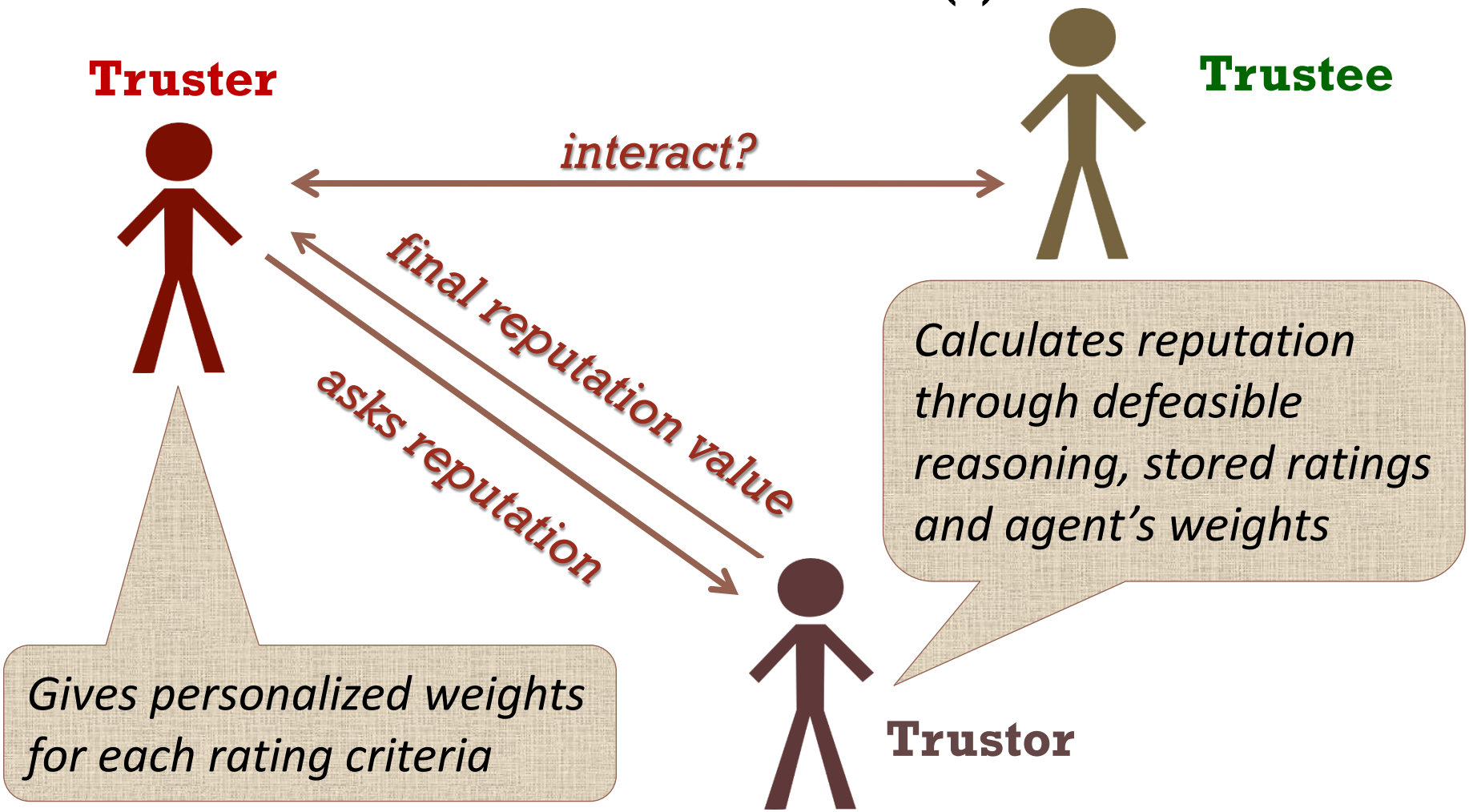
HARM – AGENT EVALUATED ABILITIES

- 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

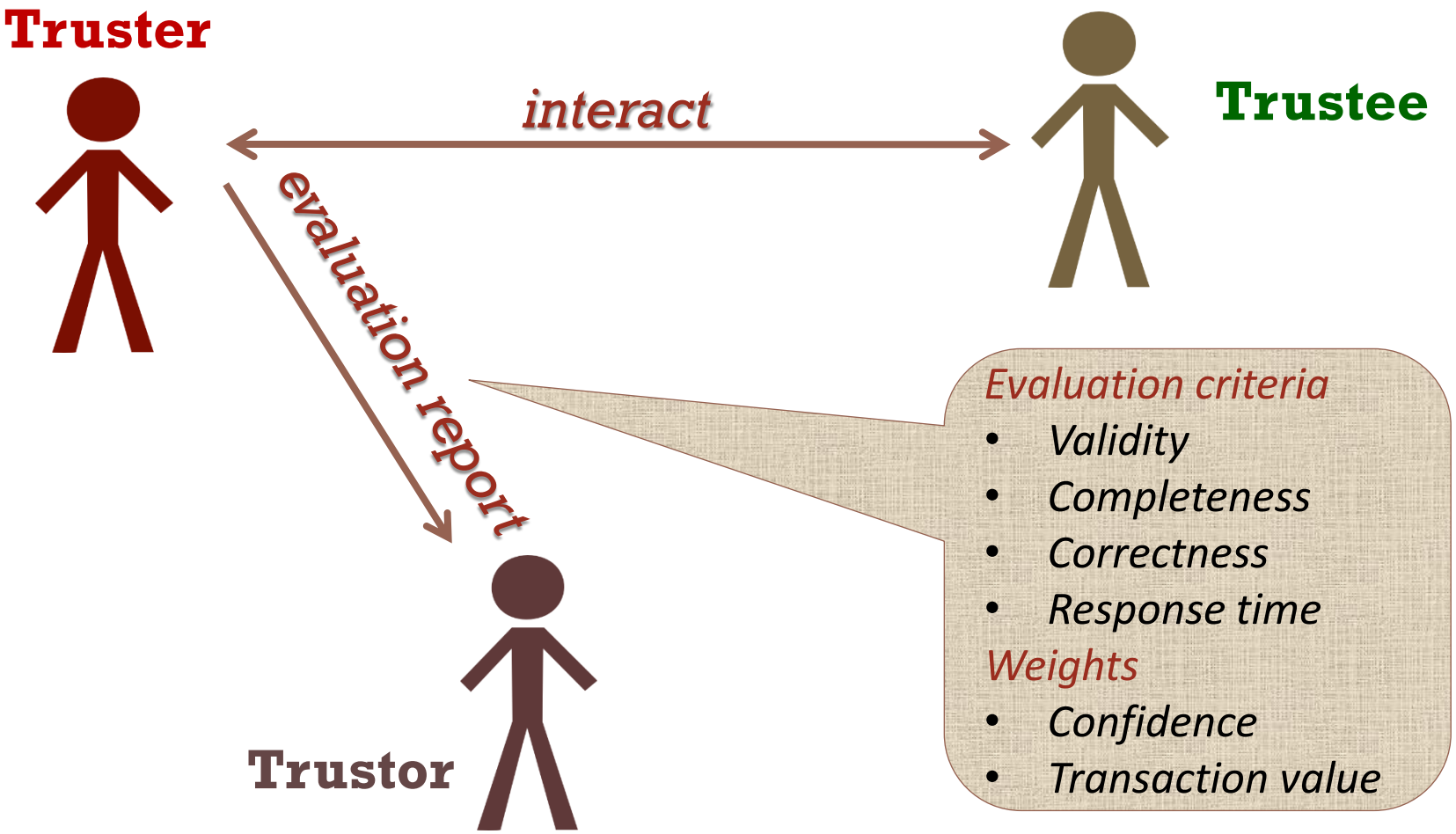
HARM INTERACTION MODEL

- 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)



HARM RATING MECHANISM (II)



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):

```
rating(id→1,truster→A,trustee→B,validity→5,completeness→6,
       correctness→6,resp_time→8,confidence→0.8,transaction_val→0.9).
```

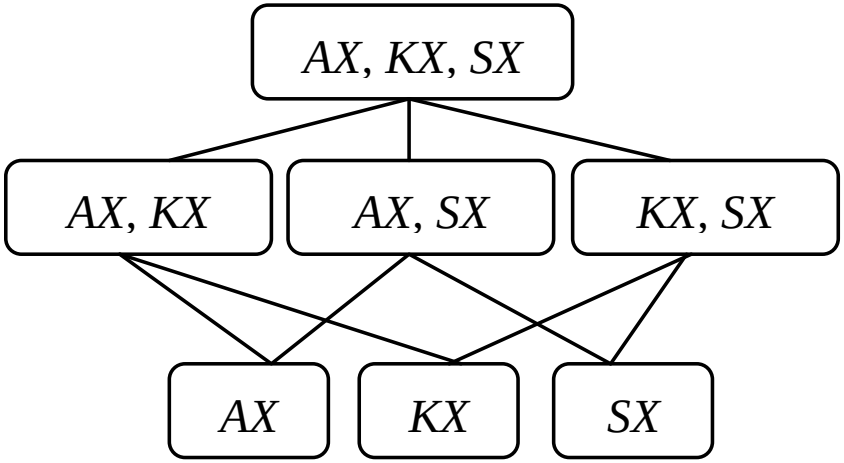
HARM – EXPERIENCE TYPES

- 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

$$R_{AX} = \{PR_{AX}, KR_{AX}, SR_{AX}\}$$

HARM – EXPERIENCE TYPES

- 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



HARM – FINAL REPUTATION VALUE

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

$$R_{AX} = \Im\left(PR_{AX}, KR_{AX}, SR_{AX} \right)$$

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

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

$$coefficient = \{validity, completeness, correctness, response\_time\}$$

HARM- WHICH RATINGS "COUNT"?

r_1: count_rating(rating→?idx, truster→?a, trustee→ ?x) :=

confidence_threshold(?conf), transaction_value_threshold(?tran),

rating(id→?idx, confidence→?confx, transaction_val→?tranx),

?confx >= ?conf, ?tranx >= ?tran.

r_2: count_rating(…) :=

…

?confx >= ?conf.

r_3: count_rating(…) :=

…

?tranx >= ?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:

$$C[\text{count\_rating}(\text{truster}\rightarrow ?a, \text{trustee}\rightarrow ?x)] =$$

$$\{ \neg\ \text{count\_rating}(\text{truster}\rightarrow ?a, \text{trustee}\rightarrow ?x) \} \cup$$

$$\{ \text{count\_rating}(\text{truster}\rightarrow ?a1, \text{trustee}\rightarrow ?x1) \mid ?a \neq ?a1 \wedge ?x \neq ?x1 \}$$

HARM - DETERMINING EXPERIENCE TYPES

known(agent1→?a, agent2→?y) :-

count_rating(rating → ?id, truster→?a, trustee→?y).

Which agents are considered as known?

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($agent_1$→?a, $agent_2$→?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($agent_1$→?a, $agent_2$→?s)).

Rating categories

HARM – SELECTING EXPERIENCES

- 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.

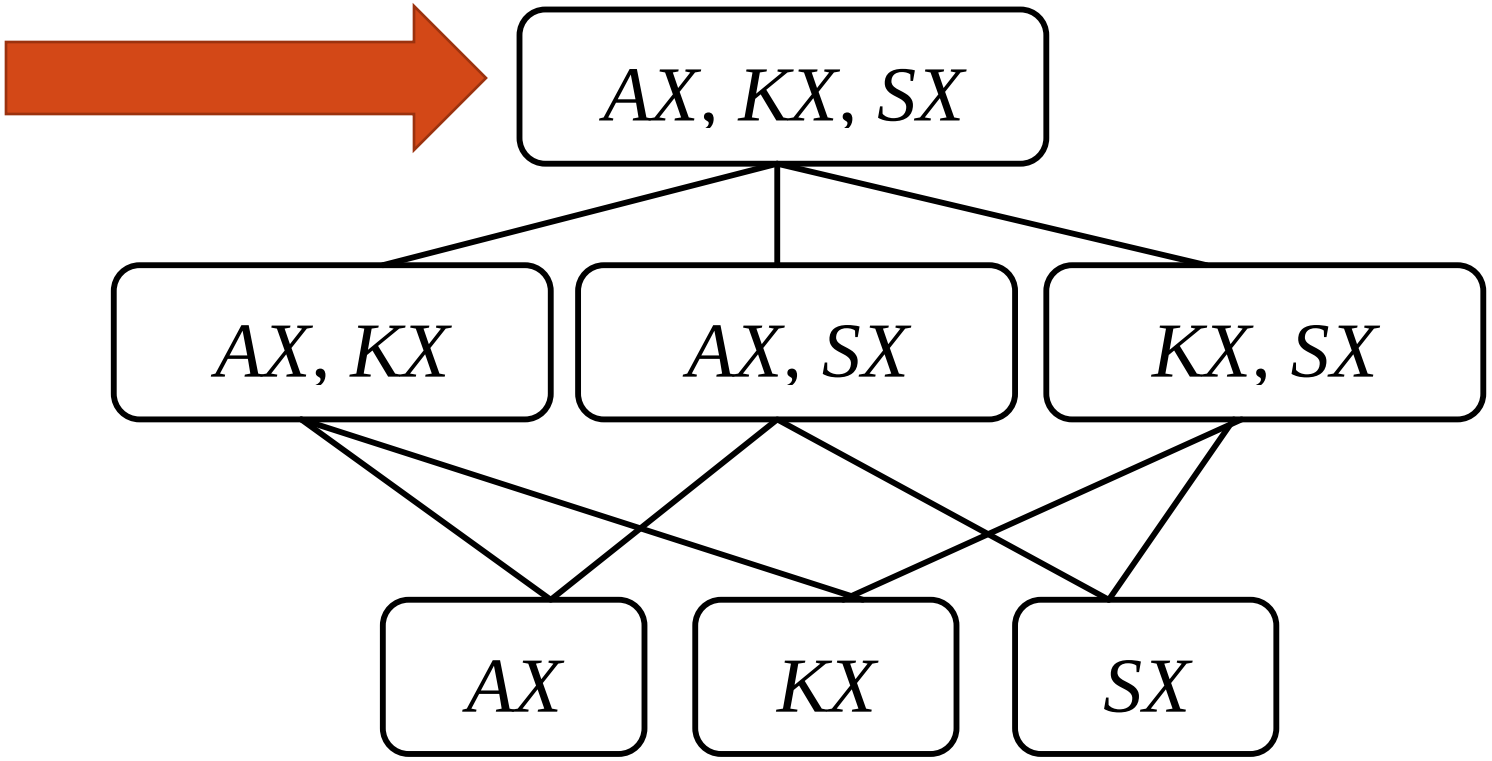
r_8: participate(agent→?a, trustee→?x, rating→?id_rating$_{AX}$) :=
count_pr(agent→?a, trustee→?x, rating→ ?id_rating$_{AX}$).

r_9: participate(agent→?a, trustee→?x, rating→?id_rating$_{KX}$) :=
count_kr(agent→?a, trustee→?x, rating→ ?id_rating$_{KX}$).

r_{10}: participate(agent→?a, trustee→?x, rating→?id_rating$_{SX}$) :=
count_sr(agent→?a, trustee→?x, rating→ ?id_rating$_{SX}$).

SELECTING EXPERIENCES

ALL CATEGORIES COUNT EQUALLY



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

r_8: participate(agent→?a, trustee→?x, rating→?id_rating$_{AX}$) :=
count_pr(agent→?a, trustee→?x, rating→ ?id_rating$_{AX}$).

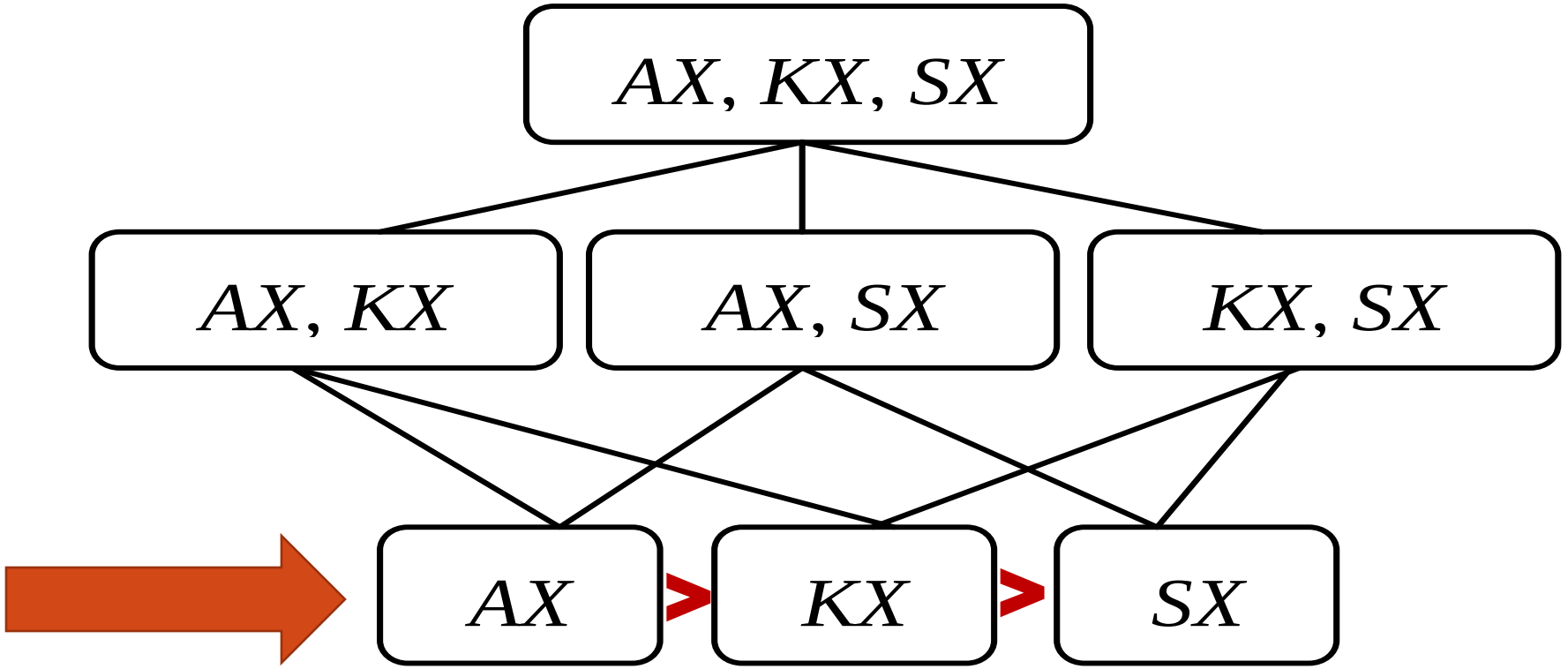
r_9: participate(agent→?a, trustee→?x, rating→?id_rating$_{KX}$) :=
count_kr(agent→?a, trustee→?x, rating→ ?id_rating$_{KX}$).

r_{10}: participate(agent→?a, trustee→?x, rating→?id_rating$_{SX}$) :=
count_sr(agent→?a, trustee→?x, rating→ ?id_rating$_{SX}$).

$r_8 > r_9 > r_{10}$

SELECTING EXPERIENCES

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



SELECTING EXPERIENCES - THEORY #3
PERSONAL EXPERIENCE AND FRIENDS' OPINION IS PREFERRED TO STRANGERS' OPINION

r_8: participate(agent→?a, trustee→?x, rating→?id_rating$_{AX}$) :=
count_pr(agent→?a, trustee→?x, rating→ ?id_rating$_{AX}$).

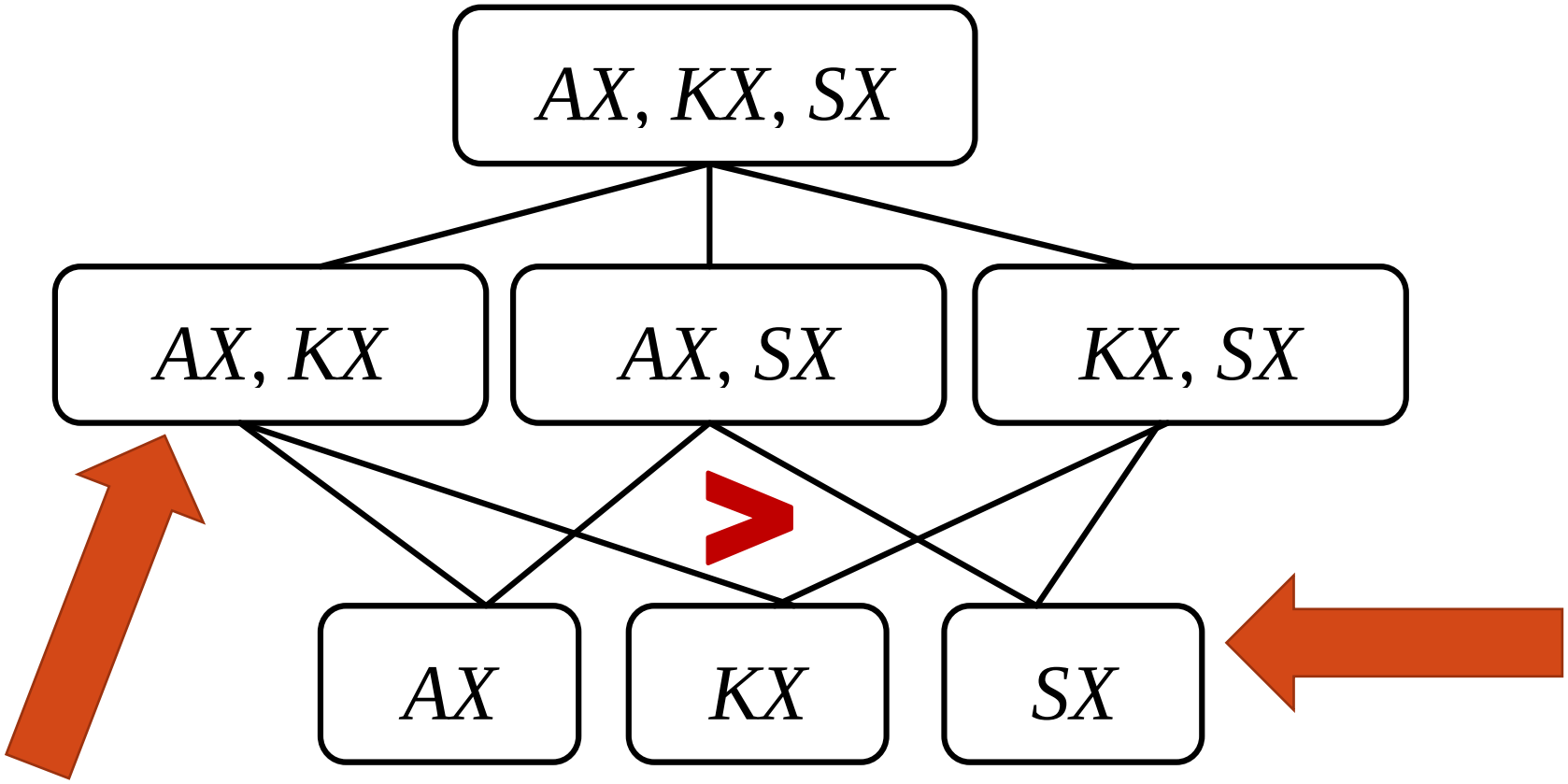
r_9: participate(agent→?a, trustee→?x, rating→?id_rating$_{KX}$) :=
count_kr(agent→?a, trustee→?x, rating→ ?id_rating$_{KX}$).

r_{10}: participate(agent→?a, trustee→?x, rating→?id_rating$_{SX}$) :=
count_sr(agent→?a, trustee→?x, rating→ ?id_rating$_{SX}$).

$r_8 > r_{10}, r_9 > r_{10}$

SELECTING EXPERIENCES

PERSONAL EXPERIENCE AND FRIENDS' OPINION IS PREFERRED TO STRANGERS' OPINION



HARM - TEMPORAL DEFEASIBLE LOGIC EXTENSION

- 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 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 duration that it holds
  - each rule has a delay between the cause and the effect

count_rating(rating→?idx, truster→?a, trustee→?x):duration :=^{delay}

confidence_threshold(?conf),

transaction_value_threshold(?tran),

rating(id→?idx, confidence→?conf$_x$,transaction_value→?tran$_x$)@t,

?conf$_x$ >= ?conf, ?tran$_x$ >= ?tran.

DISARM

K. Kravari, N. Bassiliades, "DISARM: A Social Distributed Agent Reputation Model based on Defeasible Logic", Journal of Systems and Software, Vol. 117, pp. 130–152, July 2016

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

DISARM - SOCIAL RELATIONSHIPS

- 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

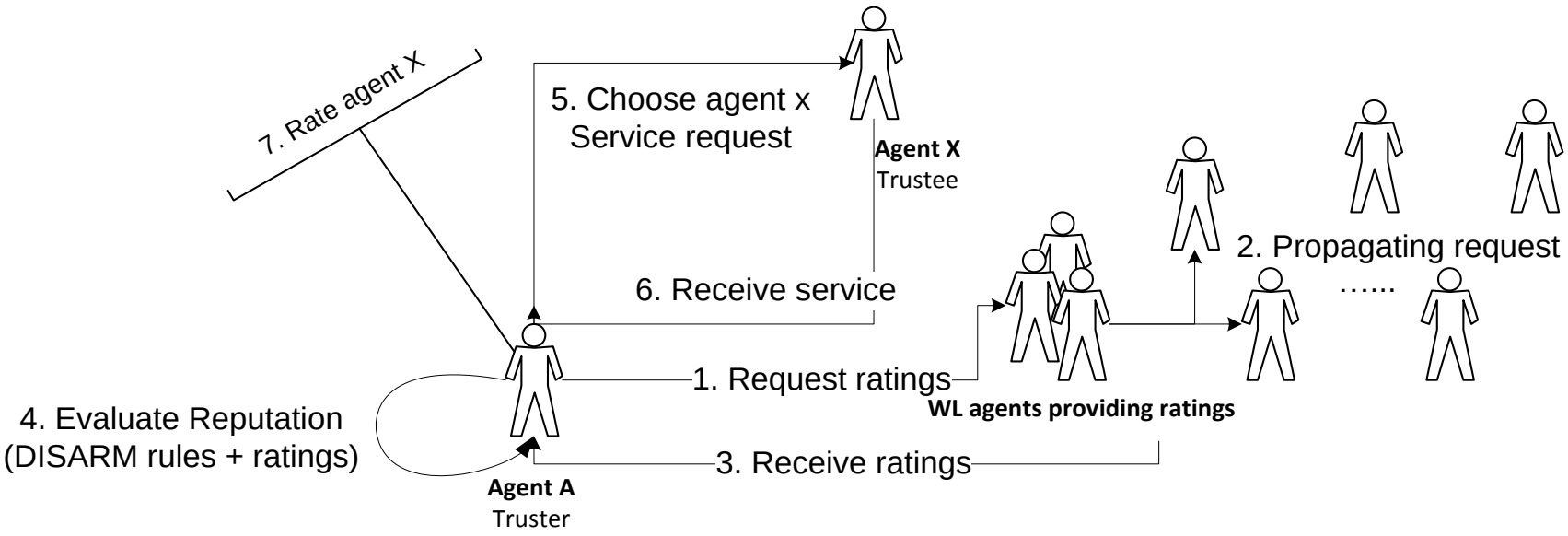
DISARM - RATINGS

- 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):

3 more than HARM

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)

DISARM MODEL



DISARM - BEHAVIOR CHARACTERIZATION

good_behavior(time → ?t, truster→ ?a, trustee→ ?x, reason → all) :-

resp_time_thrshld(?resp), valid_thrshld(?val), …, trans_val_thrshld(?trval),

rating(id→?id_x, 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→?$outf_x$),

?$resp_x$<?resp, ?val_x>?val, ?com_x>?com, ?cor_x>?cor, ?$coop_x$>?coop, ?$outf_x$>?outf.

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

add_whitelist(trustee→ ?x, time → ?t2) :=

good_behavior(time→?t1, truster→?self, trustee→?x, reason→?r),

good_behavior(time→?t2, truster→?self, trustee→?x, reason→?r),

?t2 > ?t1.

- Has been bad thrice for the same reason

add_blacklist(trustee→ ?x, time → ?t3) :=

bad_behavior(time→?t1, truster→?self, trustee→?x, reason→?r),

bad_behavior(time→?t2, truster→?self, trustee→?x, reason→?r),

bad_behavior(time→?t3, truster→?self, trustee→?x, reason→?r),

?t2 > ?t1, ?t3 > ?t2.

DISARM - MAINTAINING RELATIONSHIP LISTS

Add to the blacklist

blacklist(trustee→ ?x, time → ?t) :=

¬whitelist(trustee→ ?x, time → ?t1),

add_blacklist(trustee→ ?x, time → ?t2), ?t2 > ?t1.

Remove from the blacklist

¬blacklist(trustee→ ?x, time → ?t2) :=

blacklist(trustee→ ?x, time → ?t1),

add_whitelist(trustee→ ?x, time → ?t2),

?t2 > ?t1.

Add to the whitelist

whitelist(trustee→ ?x, time → ?t) :=

¬blacklist(trustee→ ?x, time → ?t1),

add_whitelist(trustee→ ?x, time → ?t2), ?t2 > ?t1.

...

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*).

DISARM - HANDLING RATINGS REQUEST

- 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→?t1)):=

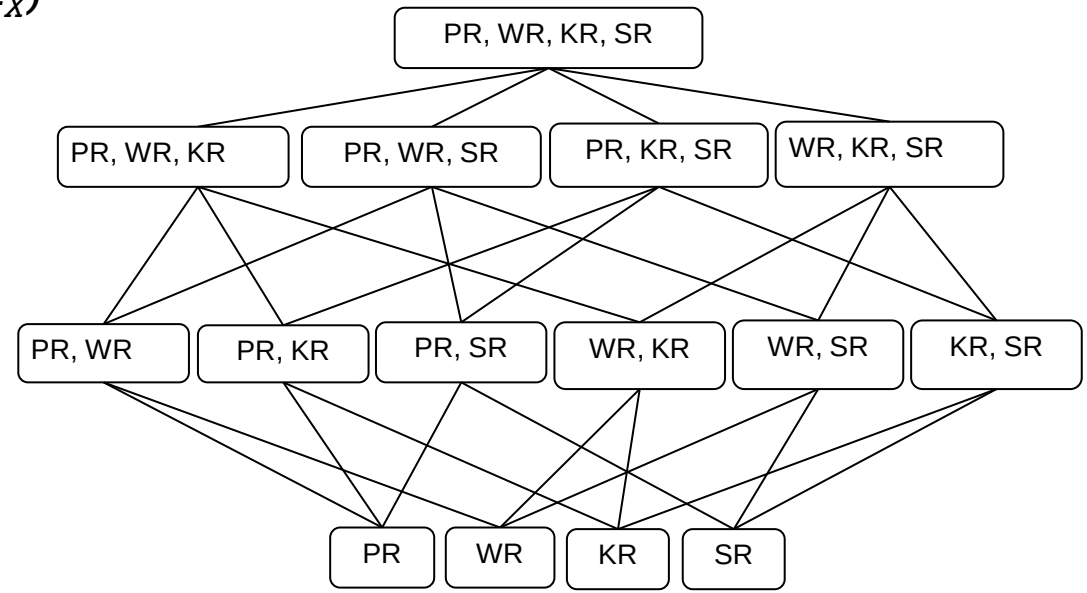
receive_message(sender→?s, receiver→?self,
msg →request_rating(about→?x,ttl→?t)),

?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

eligible_rating(rating→?id_x,truster→?a,trustee→?x,reason→cnf_imp) :=

conf_thrshld(?conf), trans_val_thrshld(?tr),

rating(id→?id_x,truster→?a,trustee→?x,conf→?$conf_x$,trans_val→?tr_x),

?$conf_x$ >= ?conf, ?tr_x >= ?tr.

- According to temporal restrictions

count_rating(rating→?id_x, truster→?a, trustee→?x) :=

time_from_thrshld(?ftime), time_to_thrshld(?ttime),

rating(id→?id_x, t→?t_x, truster→?a, trustee→ ?x),

?ftime <=?t_x <= ?ttime.

DISARM – DETERMINING RATING CATEGORIES

count_wr (rating →?id_x, trustee→?x) :-

eligible_rating(rating → ?id_x, cat→?c, truster→?k, trustee→ ?x),

count_rating(rating→?id_x, truster→?k, trustee→ ?x),

known(agent→?k),

whitelist (trustee →?k).

count_kr (rating →?id_x, trustee→?x) :-

eligible_rating(rating→?id_x, cat→?c, truster→?k, trustee→ ?x),

count_rating(rating→?id_x, truster→?k, trustee→ ?x),

known(agent→?k),

not(whitelist(trustee →?k)),

not(blacklist (trustee →?k)).

DISARM - FACING DISHONESTY

- When ratings provided by an agent are outside the standard deviation of all received ratings, the agent might behave dishonestly

bad_assessment (time → ?t, truster→ ?y, trustee→ ?x) :-
standard_deviation_value(?t,?y,?x,?stdev$_y$),
standard_deviation_value (?t,_,?x,?stdev),
?stdev$_y$ > ?stdev.

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

remove_whitelist(agent→ ?y, time → ?t2) :=
whitelist(truster→ ?y),
time_window(?wtime),
bad_assessment(time → ?t1, truster→ ?y, trustee→ ?x),
bad_assessment(time → ?t2, truster→ ?y, trustee→ ?x),
?t2 <= ?t1 + ?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

- K. Kravari, E. Kontopoulos, N. Bassiliades, "EMERALD: A Multi-Agent System for Knowledge-based Reasoning Interoperability in the Semantic Web", *6th Hellenic Conference on Artificial Intelligence (SETN 2010)*, Springer, LNCS 6040, pp. 173-182, 2010.
- K. Kravari, N. Bassiliades, "HARM: A Hybrid Rule-based Agent Reputation Model based on Temporal Defeasible Logic", *6th International Symposium on Rules: Research Based and Industry Focused (RuleML-2012)*. Springer Berlin/Heidelberg, LNCS, Vol. 7438, pp. 193-207, 2012.
- K. Kravari, N. Bassiliades, "DISARM: A Social Distributed Agent Reputation Model based on Defeasible Logic", *Journal of Systems and Software*, Vol. 117, pp. 130–152, July 2016


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

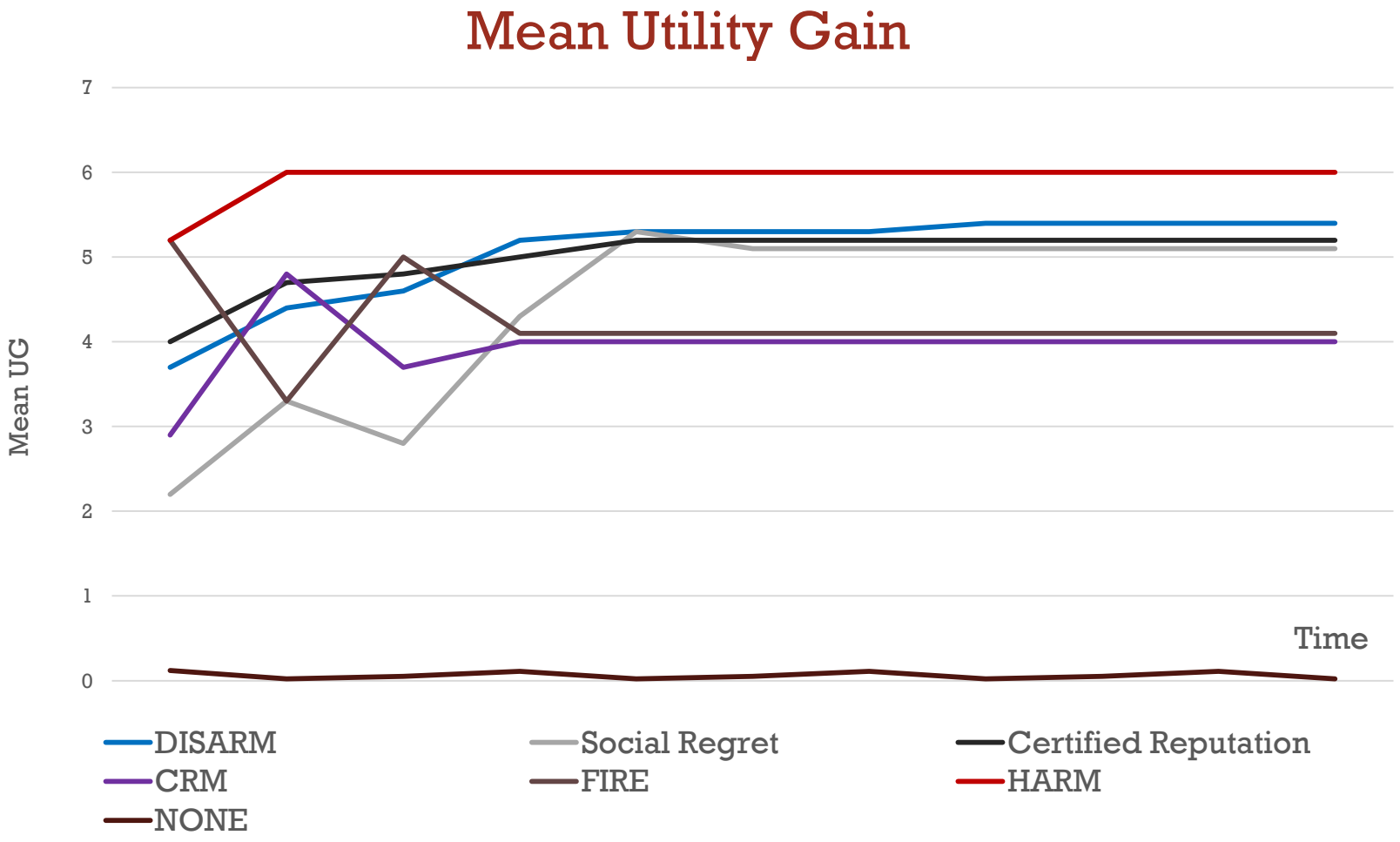
EVALUATION ENVIRONMENT

- 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

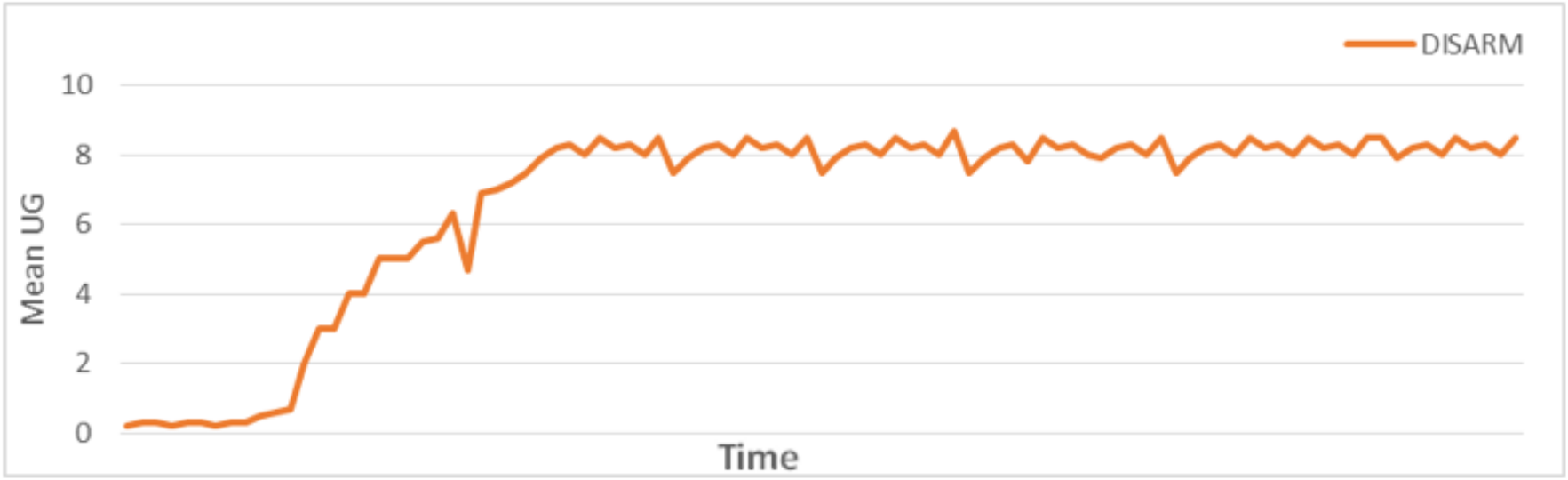
Number of simulations: 500 Number of providers: 100	
Good providers	10
Ordinary providers	40
Intermittent providers	5
Bad providers	45

*K. Kravari, E. Kontopoulos, N. Bassiliades, "EMERALD: A Multi-Agent System for Knowledge-based Reasoning Interoperability in the Semantic Web", 6th Hellenic Conference on Artificial Intelligence (SETN 2010), Springer, LNCS 6040, pp. 173-182, 2010.

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

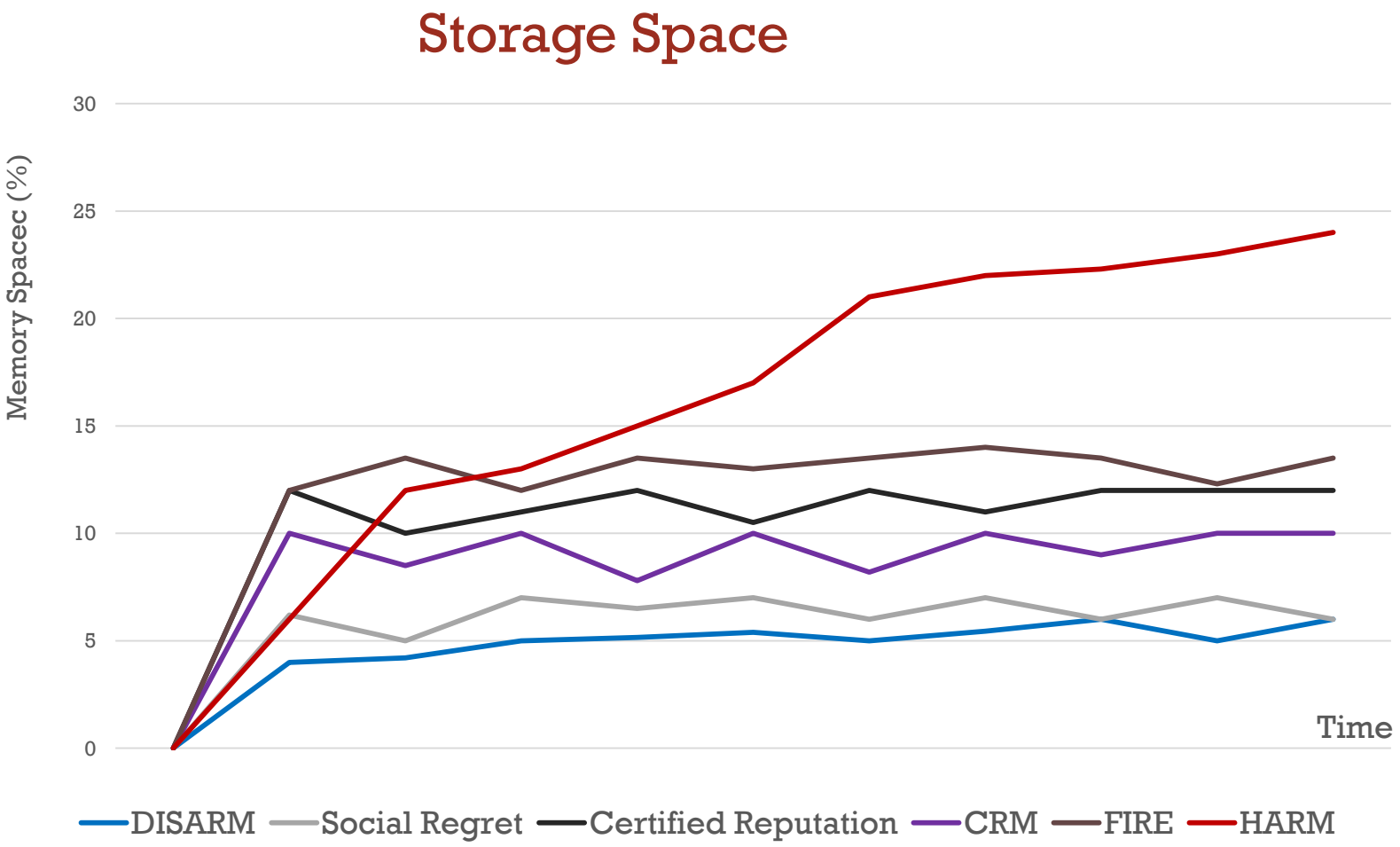


DISARM - ALONE



Better performance when alone,
due to more social relationships

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



EVALUATING DISHONESTY HANDLING