Computational Trust

Perspectives in Concurrency Theory
Chennai December 2008

Mogens Nielsen
University of Aarhus DK

Plan of talk

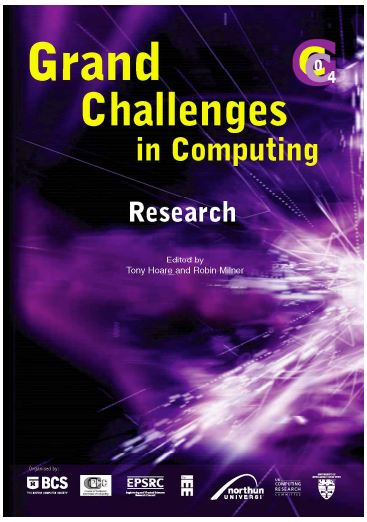
1. Computational Trust and Ubiquitous Computing - a brief survey
2. Computational Trust and Concurrency – a few comments
3. Some results towards rigorously defined Models of Trust

UK Grand Challenge

Engineering and Physical Sciences Research Council

British Computer Society

Institution of Electrical Engineers



UK Grand Challenges in Computing Research

1. In Vivo <=> In Silico
2. Ubiquitous Computing: Experience, Design and Science: UbiComp
3. Memories for Life
4. The Architecture of Brain and Mind
5. Dependable Systems Evolution
6. Non-Classical Computation
7. Learning for Life
8. Bringing the Past to Life for the Citizen

Visions of UbiComp

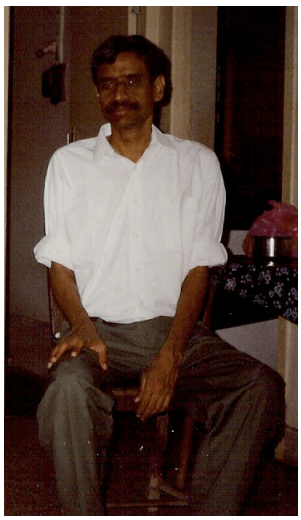
- Billions of autonomous mobile networked entities
  - Mobile users
  - Mobile software agents
  - Mobile networked devices:
    - Mobile communication devices (phones, pagers, …)
    - Mobile computing devices (laptops, palmtops, …)
    - Commodity products (embedded devices)
- Entities will collaborate with each other
  - Resource sharing
    - Ad hoc networks, computational grids, …
  - Information sharing
    - Collaborative applications, recommendation systems, …

Data Security in UbiComp

- Data Security related properties of UbiComp
  - Large number of autonomous entities
  - Large number of administrative domains
  - No common trusted computing base
  - Virtual anonymity
- - excluding the use of traditional security mechanisms used in distributed systems – e.g. passwords, certificates, keys,...!
- ONE alternative approach:
  Trust based data security

Trust between humans

Would you buy
a used car
from this man?



Trust Surveys

- Trust in the Social Sciences
  - D. H. McKnight, N.L. Chervany: *The Meaning of Trust,* Trust in Cyber-societies, Springer LNAI 2246, 2001

McKnight and Chervany

- TRUST
  - Disposition
  - Structural
  - Affect/Attitude
  - Belief/Expectancy
  - Intention
  - Behaviour
- TRUSTEE
  - Competence
  - Benevolence
  - Integrity
  - Predictability
  - Openness, carefulness,..
  - People, Institutions,…

Computational Trust

- Decisions related to data security made autonomously based on
  - entities' behaviour, reputation, credentials,..
  - other entities' recommendations,..
  - incomplete information, contexts, mobility,…
- Decisions related to data security made autonomously based on
  - a suitable *computational* notion of trust in order to achieve some required properties of communication between entities

Computational Trust Surveys

- Computational Trust in UbiComp
  - T. Grandison, M. Sloman: *A Survey of Trust in Internet Applications*, IEEE Communications Surveys & Tutorials, 3(4), 2000
  - A. Jøsang, R. Ismail, C. Boyd: *A Survey of Trust and Reputation for Online Service Provision*, Decision Support Systems, 43(2), 2006

Computational Trust

- Trust formation
  - Individual experience
  - Recommendation from known (trusted) third parties
  - Reputation (recommendation from many strangers)
- Trust evolution
  - Incorporating new trust formation data
  - Expiration of old trust values
- Trust exploitation
  - Risk analysis
  - Feedback based on experience
  - Context dependence

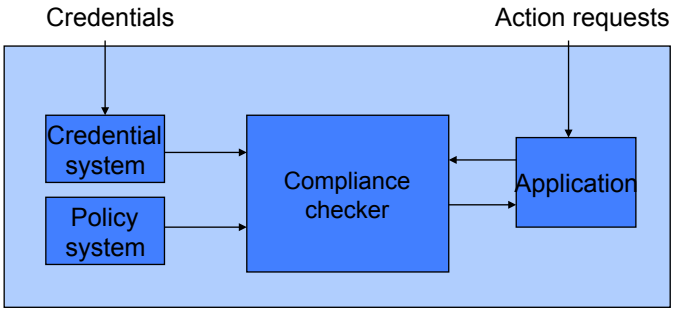
Computational Trust Applications

- Information provider applying trust in requesters
  - e.g. should I allow requester *R* to access my resource *r*?
    - Data security, Access control,..
- Information requester applying trust in providers
  - e.g. which of providers *P, Q, R,..* will provide the best service *s* for me?
    - Quality of services,..

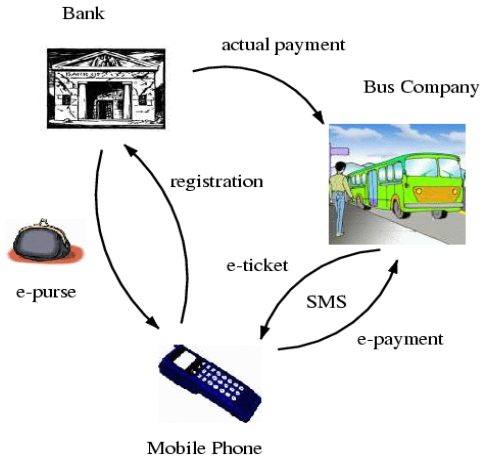
Computational Trust Systems

- *Credential based*
  - the *KeyNote* System of Blaze et al
  - the *Delegation Logic* of Li et al
  - .....
- *Reputation based*
  - the *Eigentrust System* of Kamvar et al
  - the *Beta Reputation System* of Jøsang et al
  - .....

Credential Based Trust Management
Blaze, Feigenbaum et al



E-Purse Scenario



E-Purse Scenario

- Consider a situation where a user is considering requesting an amount *m* of e-cash from a bank
- Seen from the point of view of the user, an "untrusted " bank may
  - deny the request, e.g. because the bank server is down for maintenance
  - grant the request, but withdraw an amount different from *m* from users account
  - grant the request, but the transferred e-cash may be forged

Reputation Based EigenTrust Algorithm
Kamvar et al

- Peers (i,j,..) interact and mutually rate interactions as being either *satisfactory* or *unsatisfactory*:
  - s_{ij} = max ($N_{sat}(i,j)$ - $N_{unsat}(i,j)$, 0)
- These ratings are normalised
  - c_{ij} = s_{ij} / Σ_j s_{ij}
- [c_{ij}] is a Markov chain with stationary distribution [t_j] - interpreted as the global trust in peer j

EigenTrust Algorithm for P2P Networks

- System simulations show that EigenTrust can significantly reduce the number of non-authentic file downloads in a P2P filesharing system, even when up to 70% of the peers maliciously cooperate
- But what is Eigentrust computing, - e.g. what does it mean that the trust in some peer is .75?

Plan of talk

1. Computational Trust and Ubiquitous Computing - a brief survey
2. Computational Trust and Concurrency – a few comments
3. Some results towards rigorously defined Models of Trust

UbiComp: *Computational* Trust

- On *trust:*

  "..*trust* between autonomous agents will be an important ingredient..... A discipline of trust will only be effective if it is *rigorously defined*..."

- On *rigorously defined:*

  "...tools for formalization, specification, validation, analysis, diagnosis, evaluation, ..."

Computational Trust and Concurrency

Lots of opportunities for young and talented scientists!

Some Publications

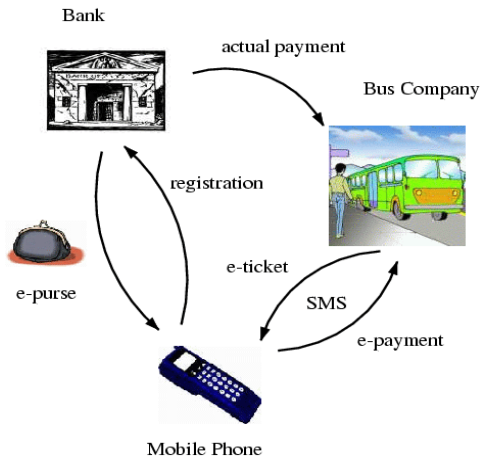
- Nielsen, Krukow, Sassone: *Trust Models in Ubiquitous Computing*, Phil. Trans. of the Royal Society, 2008
- Nielsen, Krukow, Sassone: *A Bayesian Model for Event-based Trust*, Electronic Notes in Theoretical Computer Science, 2007
- Nielsen, Krukow, Sassone: *A Logical Framework for Reputation Systems*, Journal of Computer Security, 2007
- Nielsen, Krukow, Sassone: *Towards a Formal Framework for Computational Trust*, 5th International Symposium on Formal Methods for Components and Objects, 2007
- Nielsen, Krukow, 2007, *Trust Structures*, International Journal of Information Security, 2007
- Krukow, Nielsen: *From Simulations to Theorems*, FAST'06, 2007

Plan of talk

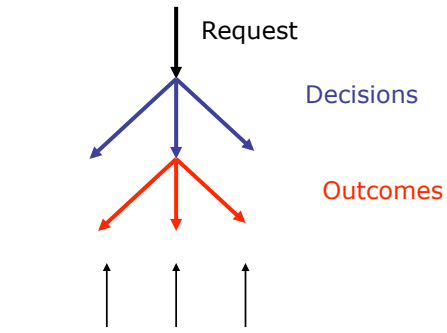
1. Computational Trust and Ubiquitous Computing - a brief survey
2. Computational Trust and Concurrency – a few comments
3. Some results towards rigorously defined Models of Trust

   joint work with Karl Krukow, Vladimiro Sassone and Catuscia Palamidessi

E-Purse Scenario

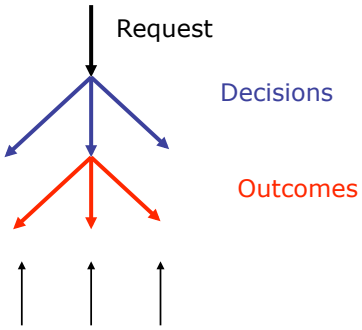


Trust/Risk Based Decisions



Trust based expected costs

Probabilistic Computational Trust

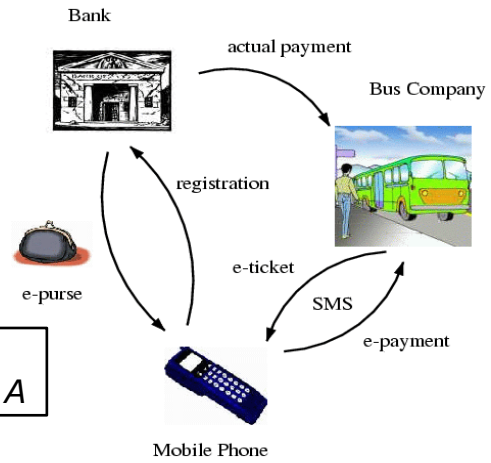


$$\exp = \sum_i \text{cost}(o_i) * \text{likelihood}(o_i)$$

Models and Algorithms

Bank
Model M

Phone
Algorithm A



Probabilistic Models for Computational Trust

- Given a (finite) set of outcomes of interactions
  - $O = \{o_1, o_2,...,o_m\}$
- A probabilistic model M of principal behaviour defines for $h \in O^*$ and $o_i \in O$
  - $P(h \mid M)$ - the probability of observing h in M
  - $P(o_i \mid h\ M)$ - the probability of o_i in the next interaction given observation h in M

Probabilistic Computational Trust Algorithms

- Given a (finite) set of outcomes of interactions
  - $O = \{o_1, o_2,...,o_m\}$
- A probabilistic computational trust algorithm A
  - takes as input a history $h \in O^*$ and
  - outputs a probability distribution on O

  $A(o_i \mid h) \in [0,1]$ for $i = 1,2,..,m$

The Goal for Probabilistic Trust Algorithms

- Algorithm *A* producing $A(o_i \mid h)$ should approximate Model *M* probabilities $P(o_i \mid h\ M)$ as well as possible!
- Notice that this gives rise to rigid versions of soft correctness question:
  - how well does a particular algorithm approximate the model?
  - how robust is it - wrt. the model and its parameters?

Probabilistic Trust Algorithms

- Focus on two example algorithms:
  - P2P Reputation Management of Despotocvic et al
  - Computational Model for eBusiness of Mui et al

A Concrete Simple Probabilistic Model

- The Bernoulli Model – $M_B(\theta)$
  - Assume that the behaviour of a particular principal, *p*, has only two outcomes, with a probability θ for *success* (and *1-* θ for *failure*)
- Algorithm *A*
  - Output: a probability distribution $\{s, f\} \rightarrow [0, 1]$
- The Goal
  - *A* should approximate $(\theta, 1-\theta)$ as well as possible

Despotovic et al 2004: *Algorithm* A_D

- The Specification (of trust computation algorithm *A*)
  - Input: a sequence of observations $h = x_1x_2..x_n \in \{s, f\}^*$
  - Output: a probability distribution $\{s, f\} \rightarrow [0, 1]$
- Maximum-Likelihood: $ML_\theta (P(h \mid M_B(\theta)))$
- The algorithm A_{ML} for $M_B(\theta)$
  - $A_D(s \mid h) = N_s(h) / |h|$
  - $A_D(f \mid h) = N_f(h) / |h|$

Mui et al 2002: *Algorithm* A_M

- The Specification (of trust computation algorithm A)
  - Input: a sequence of observations $h = x_1x_2..x_n \in \{s, f\}^*$
  - Output: a probability distribution $\{s, f\} \rightarrow [0, 1]$
- Bayes' Rule with a Uniform Prior (*Beta(1, 1)*)
- The algorithm A_{BU}:
  - $A_M(s \mid h) = (N_s(h) + 1) / (|h| + 2)$
  - $A_M(f \mid h) = (N_f(h) + 1) / (|h| + 2)$

A Question: how to choose

- The Goal
  - Algorithm A should approximate $(\theta, 1- \theta)$ as well as possible
- Which of the two algorithms A_{ML} and A_{BU} performs best relative to this goal?
  - *Experimental approach*: answers given based on experiments in simulation environments
  - *Theoretical approach*: answer given in terms of mathematical results in our probability model

A Measure: Relative Entropy

- The Relative Entropy (also called the Kullback-Leibler divergence) of two probability distributions $\boldsymbol{p}$ and $\boldsymbol{q}$ (here on $O = \{o_1, o_2, \ldots, o_m\}$) is defined as

$$D(\boldsymbol{p} \,||\, \boldsymbol{q}) = \sum_i \boldsymbol{p}(o_i) \times \log_2\left(\boldsymbol{p}(o_i) / \boldsymbol{q}(o_i) \right)$$

- Well established pseudo-distance measuring "the distance from a true distribution $\boldsymbol{p}$ to an approximation $\boldsymbol{q}$"

The Goal of a Probabilistic Algorithm: Formally

- The Goal
  - Algorithm A producing $A(o_i \mid h)$ should approximate $P(o_i \mid h\, M)$ as well as possible

- We choose to interpret "as well as possible" in terms of the expected distance between the two distributions:

$$ED^n(\boldsymbol{M} \,||\, \boldsymbol{A}) = \sum_{h \in O^n} p(h \mid \boldsymbol{M}) \times D\left(P(\cdot \mid h\boldsymbol{M}) || \boldsymbol{A}(\cdot \mid h) \right)$$

How to choose: Formally

- Comparing A_D and A_M against M_B:

  If $\theta = 0$ or $\theta = 1$ then for all n

  $ED^n(M_B(\theta), A_D) = 0 < ED^n(M_B(\theta), A_M)$

  If $0 < \theta < 1$ then for all n

  $ED^n(M_B(\theta), A_M) < ED^n(M_B(\theta), A_D) = \infty$

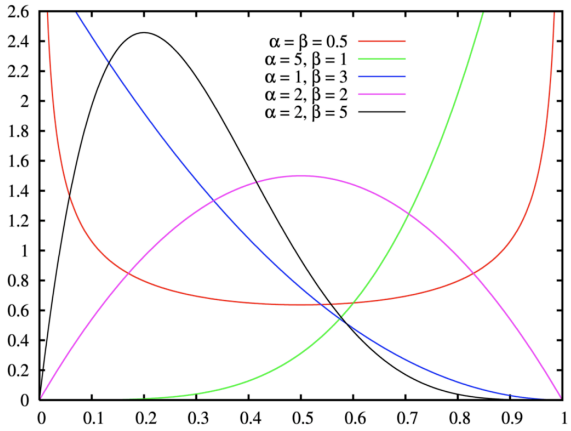
Bayesian Approach

- Bayes' theorem:

  $P(\theta \mid h,M) = P(\theta \mid M) \times (P(h \mid \theta,M) / P(h \mid M))$

- For the model M_B choosing
  - $P(\theta \mid M_B) = Beta(a, \beta)\ (\theta)$
- Allows the following simple "algorithms" computing the a posteriori information
  - $P(\theta \mid h,M_B) = Beta(a + N_s(h), \beta + N_f(h))$
  - $E\ (Beta(a, \beta)) = a / (a + \beta)$

An example Model *M*: *Beta (a, β)*



Two Examples Generalised

- A_D the P2P Reputation Management of Despotocvic et al
  - an example of the Bayesian approach with $a=\beta=0$
- A_M the Computational Model for eBusiness of Mui et al
  - an example of the Bayesian approach with $a=\beta=1$

- Generalize to all uniform Beta priors, i.e. for arbitrary real numbers $\varepsilon \geq 0$:
  - $A_\varepsilon(s \mid h) = (N_s(h) + \varepsilon) / (|h| + 2\varepsilon)$
  - $A_\varepsilon(f \mid h) = (N_f(h) + \varepsilon) / (|h| + 2\varepsilon)$
- What is a good choice of ε - and how does this choice depend on θ and n?

Some Theoretical Answers: how to choose

For any $\theta \in [0,1]$, $\theta \neq 1/2$, there exists an ε_θ which for any n minimizes $ED^n(M_B(\theta), A_\varepsilon)$. Furthermore, ε_θ is defined as the following function of θ

$$\varepsilon_\theta = 2\theta(1-\theta) \;/\; (2\theta-1)^2$$

Meaning: unless behaviour is completely random, there is a unique best algorithm (choosing $\varepsilon := \varepsilon_\theta$) outperforming all other A_ε algorithms, $\varepsilon \geq 0$

Some Theoretical Answers: *Robustness*

Furthermore, $ED^n(M_B(\theta), A_\varepsilon)$ is continuous (as a function of ε) – decreasing on the interval $(0, \varepsilon_\theta)$ and increasing on $(\varepsilon_\theta, \infty)$

Meaning: The closer ε is to ε_θ the better performance of A_ε

Another Distance Measure

- The distance between two probability distributions ***p*** and ***q*** (here on $O = \{o_1, o_2, \ldots, o_m\}$) is defined as

$$D(\mathbf{p} \,||\, \mathbf{q}) = \sum_i \left(\mathbf{p}(o_i) - \mathbf{q}(o_i) \right)^2$$

- Well established distance measuring "how well two probability distributions approximate each other"
- Results robust wrt choice of measure!

Some Theoretical Answers: how to choose

Given a particular ε, the algorithm A_ε is an optimal choice (for all n, and amongst all the A_ε algorithms) for

$$\theta = {}^1/_2 \;+/-\; {}^1/_{2\ sqrt(2\varepsilon+1)}$$

Example: A_M is optimal for $\theta = {}^1/_2 \;+/-\; {}^1/_{sqrt(12)}$

Non-uniform priors

- Using the prior *Beta(a, β)* yields the following algorithm computing the mean of the posterior distribution:
  - $A_{\alpha,\beta}\,(s \mid h) \quad = \quad (N_s(h) + \alpha) \,/\, (|h| + a + \beta)$
  - $A_{\alpha,\beta}\,(f \mid h) \quad = \quad (N_f(h) + \beta) \,/\, (|h| + a + \beta)$
- How to choose the parameters α and *β?*

Non-uniform priors

- Assume no knowledge of the true behaviour (θ in M_B), define the "risk" of an algorithm $A_{\alpha,\beta}$
- $R^n(A_{\alpha,\beta}) \quad = \quad \int_{[0;1]} ED^n\,(M_B(\theta), A_{\alpha,\beta})\, d\theta$
- *Theorem*
  For all *n*, $R^n(A_{\alpha,\beta})$ is minimum for α = β = 1

Non-uniform priors

- Assume the true behaviour (M_B) to be $Beta(a_t, \beta_t)$, define the "risk" of an algorithm $A_{a,\beta}$
- $R^n(A_{a,\beta}) = \int_{[0;1]} Beta(a_t, \beta_t)\, ED^n(M_B(\theta), A_{a,\beta})\, d\theta$
- *Theorem*
  For all n, $R^n(A_{a,\beta})$ is minimum for $a = a_t$ and $\beta = \beta_t$

Many More Issues to be Modelled....

- Trust formation
  - Individual experience
  - Recommendation from known (trusted) third parties
  - Reputation (recommendation from many strangers)
- Trust evolution
  - Incorporating new trust formation data
  - Expiration of old trust values
- Trust exploitation
  - Risk analysis
  - Feedback based on experience
  - Context dependence

Expiration of old trust values

- In order to cope with dynamically changing behaviour, it has been suggested to decay the observations in *h*
  - exponentially, linearly, …
- Under what circumstances is this a sensible approach, and if so, which of the many proposed decay functions are "best"?

Towards more answers

- The underlying model is now naturally a Hidden Markov Model - often used to model dynamically changing behavior
- Scene is set to investigate the questions above in terms of properties of e.g.

$$ED^n(M_{HMM} \,||\, A_{exp}) \quad \text{and} \quad ED^n(M_{HMM} \,||\, A_{lin})$$

where

$$ED^n(\boldsymbol{M} \,||\, \boldsymbol{A}) = \Sigma_{h \in O^n}\, p(h \mid \boldsymbol{M}) \;\times\; D(\, P(\cdot \,|\, h\boldsymbol{M}) || \boldsymbol{A}(\cdot \,|\, h))$$


Thank you !