INFORMATICA, 2014, Vol. 25, No. 1, 139–154

Computational Trust Management, QAD, and Its Applications

Denis TRČEK
Laboratory of e-Media, Informatics Deptartment
Faculty of Computer and Information Science, University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana, Slovenia
e-mail: denis.trcek@fri.uni-lj.si

Received: April 2012; accepted: September 2012

Abstract. Trust is an important factor for successful e-commerce and e-media applications. However, these media inherently disable many ordinary communication channels and means, and affect trust forming factors. Therefore cyber environment requires additional support when it comes to trust. This is also one key reason why computational trust management methods are being developed now for some fifteen years, while another key reason is to enable better decision making through mathematical modeling and simulations in other areas. These methods are grounded on certain premises, which are analyzed in this paper. On this basis, Qualitative assessment dynamics (QAD for short) is presented that complements the above methods. As opposed to other methods, it is aligned with certain principles of human reasoning. Therefore it further extends the scope of other computational trust management technologies that are typically concerned with artificial ways of reasoning, while QAD gives a basis also for applications in ordinary environments where humans are involved. By using this methodology, experimental work will be presented, applied to the area of organizations and human factor management.

Key words: e-services, trust, modeling, simulation, human factor, management.

1. Introduction

Trust management systems in computing environments have many interesting applications. Considering not only most common applications like e-commerce, they can be deployed for infrastructural communications purposes like Quality of services, QoS, where nodes would exchange trust related information about other nodes and accordingly route traffic to exclude misbehaving nodes (Schonwalder *et al.*, 2009). Further, there exist applications of such systems for e-environments like virtual auctions and automated stocks trading, which demonstrates how trust management can be incorporated as one key pillar of autonomic computing that is concerned with improved ability of network and services with unpredicted changes (Dobson *et al.*, 2006). Moreover, recent applications deploy such systems also in business intelligence for, e.g., finding strategies for mobile services and applications (Raza *et al.*, 2011).

So it is no wonder why research of trust management is now taking place for over a decade in computing society. During this period many promising methods have been developed, but it is turning out that it is not likely that only one of them will prevail. Based on deployment context, various methods (or their combinations) will likely coexist. To complement other computational trust management methods, Qualitative assessment dynamics, or QAD, is being developed now for almost ten years. In the beginning it was developed in an informal way, while in 2009 its basics were formally presented in Trček (2009). Since then some informal extensions followed – these are mainly summarized in Trček (2011), while some recent ones can be found in Zupančič and Trček (2012). This paper gives additional extensions (to QAD matrices calculus) that are needed to improve modeling of the dynamics of trust in societies. Further, it provides complete formalization of the apparatus extensions since 2009. Last but not least, it provides now concrete ways of applications not only in e-environments, but also in human factor and organizations management, i.e., in decision making support. Therefore QAD is not only a formal system, but can be seen also as a simulation framework – clearly, trust phenomenon and its research is about complex systems where pure analytic solutions are rather exceptions than a rule. Therefore computer simulations are (and will remain) a valuable tool for researching this field.

In line with the above reasoning, this paper first provides a brief analysis of some most important trust management methods in section two. Afterwards it formally presents QAD in section three, gives experimental application of QAD to organizations management in section four together with implementation issues, and states conclusions with future research directions in section five. The paper is rounded up by acknowledgments and references.

2. A Brief Overview of the Field

This section provides an overview of typical computational trust management methods that have been developed during the last decade, and gives also their brief analysis.

- Naive trust management (Wang and Vassileva, 2003) – with this approach agents use Bayesian inference. By deciding on a certain probability threshold, obtained conditional probabilities from past interactions are used to calculate whether this threshold is met or not, and consequently, whether to get into new interaction or not. More precisely, each agent builds a table, where two columns contain vales about trusted and non-trusted interactions, and (one or more) rows contain attributes of these interactions (e.g., for books transactions these may be book quality, book readability, book delivery time, etc.). After each interaction appropriate values are put into cross-sections and they denote basic conditional probabilities: value in the first column is $p(T = 1) = m/n$ and in the second $p(T = 0) = n - m/n$, where m stands for number of trusted interactions, and n for all interactions. If we consider a row that is about book exchange and the first column, then this denotes conditional probability $p(bookExch|T = 1)$. After having sufficient number of interactions, these values can be used to address more complex questions that include generalized forms of conditional probabilities like: What is the probability that the given transaction be trusted, given that it is about book exchange where the book has to be of a high quality and efficiently delivered?

- Theory of evidence, ToE (Shafer, 1976), and Subjective logic, SL (Josang, 2001) – Theory of evidence can be seen as an extension of Bayesian inference, where the basic probabilities of possible events are subject to new functions. The final result of their mappings is as follows. With classical probability, an event x probability results in splitting the probability interval [0, 1] into intervals $p(x)$ and $1 - p(x)$ (assuming that $p(x)$ and $p(\neg x)$ is a sure event). With new functions defined in ToE and SL the interval [0, 1] is split into three sub-intervals: into belief part $b(x)$ (which can be roughly seen as an analog to $p(x)$), into disbelief part $d(x)$ (which can be roughly seen as analog to $p(\neg x)$), and into uncertainty part $u(x)$ (which does not have an adequate analog). Thus $u(x) = 1 - (b(x) + d(x))$. Now having b, d and u values as a basis, operators of Subjective logic are defined that preserve hard formal basis of Theory of evidence, while enabling modeling trust processes among agents.
- Game theory (Tennenholz, 2008) – game theory studies the ways in which strategic interactions among economic agents produce outcomes with respect to their preferences (defined by utilities functions), where the outcomes might have been intended by none of the agents (Ross, 2010). So studying trust within game theory can be seen somewhat similar to the above two approaches. However, in this case we do not deal with (explicit) probabilities, but anticipation of some actor(s) actions who take(s) into account other actors actions. Clearly, with each of these actions agents try to maximize their utilities.
- Multi-agent systems (Sabater and Sierra, 2005) – multi-agent systems (MAS) are deploying modeling of various kinds of behavior of artificial agents through defining their reasoning. The basis for this reasoning is often one of the above mentioned methods, but also some advanced machine learning methodology can be deployed (one such example is given in Rettinger *et al.* (2011) where trust is treated in relational way by also including context). MAS based research has an advantage of enabling study of behavior of whole societies during longer periods, and making the dynamics of this behavior visible. Therefore it is also the basis for QAD.

QAD complements these methods mentioned above. From users point of view, it is (probably the only one) human-agent focused methodology, while being a formal system and otherwise belonging to computational trust management methods. More precisely, QAD is based on operands and operators that are used in human reasoning processes and reflected in languages. Put another way, QAD operators and operands have linguistic basis and thus they are meaningful and understandable in many cultural settings. Further, QAD addresses many important issues related to trust (management) when it comes to human agents:

- First, human-agents are not (always) rational, which is proven now in numerous scientific articles (in economics settings probably the most influential such research is prospect theory Kahneman and Tversky, 1979).
- Second, human-agents will likely not understand sophisticated mathematics behind methods like Theory of evidence and similar (this claim is partially also a consequence of the first claim).

- Third, human-agents, when it comes to trust, may have no preferences. Even if they have preferences, these may not be transitive, which are both the necessary conditions for any game-theoretic approach.
- Fourth, when it comes to assessing trust, a significant number of users would opt for a five level (Likert like) ordinal scale of assessments. In addition, users often form initial assessments on a non-identifiable basis, and the same holds true for their occasional changes (Trček, 2011).
- Fifth, trust assessments cannot be generally considered as reflexive, or symmetric, or transitive (this can be verified also by a few simple mental experiments).

More detailed elaboration of the above facts can be found in Trček (2011). As to the historical development of QAD, it was initially an algebraic structure, i.e., a group. But following the main idea that it should reflect human reasoning as close as possible, it was soon redefined and resulted in semi-group (Trček, 2009; Kovač and Trček, 2010). However, further incorporation of other research required additional rework that has resulted in a stable mathematical structure, a formal system on its own, which is in the focus of this paper.

3. Formal Presentation of QAD

Despite taking into account the above reflections of the real-world needs, QAD is a formal system that enables rigorous mathematical and computational treatment. Let us introduce it with the definition of trust assessment matrix:

Definition 1. Entity i assessment of entity j is a qualitatively weighted relation denoted as $\alpha_{i,j}$, where weights can be totally trusted, partially trusted, undecided, partially distrusted, and totally distrusted (denoted as 2, 1, 0, -1, -2). In cases where assessment is not known or disclosed it is denoted by "$-$".

Definition 2. Assessments matrix A of a society with n entities consists of elements $\alpha_{i,j} \in \{-2, -1, 0, 1, 2, -\}$, where $\alpha_{i,j}$ denotes entity's i assessment of entity j, and $i, j = 1, 2, \ldots, n$.

Columns in assessments matrix are referred to as trust vectors, because a column contains assessments of all entities in a community toward an observed entity. As it will be seen in the rest of the paper, operators are applied to trust vectors, where these trust vectors are operands.

Definition 3. Let $\kappa = \{1, 2, \ldots, n\}$, where n is the number of entities in a society. When the i-th row of the assessment matrix A is preserved, while all other rows are filled with elements "$-$", then this matrix is referred to as the (i)-th constituency matrix, A^i, of assessment matrix A. When more than one row is preserved in the same manner, such matrix is referred to as partial assessment matrix and it is denoted by A^{κ_p}, where κ_p denotes the set with the sequence of the preserved rows from κ. Further, two partial assessment matrices A^{κ_i} and A^{κ_j} are referred to as non-overlapping matrices when $\kappa_i \cap \kappa_j = \emptyset$.

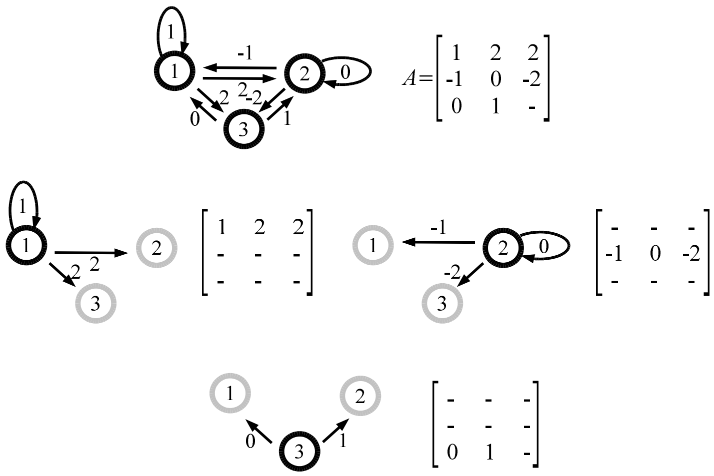


Fig. 1. An example society with assessment matrix and its decomposition into constituent assessment matrices.

Thus constituent matrices are a special case of partial assessment matrices that are characterized by having at most one row of assessments with elements being different from "−" (an example of splitting assessments matrix into constituent matrices is given in Fig. 1). It should be noted that some societies may already initially produce partial constituency matrices, e.g., societies that contain one or more dumb entities (in this case the rows that correspond to dumb entities contain only elements "−").

Definition 4. Operator θ_i is a function with the argument being trust vector values $[\alpha^-_{1,j}, \alpha^-_{2,j}, \ldots, \alpha^-_{n,j}]^T$ and the mapped value being a single assessment value $\alpha^+_{i,j}$, where the pre-operation value is denoted by superscript "−", while the after-operation value is denoted by superscript "+".

Definition 5. Operators set consists of the following operators (n_1 is obtained by subtracting the number of undefined assessments in a trust vector from n; $i, j = 1, 2, \ldots, n$):

$$\Uparrow_i : \alpha^+_{i,j} ::= \max\left(\alpha^-_{1,j}, \alpha^-_{2,j}, \ldots, \alpha^-_{n_1,j}\right),$$

$$\Downarrow_i : \alpha^+_{i,j} ::= \min\left(\alpha^-_{1,j}, \alpha^-_{2,j}, \ldots, \alpha^-_{n_1,j}\right),$$

$$\uparrow_i : \begin{cases} \alpha^+_{i,j} ::= \alpha^-_{i,j} & \text{if } \frac{1}{n_1}\sum_{k=1}^{n_1} \alpha^-_{k,j} \leqslant \alpha^-_{i,j}, \\ \alpha^+_{i,j} ::= \alpha^-_{i,j} + 1 & \text{otherwise,} \end{cases}$$

$$\downarrow_i : \begin{cases} \alpha^+_{i,j} ::= \alpha^-_{i,j} & \text{if } \frac{1}{n_1}\sum_{k=1}^{n_1} \alpha^-_{k,j} \geqslant \alpha^-_{i,j}, \\ \alpha^+_{i,j} ::= \alpha^-_{i,j} - 1 & \text{otherwise,} \end{cases}$$

$$\rightsquigarrow_i : \begin{cases} \alpha^+_{i,j} ::= \left\lceil \frac{1}{n_1}\sum_{k=1}^{n_1} \alpha^-_{k,j} \right\rceil & \text{if } \frac{1}{n_1}\sum_{k=1}^{n_1} \alpha^-_{k,j} < 0, \\ \alpha^+_{i,j} ::= \left\lfloor \frac{1}{n_1}\sum_{k=1}^{n_1} \alpha^-_{k,j} \right\rfloor & \text{otherwise,} \end{cases}$$

$$\leftrightarrow_i: \begin{cases} \alpha_{i,j}^{+} ::= \left\lceil \frac{1}{n_1} \sum_{k=1}^{n_1} \alpha_{k,j}^{-} \right\rceil & \text{if } \frac{1}{n_1} \sum_{k=1}^{n_1} \alpha_{k,j}^{-} > 0, \\ \alpha_{i,j}^{+} ::= \left\lfloor \frac{1}{n_1} \sum_{i=1}^{n_1} \alpha_{k,j}^{-} \right\rfloor & \text{otherwise,} \end{cases}$$

$$\odot_i: \alpha_{i,j}^{+} ::= \alpha_{i,j}^{-},$$

$$\updownarrow_i: \alpha_{i,j}^{+} ::= random(-2, -1, 0, 1, 2),$$

$$*_i: \alpha_{i,j}^{+} ::= \text{“} - \text{”}.$$

Definition 6. Regardless of an operator, if the pre-operation value (assessment) is not known or disclosed, i.e., $\alpha_{i,j}^{-} = -$, it remains unchanged also after the operation, i.e., $\alpha_{i,j}^{+} ::= \text{“} - \text{”}$ $(i = 1, 2, \ldots, n)$.

The names of the above operators go in turn as follows: extreme optimistic assessment operator, extreme pessimistic assessment operator, moderate optimistic and moderate pessimistic assessment operator, centralist operator, non-centralist operator, self-confidence assessment operator, assessment hoping operator, and assessment hiding operator. To enable better understanding of functioning of these operators, their informal description is provided:

- Totally trusted assessment operator models such kind of reasoning where an agent takes for his/her new assessment the most positive value in a trust vector, while totally distrusted assessment operators does the opposite.
- Partially trusted assessment operator models such kind of reasoning where an agent averages values in a trust vector and in case of rounding rounds it up to the next higher assessment value – partially distrusted assessment operators does the opposite.
- Centralist consensus seeker operator models such kind of reasoning where an agent averages values in a trust vector and rounds the result towards undecided value (in case of positive average value it rounds it down, while in case of negative value it rounds it up), while the non-centralist consensus seeker makes rounding in the opposite direction.
- Self-confident operator models such kind of reasoning where an agent sticks with the same assessment that he/she had in previous iterations.
- Assessment hoping operator models such kind of reasoning where assessments are (randomly) changed on a non-identifiable basis.
- Assessment hiding operator models such kind of reasoning where an agent decides not to express his/her assessment in the rest of iterations.

Definition 7. Operators vector $\Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$ consists of elements $\theta_i \in \{\Uparrow, \Downarrow, \uparrow, \downarrow, \rightsquigarrow, \leftrightarrow, \odot, \updownarrow, *\}$, $i = 1, 2, \ldots, n$. Further, Θ_i^{op} denotes operators vector, where in case of dropped superscript the i-th element of the original vector is preserved, while other elements are replaced by $*$: $\Theta_i = [*_1, *_2, \ldots, \theta_i, \ldots, *_n]^T$. In case of existing superscript the i-th element is replaced by the operator in the superscript: $\Theta_i^{op} = [*_1, *_2, \ldots, *_{i-1}, op, *_{i+1}, \ldots, *_n]^T$.

It is straightforward to see that by consecutively applying operator vectors $\Theta_i^{\odot} = [*_1, *_2, \ldots, \odot_i, \ldots, *_n]^T$, $i = 1, 2, \ldots, n$, an assessments matrix can be split into constituent matrices.

When assessments matrix has to be obtained from constituent matrices, merge operation is used:

Definition 8. Two non-overlapping matrices A^{κ_i} and B^{κ_j}, $\kappa_i \cap \kappa_j = \emptyset$, are processed into a new matrix $C^{\kappa_i \cup \kappa_j}$ through merge operation that is denoted by $\uplus$, i.e., $A^{\kappa_i} \uplus B^{\kappa_j} = C^{\kappa_i \cup \kappa_j}$. Elements in C are mappings from corresponding pairs $\alpha_{i,j}^A, \alpha_{i,j}^B \to \alpha_{i,j}^C$ according to the following rules:

(i) if $\alpha_{i,j}^A =$ " $-$ " and $\alpha_{i,j}^B =$ " $-$ " then $\alpha_{i,j}^C ::=$ " $-$ ";
(ii) if $\alpha_{i,j}^A \neq$ " $-$ " and $\alpha_{i,j}^B =$ " $-$ " then $\alpha_{i,j}^C ::= \alpha_{i,j}^A$;
(iii) if $\alpha_{i,j}^A =$ " $-$ " and $\alpha_{i,j}^B \neq$ " $-$ " then $\alpha_{i,j}^C ::= \alpha_{i,j}^B$.

Theorem 1. *Merge operation is commutative and associative.*

Proof. Let $\alpha_{i,j}^A = x$ and $\alpha_{i,j}^B = y$. If $x = y =$ " $-$ " then commutativity holds true according to (i). If $x \neq$ " $-$ " and $y =$ " $-$ ", which is (ii), then exchanging the operands results in (iii). If $x =$ " $-$ " and $y \neq$ " $-$ ", which is (iii) then exchanging the operands results in (ii). By definition, merge operation requires non-overlapping A and B, thus these are the only possibilities and commutativity is proved. As to associativity, let $A^{\kappa_i} \uplus (B^{\kappa_j} \uplus C^{\kappa_k})$. By definition, $D^{\kappa_j \cup \kappa_k} = B^{\kappa_j} \uplus C^{\kappa_k}$. Using the definition again, one gets $E^{\kappa_i \cup (\kappa_j \cup \kappa_k)} = A^{\kappa_i} \uplus D^{\kappa_j \cup \kappa_k}$. Now let $(A^{\kappa_i} \uplus B^{\kappa_j}) \uplus C^{\kappa_k}$. By using two times the definition of merge operation, one obtains matrix $H^{(\kappa_i \cup \kappa_j) \cup \kappa_k}$. Clearly, matrices E and H are equal: both have n rows with n elements, both have equal superscripts because of union being associative operation, which means that both have equal non-empty rows, i.e., $\kappa_i \cup \kappa_j \cup \kappa_k$, while empty rows in both cases contain only "$-$" elements. □

Ideally, agents use all other agents' assessments. But in many realistic scenarios this is not the case. Therefore dependency matrix has to be defined:

Definition 9. Dependency matrix Ξ of a society with n agents contains elements $\xi_{j,k} \in \{0, 1\}$, where $j, k = 1, 2, \ldots, n$. If $\xi_{j,k} = 1$ then entity j takes assessments of entity k into account when making trust calculations, otherwise not.

Dependency matrix is needed because despite existence of some assessments, certain entities may not be aware of them or may want to intentionally exclude them (note that $\xi_{j,k} = 1$ does not imply $\xi_{k,j} = 1$, because entity j may exclude k's assessment, while entity k may take j's assessment into account). Therefore where matrix Ξ contains at least one element that equals 0, the *dependecyExtraction* procedure, given in pseudo code below, has to be applied (note that the output of this procedure results in n partial assessment matrices, not constituency matrices):

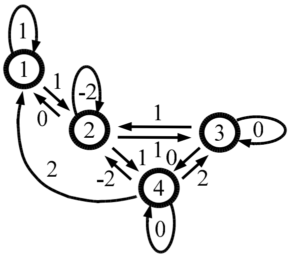
```
begin dependencyExtraction
input A^in, Ξ, n
output A_1^int, ..., A_n^int
  Ξ ::= Ξ^T
  for k ::= 1 to n
    for j ::= 1 to n
      for i ::= 1 to n
        if ξ_{i,k} == 1 then α_{i,j}^int ::= α_{i,j}^in
        else α_{i,j}^int ::= −
        i ::= i + 1
      endfor
    j ::= j + 1
    endfor, out A_k^int
  k ::= k + 1
  endfor
end
```

In the above procedure, superscript *int* denotes intermediate values, while superscript T denotes transposed matrix.

Definition 10. Operation dependency extraction is performed by procedure dependencyExtraction and is denoted by $\triangleleft$, i.e., $A^{in} \triangleleft \Xi$.

The computing of new assessment matrix procedure follows next (the notation [**1**] denotes a matrix with all elements being 1):

```
begin newAsessmentMatrixComputation
input A^in, Ξ, Θ, n
output A^out
  if Ξ ≠ [1] then call dependencyExtraction
  else
    for i = 1 to n
    A_i^int ::= A^in
    i ::= i + 1
    endfor
  endif
  for i = 1 to n
    A_i^int ◁◁ Θ_i
    i ::= i + 1
  endfor
  for i = 1 to n − 1
    A_{i+1}^int ::= A_i^int ⊎ A_{i+1}^int
    i ::= i + 1
  endfor
out A^out ::= A_n^int
end
```



$$A = \begin{bmatrix} 1 & 1 & - & - \\ 0 & -2 & 1 & 1 \\ - & 1 & 0 & 0 \\ 2 & -2 & 2 & 0 \end{bmatrix} \qquad \Xi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Fig. 2. An example society with assessment and dependency matrix.

DEFINITION 11. Assessment computing operation, denoted by infix operator $\triangleleft\triangleleft$, is done by applying operators vector $[\theta_1, \theta_2, \ldots, \theta_n]^T$ to assessment matrix A according to procedure *newAssessmentMatrixComputation*.

Operators vector dictates the processing of assessment matrix in a way that operator θ_i defines mappings of elements in i-th row, $\alpha_{i,j}$, $j = 1, 2, \ldots, n$. The general form of assessment processing operation is given below:

$$\begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \ldots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \ldots & \alpha_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,1} & \alpha_{n,2} & \ldots & \alpha_{n,n} \end{bmatrix} \triangleleft\triangleleft \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}.$$

Another example society is given in Fig. 2, where trust graph is presented together with corresponding trust matrix and dependency matrix. Further, entities take into account only assessments of topological neighbors, i.e., entity 1 has only one neighbor, 2, while entity 2 has there topological entities $(1, 3, 4)$, and entities 3 and 4 have two topological neighbors, 2 and 4, and 2 and 3, respectively.

Now let's assume that the entities in Fig. 2 are governed by the following operators vector $\Theta = [\Uparrow, \Downarrow, \rightsquigarrow, \Downarrow]$. Dependency extraction procedure is applied to the above assessment matrix first:

$$\begin{bmatrix} 1 & 1 & - & - \\ 0 & -2 & 1 & 1 \\ - & 1 & 0 & 0 \\ 2 & -2 & 2 & 0 \end{bmatrix} \triangleleft \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}^T.$$

The dependency extraction procedure gives four partial assessment matrices, given below on the left side of $\triangleleft\triangleleft$ operators. On their right side the corresponding operators vectors are given as determined by the second for loop in the *newAssessmentMatrixComputation* procedure (note again that dependency extraction has to be applied first in every simulation iteration if at least one element in Ξ equals zero – only afterwards operators can be applied):

$$\begin{bmatrix} 1 & 1 & - & - \\ 0 & -2 & 1 & 1 \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \lhd\lhd \begin{bmatrix} \Uparrow \\ * \\ * \\ * \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & - & - \\ 0 & -2 & 1 & 1 \\ - & 1 & 0 & 0 \\ 2 & -2 & 2 & 0 \end{bmatrix} \lhd\lhd \begin{bmatrix} * \\ \Downarrow \\ * \\ * \end{bmatrix},$$

$$\begin{bmatrix} - & - & - & - \\ 0 & -2 & 1 & 1 \\ - & 1 & 0 & 0 \\ 2 & -2 & 2 & 0 \end{bmatrix} \lhd\lhd \begin{bmatrix} * \\ * \\ \rightsquigarrow \\ * \end{bmatrix}, \quad \begin{bmatrix} - & - & - & - \\ 0 & -2 & 1 & 1 \\ - & 1 & 0 & 0 \\ 2 & -2 & 2 & 0 \end{bmatrix} \lhd\lhd \begin{bmatrix} * \\ * \\ * \\ \Downarrow \end{bmatrix}.$$

Now merge operation follows as required by the third for loop in the *newAssessmentMatrixComputation* procedure that gives the final result, the new assessment matrix:

$$A = \begin{bmatrix} 1 & 1 & - & - \\ 0 & -2 & 0 & 0 \\ - & -1 & 0 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix}.$$

Theorem 2. *Let the number of atomic entities in a society be denoted by n. Then trust assessment computation is computationally hard problem.*

Proof. When the number n of atomic entities grows linearly, the number of all entities that should be taken into account, including aggregate entities, grows exponentially – the number of all entities corresponds roughly to power set of n, which is 2^n. Assuming that obtaining an assessment from an entity takes constant time, the complexity for getting assessments from all entities, including aggregate entities, is $O(2^n)$. The next step is forming a matrix. By using some efficient sorting algorithm like Quick Sort this can be done in $O(n \log n)$ time on average if n atomic elements are sorted. In the case of trust, aggregate entities have to be taken into account, resulting in $O(2^n \log(2^n))$, which is $O(n2^n)$. Therefore before any simulation takes place, $O(n2^n)$ time on average is needed, thus trust assessment computation belongs to **EXP** class of problems. $\square$

4. Experiments and Discussion

This section gives a QAD application scenario to organizations management in the area of critical infrastructures (CIs) protection. Nowadays many CIs, most notably information and communications services, are being commercialized even for such applications like emergency services (in this segment Tetra technology was dominating in the past, but is

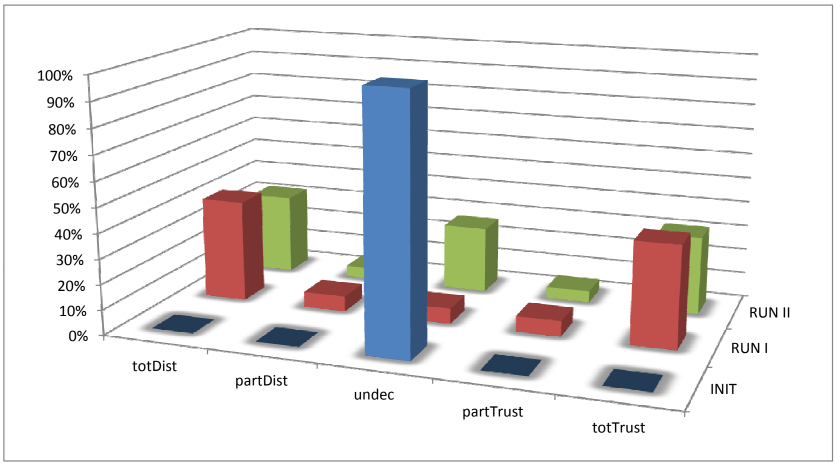


Fig. 3. Histograms of experiments I and II ("INIT" denotes initial distribution, "totDist" denotes "totally distrusted", "partDist" denotes "partially distrusted", "undec" denotes "undecided", "partTrust" denoted "partially trusted" and "totTrust" denotes "totally trusted").

being replaced by widely accepted commercial technologies like LTE). Such trend is not limited only to technology. Related services and other information services are actually outsourced and decreasingly run by states. This means that the system of commercial entities, services providing organizations, will compete on the market, and certain attitudes between them will develop through time.

Now the first vital question is: What is the most likely trust related dynamics within this system if information and communications providing entities are left on their own on the market? It is certainly in a state's interest that the more cooperative they are, the better, because in case of emergency, processes would run smoother, would be streamlined, etc. Many experiments prove that lack of trust leads to hampered information exchange, increased possibility of misunderstanding and, consequently, poor decision making (Koehn, 2003), which are clearly inter-organizational issues that affect CIs.

Therefore it is natural to assume that in case of urgent need imposed by the state, the CIs operated by these entities will perform better in case where more trust exist between them. So if distrust level reaches a certain degree, the state has to be warned. The next question is: If the state is supposed to intervene, can we identify the ways how to drive this system into more desirable states with higher degrees of trust?

Suppose the number of CI services providing entities is 100. Suppose further that initially they are undecided one about another, while 47% are extreme optimists, 47% are extreme pessimists, and 6% are assessment hoping. Let us simulate trust dynamics in this society by running 35 simulations, while each of these simulations takes 50 iterations (which roughly corresponds to the number of weeks in a year) and where in each simulation iteration (this means each week) 5% of entities randomly change their operators, choosing with equal probability between extreme optimists, extreme pessimists and assessment hoping. We obtain the histogram shown in Fig. 3 (see run I).

This result is not very satisfying. What can be done to make a more desirable outcome? Assuming that we can easier affect operators than initial values of operands (which likely

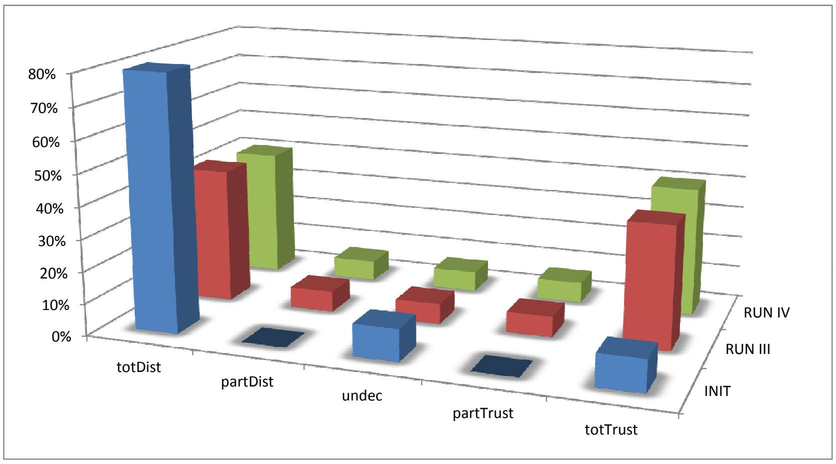


Fig. 4. Histograms of experiment III and IV ("INIT" denotes initial distribution, "totDist" denotes "totally distrusted", "partDist" denotes "partially distrusted", "undec" denotes "undecided", "partTrust" denoted "partially trusted" and "totTrust" denotes "totally trusted").

have less influence on the outcomes as operators), let us try to find the strategy to manage the system towards more desirable states. Suppose now that initially 40% of entities are extreme optimists, 40% extreme pessimists, 10% assessment hoping, and 10% self-confident. Again, let in each iteration 5% of entities choose a new operator, this time randomly with equal probability among extreme optimistic, extreme pessimistic, centralistic, assessment hoping and self confident operator. Running again this society for 35 runs, where each has 50 iterations, the histogram in Fig. 3 (see run II) is obtained.

What is surprising already in these two simulations is a fact that in a population, where initially nobody cared about one another ended up in society, "extremists wings" appeared, both positive and negative. From these simulations we may assume that assessment hoping operator was the main cause behind these changes.

Clearly, this is not some kind of a favorable state, where in the ideal case 100% of assessments would be totally trusted. On the other hand, the worst case would be where all assessments would be totally distrusted. Now based on the lesson learned above, can we expect that such high degree of distrust can be overcome somehow by, i.e., messing with entities opinions (choosing these entities randomly) so that these entities randomly change their operator each week? Put another way – would it be fair, if the state promises that after one year the level of trust in a society will be increased, despite the fact that it will just mess randomly with chosen entities?

To check this let us assume that the observed society is in a state where entities are mainly un-cooperative with a high degree of distrust. Putting this into numbers would mean, e.g., 80% of assessments being totally distrusted, 10% undecided, and 10% totally trusted. As to operators, 80% of entities are governed by extreme pessimistic operator, while only 10% are governed by extreme optimistic operator and 10% by assessment hoping. Now let us run again simulation for 50 iterations, 35 times, where in the first case only 5% of randomly chosen agents are randomly changing their operators in each iteration by choosing with equal probability among these same three operators (see run III in Fig. 4), while in the last run these percentage of assessment hoping operators entities is increased to 30%, and all other conditions remain unchanged (see run IV in Fig. 4).

These two last simulations show that such intentional “messing with entities” would “dilute” high degree of distrust to much more favorable outcome as shown in Fig. 4. Clearly, lessons learned in the first case paid off when managing organizations in the second case.

It should be added that the elements in the dependency matrix in the above simulations were set to 1, while 50 iterations for simulations were sufficient – at this point the histograms were getting their stable shapes and the processes were close to their convergence points.

Finally, two additional issues have to be discussed. The first is about QAD implementation. We have implemented trust management solution as a standalone simulator and as a SOA service. In the latter case the architecture consists of a distributed database where trust values (matrices) are stored, and of a user interface (for accessing this database) that inserts and retrieves values. More precisely, the distributed database is implemented using SOA standards, while user interface is linked to it through SOAP protocol and it deploys two primitives: *trustQuery* and *trustReply*, which are both defined with XML schema (more details on implementation can be found in Kovač and Trček (2010)). The second issue is about comparing performance of QAD to other trust management methods. For this purpose we have developed a simulating environment, a novel test-bed that directly measures the outputs of trust models and uses two metrics for this purpose, accuracy and coverage, to quantitatively assess their performance (Jelenc *et al.*, 2012). Interestingly, QAD performs very well with two of its operators: $\uparrow$ and $\downarrow$. But one should note that this not renders other operators useless – on the contrary, they just reflect humans reasoning, and this is often not perfect.

5. Conclusions

Trust related issues in e-environments spawned computational trust management research some fifteen years ago, also as a consequence of the fact that e-environments do not provide such feedback like ordinary communications channels. Many trust management methods have been proposed so far and some of them are already crossing the border between research and (commercial) applications. However, e-environments are about supporting human users, while the methods developed so far are based on advanced mathematics and hardly understood by ordinary users. Qualitative Assessment Dynamics, QAD, which is presented in this paper, is aligned with some typical human reasoning principles when it comes to trust. Therefore it does not replace the afore mentioned methods, but it complements them. QAD operands and operators are common to human reasoning processes and have background in language descriptions. They are meaningful and understandable in various cultural settings. In addition, QAD is a formal system that enables hard formal treatment and is suitable for direct implementation in computing environments.

These properties enable applications not only for on-line support like e-commerce, but also for a more general decision support issues. Namely, QAD can be easily implemented in multi-agent systems, where appropriate distribution of operators can be chosen together with appropriate distribution of initial trust values. Afterward, the dynamics of trust in this community can be studied and outcomes obtained together with a possibility of studying modifications to these settings to achieve desirable outcomes in terms of trust in the society, or at least to avoid unwanted outcomes. This was demonstrated by an experiment applied to the area of organizations management. This experiment also suggests how research in mathematics and computer sciences area can be applied to other fields, in particular human resources and organizations management.

Future work will address inclusion of mechanisms that could improve the accuracy of collected assessments through, e.g., an analysis of large-scale data collected from heterogeneous resources (ranging from web data to sensors data) and that are related to these assessments (a comprehensive survey of implementable methods is given in Dzemyda and Sakalauskas (2011)). Future work will also focus on further refinement of QAD to even closer reflect human reasoning. One such case is relaxation of the requirement that if an assessment, which is currently being a subject of calculation, is not known or disclosed, that it remains disclosed also after the operation. Clearly, in reality an entity may at some point decide to disclose its assessment(s), and thus it becomes cooperative. In addition, new operators are expected to be added in the future.

Acknowledgments. Author acknowledges the support of the Slovenian Research Agency ARRS through program P2-0359. In addition, this research is also a partial result of collaboration in COST Agreement Technologies project and COST IS1104. Author wants to express thanks also to Mrs. E. Zupančič for programming the QAD simulation environment. Last but not least, our thanks go to anonymous reviewers that have helped to improve the paper.

References

Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F. (2006). A survey of autonomic communications. *ACM Transactions on Autonomous and Adaptive Systems*, 1(2), 223–259.

Dzemyda G., Sakalauskas L., (2011). Large-scale data analysis using heuristic methods. *Informatica*, 22(1), 1–10.

Fonte, A., Curado, M., Monteiro, E. (2008). Interdomain quality of service routing: setting the grounds for the way ahead. *Annals of Telecommunications*, 63(11), 683–695.

Jelenc, D., Hermoso, R., Ossowski, S., Trček, D. (2012). Alpha test-bed: a new approach for evaluating trust models. In: *Proceedings of ITMAS 2012, The Third International Workshop on Infrastructures and Tools for MAS*, Valencia, pp. 51–64.

Josang, A. (2001). A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3), 279–311.

Kahneman, D., Tversky, A. (1979). Prospect theory: an analysis of decision under risk. *Econometrica*, XLVII, 263–291.

Koehn, D. (2003). The nature of and conditions for on-line trust. *Journal of Business Ethics*, 43, 3–19.

Kovač, D., Trček, D. (2010). Qualitative trust modeling in SOA. *Journal of Systems Architectures*, 55(4), 255–263.

Raza, M., Khader, H.F., Khadeer, H.O. (2011). Neural network-based approach for predicting trust values based on non-uniform input in mobile applications. *Computer Journal*. doi:10.1093/comjnl/bxr104 (on-line publication), Oxford University Press.

Rettinger et al., (2011). Statistical relational learning of trust. *Machine Learning*, 82(2), 191–209.

Ross, D. (2010). Game theory. In: Zalta, E.N. (ed.), *The Stanford Encyclopedia of Philosophy*, `http://plato.stanford.edu/archives/fall2011/entries/game-theory/`.

Sabater, J., Sierra, C., (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1), 33–60.

Schonwalder, J., Fouquet, M., Rodosek, G., Hochstatter, I. (2009). Future Internet = content + services + management. *Communications Magazine*, 47(7), 27–33.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton.

Tennenholtz, M. (2008). Game-theoretic recommendations: some progress in an uphill battle. In: *Proceedings of AAMAS 2008*, Estoril, pp. 10–16.

Trček, D. (2009). A formal apparatus for modeling trust in computing environments. *Mathematical and Computer Modeling*, 49(1/2), 226–233.

Trček, D. (2011). Trust management methods for the web. In: *Lecture Notes in Computer Science*, Vol. 6848(X), pp. 445–459. Springer, Berlin.

Wang, Y., Vassileva, J., (2003). Trust and reputation model in peer-to-peer networks. In: *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P'03)*. Linkoping, 150 pp.

Zupančič E., Trček D., (2012). Deployment of trust management system in environment of e-commerce. In: *Lecture Notes in Computer Science*, vol. 7327, pp. 484–493.

D. Trček is with Faculty of Computer and Information Science, University of Ljubljana, where he heads Laboratory of e-media. He has been involved in the field of computer networks and IS security and privacy for over twenty years. He has taken part in various EU and national projects in government, banking and insurance sectors (projects under his supervision totaled to approx. one million EUR). His bibliography includes over one hundred titles, including monograph published by renowned publisher Springer. D. Trek has served (and still serves) as a member of various international bodies and boards (MB of the European Network and Information Security Agency, etc.). His interests include security, trust management, privacy and human factor modeling.

Skaičiuojamasis pasitikėjimo valdymas, QAD ir taikymai

Denis TRČEK

Pasitikėjimo faktorius yra svarbus sėkmingam elektroninės komercijos plėtojimui bei panašiose elektroninėse terpėse. Straipsnyje pateikiama skaičiuojamojo pasitikėjimo valdymo metodų apžvalga. Metodai yra taikomi ne tik elektroninėse aplinkose, bet ir sprendimų priėmime, taikant matematinį ir imitacinį modeliavimą. Šiems metodams tobulinti yra siūlomas kokybinio vertinimo dinamikos (QAD) metodas, kuris formalizuoja žmogiškąjį faktorių ir jo samprotavimus. Pateikiama išsami QAD operatorių notacija. Naudojant straipsnyje pateikiamą metodologiją yra pristatomas organizacijos valdymo eksperimentas.