A Methodology for Trust Control Mode Prediction

ZHENG YAN
Nokia Research Center
Itämerenkatu 11-13, 00180 Helsinki, Finland

Abstract: - Trust has been recognized as an important factor for a component software platform. Inside the platform, trust can be controlled according to its evaluation result. Special control modes can be applied into the software platform in order to ensure a trustworthy system. In this paper, we present a methodology for trust control mode prediction and selection in order to support autonomic platform trust management. The methodology is based on Fuzzy Cognitive Maps. The simulation results show this method is effective for predicting and selecting the feasible trust control modes.

Key-Words: - Trusted computing, Trust modeling, Trust management, Security, Dependability

1 Introduction

The growing importance of software in the domain of mobile systems introduces special requirements on trust due to the nature of applications they provide. In particular when the software is component based and varies due to components joining and leaving the system. However, the lack of a trusted software platform could be the main reason that retards the further development of mobile applications and services.

From a system point of view, trust is the assessment of trustor on how well the observed behavior (quality attributes) of trustee meets trustor's own standards for an intended purpose [1]. From this, the critical characteristics of trust can be summarized, it is: subjective, different for each individual in a certain situation; and dynamic, sensitive to change due to the influence of many factors. Therefore, we need a proper mechanism to support autonomic trust management not only on trust establishment, but also on trust sustaining.

A number of trusted computing and management work have been conducted in the literature and industry. For example, TCG (Trusted Computing Group) aims to build up a trusted computing device on the basis of a secure hardware chip [2-6]. Some of trust management systems focus on protocols for establishing trust in a particular context, generally related to security requirements. Others make use of a trust policy language to allow the trustor to specify the criteria for a trustee to be considered trustworthy [7]. However, the focus of the security aspect of trust tends to assume that the other non-functional requirements [8], such as availability and reliability, have already been addressed.

Recently, many mechanisms and methodologies are developed for supporting trusted communications and collaborations among computing nodes in a distributed system (e.g. Ad Hoc Networks, P2P systems and GRID computing systems) [9-12]. These methodologies are based on digital modeling of trust for trust evaluation and management. We found that these methods are not very feasible for supporting the trust of a device software platform.

Regarding software engineering, trust has been recognized as an important factor for the component software platform. A couple of interesting models have been proposed to ensure the quality of component services at runtime and protect the users [13-15]. However, we found that the trust model proposed in [13, 14] mainly focuses on the runtime component configuration support, while the model in [15] aims to prevent that a component user sends wrong reports resulting in a bad trust value of the component, especially at component download time. We argue that trust can be controlled according to its evaluation result. Special control modes can be applied into the software platform in order to ensure a trustworthy system.

The work presented in this paper is conducted in EU ITEA Trust4All project. This project aims to build up trustworthy middleware architecture to support easy and late integration of software from multiple suppliers and still have dependable and secure operation of the resulting system. This paper presents our continuous work of [16]. The focus is to propose a methodology for trust control mode prediction and selection targeting to support autonomic trust management.

The rest of the paper is organized as follows. Section 2 specifies the requirements of platform trust management. Section 3 presents the trust control model and an algorithm used for the trust control mode prediction. Section 4 reports our simulation results. Section 5 discusses implementation strategies for deploying this methodology into the Trust4All platform. Finally, conclusions and future work are presented in Section 6.

2 Platform Trust Management

As defined in [7], trust management is concerned with: collecting the information required to make a trust relationship decision; evaluating the criteria related to the trust relationship as well as monitoring and reevaluating existing trust relationships; and automating the process. We think extension is needed in order to provide software platform trust. We proposed in [16] that autonomic trust management includes four aspects:

- Trust establishment: the process for establishing a trust relationship between a trustor and a trustee.

- Trust monitoring: the trustor or its delegate monitors the performance of the trustee. The monitoring process aims to collect useful evidence for the trust assessment.

- Trust assessment: the process for evaluating the trustworthiness of the trustee by the trustor or its delegate. The trustor assesses the current trust relationship and decides if this relationship is changed. If it is changed, the trustor will make decision which measure should be taken.

- Trust control and re-establishment: if the trust relationship will be broken or is broken, the trustor will find reasons and take corresponding measures to control or re-establish the trust.

A number of requirements can be summarized in order to support autonomic platform trust management. Firstly, for urgent and trust priority high service request, the device should handle it adaptively. This can be solved by system architecture design to support the collaboration between trust management framework and resource management framework through component trust modeling. Secondly, for trust crash, the device should react adaptively as expected within some limited time. Trust evaluation based detection and assessment on selected control modes can be applied to solve this issue. Finally, the platform should be intelligent for trust management. 'Which trust control mechanism is good for improving which quality attributes in what kind of context' should be well addressed. The trust control modes should be predicted for selection and deployment.

In [16], we proposed autonomic trust management architecture for component software platform and applied Subjective Logic (SL) to implement trust evaluation and control mode assessment in order to fulfill the first two requirements. Herein, we present a methodology for trust control mode prediction and selection based on a Fuzzy Cognitive Map in order to address the third requirement [17].

3 Trust Control Mode Prediction

3.1 Factors related to platform trust

The component software platform is composed of a number of entities, e.g. a component (composition of components), an application, a sub-system and the whole platform system. The trustworthiness of a platform entity depends on a number of quality attributes of this entity. The quality attributes can be the entity properties and recommendations or reputations regarding this entity. The decision or evaluation of trust is conducted based on the trustor (e.g. a platform user or his/her delegate)'s subjective policy and the entity (trustee)'s quality attributes, as well as influenced by the context. The context specifies any information that can be used to characterize the situation of the trustee entity. The quality attributes of the platform entities can be controlled or improved by applying a number of trust control mechanisms. The relationships of those factors related to platform trust are illustrated in Figure 1.

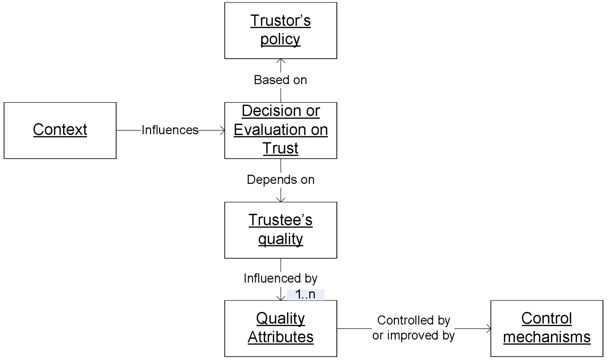

Figure 1: Factors influencing the decision or evaluation of trust

3.2 Trust control modeling

Considering a platform entity's trustworthiness, it is influenced by a number of quality attributes $QA_i (i = 1,...,n)$. These quality attributes are ensured or controlled through a number of control modes supported by the platform system $C_j (j = 1,...,m)$. A control mode contains a number of control mechanism or operations, e.g. encryption, duplication of process, man-in-middle solution, etc. It can be treated as a special configuration of trust management that can be provided by the system.

The trustworthiness value can be described as:

$$T = f\left(\sum_{i=1}^{n} w_i V_{QA_i} + T^{old}\right),$$

where $\sum_{i=1}^{n} w_i = 1$. w_i is a weight that indicates the importance rate of the quality attribute QA_i regarding how much this quality attribute is considered at the trust decision or evaluation. w_i can be decided based on the trustor's policy. The value of the quality attribute is denoted by V_{QA_i}. It can be calculated according to the following formula:

$$V_{QA_i} = f\left(\sum_{j=1}^{m} cw_{ji} V_{C_j} B_{C_j} + V_{QA_j}^{old}\right),$$

where cw_{ji} is influence factor of control mode C_j to QA_i, cw_{ji} is set based on impact of C_j to QA_i. Positive cw_{ji} means positive influence of C_j on QA_i. Negative cw_{ji} implies negative influence of C_j on QA_i. B_{Cj} are the selection factors of the control mode C_j, which can be either 1 if C_j is applied or 0 if C_j is not applied.

The value of the control mode can be calculated using $V_{C_j} = f\left(T \cdot B_{C_j} + V_{C_j}^{old}\right)$, where B_{C_j} reflects the current system configurations about which control modes are applied. Note that $V_{QA_i}, V_{C_j}, T \in [0,1]$, $w_i \in [0,1]$, and $cw_{ji} \in [-1,1]$. We apply sigmoid function as threshold function *f:* $f(x) = \frac{1}{1+e^{-\alpha x}}$ ($\alpha = \frac{1}{2}$ or $\alpha = \frac{1}{3}$), to map V_{QA_i}, V_{C_j}, T into [0, 1]. In addition, $\Delta T = T - T^{old}$ stands for the change of trustworthiness value. T^{old}, $V_{QA_i}^{old}$ and $V_{C_j}^{old}$ are old value of T, V_{QA_i}, and V_{C_j}, respectively.

The above modeling can also be described as a graphical illustration FCM, as shown in Figure 2. It is a signed directed graph with feedback, consisting of nodes and weighted arcs. Nodes of the graph stand for the concepts in the system and they are connected by signed and weighted arcs representing the causal relationships that exist between the concepts. In Figure 2, there are three-layer concept nodes in the graph. The node in the top layer is the trustworthiness of the platform entity. The nodes located in the middle layer are the quality attributes of the entity, which has direct influence on the entity's trustworthiness. The nodes at the bottom layer are control modes that could be supported and applied inside the system. These control modes can control and thus improve the quality attributes. Therefore, they have indirect influence on the trustworthiness of the entity.

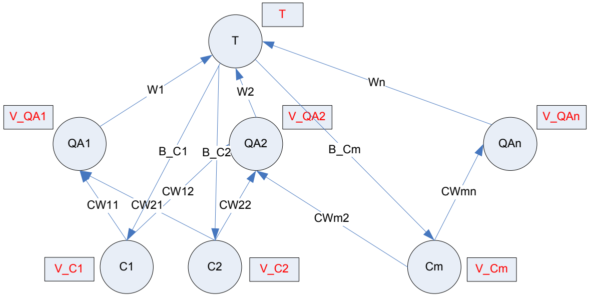

Figure 2: Graphical modeling of trust control

3.3 Algorithm

The control modes are predicted through evaluating all possible modes and their compositions based on the following algorithm.

```
- For every composition of control modes
$\forall S_k (k = 1,...,K)$, While $\Delta T = T - T^{old} < \delta$, do
    $V_{C_j} = f\left(T \cdot B_{C_j} + V_{C_j}^{old}\right)$
    $V_{QA_i} = f\left(\sum_{j=1}^{m} cw_{ji} V_{C_j} B_{C_j} + V_{QA_j}^{old}\right)$
    $T = f\left(\sum_{i=1}^{n} w_i V_{QA_i} + T^{old}\right)$
- Compare $V_{QA_i}$ and $T$ for different
control mode compositions, select a
composition whose $V_{QA_i}$ and $T$ pass the
threshold $tr$.
```

Herein, threshold (tr) is the average of trust value T_k of $S_k (k = 1,...,K)$, i.e. $tr = \sum_{k=1}^{K} T_k$. $S_k (k = 1,...,K)$ can be expressed by the control mode selection factors B_{C_j}, which represents which control mode is selected and applied in the system. δ is the accepted change of the trustworthiness value.

4 Simulation

Our simulation is based on a practical example, as shown in Figure 3. The trustworthiness of the platform is influenced by three quality attributes: QA_1 - Security; QA_2 - Availability; QA_3 - Reliability, with important rates $w_1 = 0.4$, $w_2 = 0.3$, and $w_3 = 0.3$, respectively. There are three control modes that could be provided by the system:

- C_1: security mode 1 with strong encryption, but medium negative influence on availability.
- C_2: security mode 2 with light encryption and light negative influence on availability
- C_3: fault management mode with positive improvement on availability and reliability.

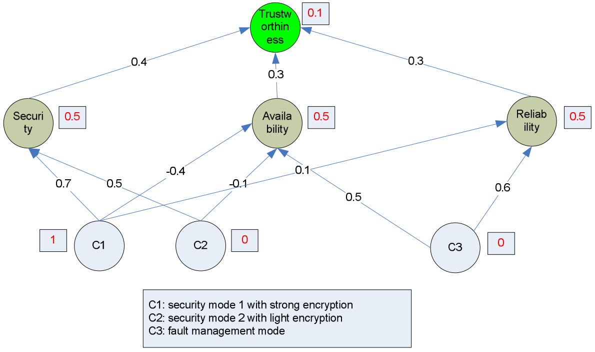
Trustw orthin ess
0.1
0.4
0.3
0.3
Securi ty
0.5
Availa bility
0.5
Reliab ility
0.5
0.7
0.5
-0.4
-0.1
0.1
0.5
0.6
1
C1
C2
0
C3
0
C1: security mode 1 with strong encryption
C2: security mode 2 with light encryption
C3: fault management mode

Figure 3: FCM of simulation

The influence of each control mode to the quality attributes is specified by the arc weights. The values in the square boxes are initial values of the concept nodes. In practice, the initial value can be set as asserted one or expected one, which can be specified in the system trust policy profile.

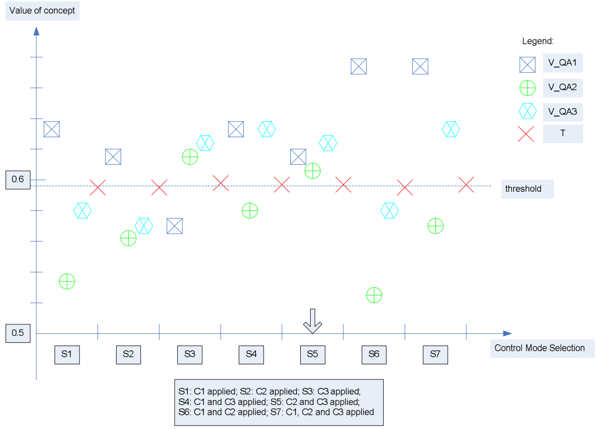

Figure 4: Simulation results ($\alpha = \frac{1}{2}$ and $\delta = 0.0001$)

The simulation results are shown in Figure 4. In this case, there are seven control mode compositions.

- S_1: C_1 is applied, i.e. $B_{C_1} = 1; B_{C_2} = 0; B_{C_3} = 0$;
- S_2: C_2 is applied, i.e. $B_{C_1} = 0; B_{C_2} = 1; B_{C_3} = 0$;
- S_3: C_3 is applied, i.e. $B_{C_1} = 0; B_{C_2} = 0; B_{C_3} = 1$;
- S_4: C_1 and C_3 are applied, i.e. $B_{C_1} = 1; B_{C_2} = 0; B_{C_3} = 1$;
- S_5: C_2 and C_3 are applied, i.e. $B_{C_1} = 0; B_{C_2} = 1; B_{C_3} = 1$;
- S_6: C_1 and C_2 are applied, i.e. $B_{C_1} = 1; B_{C_2} = 1; B_{C_3} = 0$;
- S_7: C_1, C_2 and C_3 are applied, i.e. $B_{C_1} = 1; B_{C_2} = 1; B_{C_3} = 1$;

We can see that S_5 is the best choice since both the quality attribute values and the trustworthiness value are above the threshold.

5 Implementation Strategies

Special issues should be considered for the methodology implementation on the Trust4All platform [18]. We outline some strategies in this section.

5.1 Resource consideration

For some devices with limited resources, we should add additional checking steps in the implementation regarding resource management. Two checks are needed. One is conducted before running the prediction functions in order to find all possibly supported control modes. The other check is needed after the prediction in order to ensure the resources required by the selected control modes can be satisfied by the system. If not, we need to select second best solution. In our simulation case, it could be S_4. Otherwise, the system will raise warning.

5.2 Model optimization

The model for trust control mode prediction can be dynamically maintained and optimized in order to make it context-aware. In Trust4All platform, we apply observation based trust evaluation, which can play the basis for the control mode assessment [16]. If trust value is below threshold, increase the negative point of applied control modes; else, increase the positive point of applied control modes. We further calculate the trust value of applied control modes. If the trust value of applied control mode is below threshold, switch it off.

In addition, the arc weights cw_{ji} can be further adjusted based on the following scheme in order to make it match real context. Here, we use $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$ to stand for V_{QA_i} generated based on real system monitoring and by prediction, respectively. ω is a unit deduction factor and suppose C_j with cw_{ji} is currently applied in the system. σ is the accepted error between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$.

```
While $|V_{QA_i}\_monitor - V_{QA_i}\_predict| > \sigma$ ,
do
```

```
      1.1 If $V_{QA_i}\_monitor < V_{QA_i}\_predict$,
          $cw_{ji} = cw_{ji} - \omega$; Else,
          $cw_{ji} = cw_{ji} + \omega$
      1.2 Run the control mode
          prediction function and get a
          new selection S
  2 If $S \neq S_{old}$, apply S and go to 1.
  3 If S is not existed, raise warning.
```

5.3 Implementation

The function of the trust control mode prediction has been designed as a number of system components in the Trust4All middleware platform [18]. The model used for the trust control mode prediction and selection is described by XML files. These XML files can be dynamically maintained according to the real system context. For example, new control modes can be added and ineffective ones can be removed. The parameters of the model (e.g. cw_{ji}) can be adjusted based on the real system behavior. We initialize the trust control mode prediction via parsing the above XML files. The following interfaces are designed for the Trust4All implementation.

- ControlGraph **Initialize_ControlGraph** (File* controlmodedescription);
- PredictionResult **ControlMode_Prediction** (ControlGraph cgraph);
- Int[] **ControlMode_Selection** (PredictionResult presult);
- ControlModeEvaluationResult **ControlModeEvaluation** (Opinion trustvalue, Opinion[] QAValues, int[] appliedcontrolmode, Context currentcontext);
- ControlGraph **ControlMode_Weights_Change** (ControlModeEvaluationResult cmevaluateresult, PredictionResult presult);
- **Save_ControlGraph**(ControlGraph cGraph, Context ct, File* controlmodedescription);

6 Conclusions

This paper proposed a methodology for the trust control mode prediction and selection aiming at autonomic platform trust management. We made use of the Fuzzy Cognitive Maps to model the relationships among the trust control modes, the quality attributes of the platform entity and its trustworthiness. Based on this model, we proposed an algorithm to conduct the control mode predication and selection. The simulation result shows this method is effective for predicting and selecting the suitable trust control modes for the system. Furthermore, we discussed the strategies for the methodology implementation on the Trust4All middleware platform.

For future work, we will further prove and optimize our algorithms based on the implementation, especially the context-aware adjustment of the trust control model at the system runtime.

Acknowledgments

This work is partially sponsored by European Union ITEA project Trust4All. The authors would like to thank Trust4All project members and Nokia internal reviewers for valuable discussions and comments.

References:

[1] DE Denning, A New Paradigm for Trusted Systems, *Proceedings of the IEEE New Paradigms Workshop*, 1993.

[2] TCG TPM Specification v1.2, 2003. https://www.trustedcomputinggroup.org/specs/TPM/

[3] Vaughan-Nichols, S.J. How trustworthy is trusted computing? *Computer*, Volume 36, Issue 3, March 2003.

[4] Davis P.T. TCPA: who can you trust. EDPACS: the EDP Audit, *Control and Security Newsletter*, Dec. 2002.

[5] England, P.; Lampson, B.; Manferdelli, J.; Peinado, M.; Willman, B. A Trusted Open Platform, *IEEE Computer Society*, p55-62, July 2003.

[6] Baldwin, A.; Shiu, S. Hardware Security Appliances for Trust. *In Proceedings of the First International Conference of Trust Management* (iTrust 2003), Crete, Greece, May 2003.

[7] T. Grandison, M. Sloman, A Survey of Trust in Internet Applications, *IEEE Communications and Survey*, Forth Quarter, 3(4), pp. 2–16, 2000.

[8] S. Banerjee, Chris A. Mattmann, Nenad Medvidovic, Leana Golubchik, Leveraging Architectural Models to Inject Trust into Software Systems, ACM SIGSOFT Software Engineering Notes , *Proceedings of the 2005 workshop on software engineering for secure systems—building trustworthy applications SESS '05*, Volume 30 Issue 4.

[9] Z. Zhang, X. Wang, Y. Wang, A P2P Global Trust Model Based on Recommendation, *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, Vol. 7, Aug. 2005, pp.3975 – 3980.

[10] G. Theodorakopoulos, J.S. Baras, On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks, *IEEE Journal on Selected Areas in*

Communications, Vol. 24, Issue 2, Feb. 2006, pp. 318 – 328.

[11] C. Lin, V. Varadharajan, Y. Wang, V. Pruthi, Enhancing Grid Security with Trust Management**,** *Proceedings of IEEE International Conference on Services Computing* (SCC 2004), Sept. 2004, pp. 303 – 310.

[12] Y. Sun, W. Yu, Z. Han, K.J.R. Liu, Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks, *IEEE Journal on Selected Area in Communications*, Vol. 24, Issue 2, Feb. 2006, pp. 305 – 317.

[13] M. Zhou, H. Mei, L. Zhang, A Multi-Property Trust Model for Reconfiguring Component Software, *the Fifth International Conference on Quality Software QAIC2005*, 19-20 Sept. 2005, pp. 142 – 149.

[14] M. Zhou, W. Jiao, H. Mei, Customizable Framework for Managing Trusted Components Deployed on Middleware, *In Proceedings of 10th IEEE International Conference Engineering of Complex Computer Systems ICECCS 2005*, 16-20 June 2005, pp. 283 – 291.

[15] P. Herrmann, Trust-Based Protection of Software Component Users and Designers. *In Proceedings of the First International Conference of Trust Management (iTrust 2003)*, Crete, Greece, May 2003.

[16] Zheng Yan, Ronan MacLaverty, "Autonomic Trust Management in a Component Based Software System", *the Third International Conference on Autonomic and Trusted Computing (ATC'06)*, China, Sept. 2006.

[17] Kosko, B. Fuzzy Cognitive Maps. *International Journal Man-Machine Studies*, vol.24, pp.65-75, 1986.

[18] Robocop, Space4U and Trust4All website: https://nlsvr2.ehv.campus.philips.com/