A Role-based Trust Model assessing IoA services: First Results on Real MAS Implementation by using ROS 2

Runbo Su [*]
LORIA, CNRS, Université de Lorraine
runbo.su@loria.fr

Amaury Saint-Jore [*]
LORIA, CNRS, Université de Lorraine
amaury.saint-jore@loria.fr

Abstract

IoA (Internet of Agents) [13], combining MAS (Multi-Agent Systems) and IoT (Internet of Things), aims to provide an environment where diverse services can be performed and evaluated through heterogeneous agents' cooperative participation. Given this, IoA requires a mechanism monitoring agents' real-time behaviors to prevent undesired activities caused by compromised ones. This paper proposed a trust model to detect agents' misbehavior by taking their roles into consideration, namely SP (Service Provider), SR (Service Rater), and SR+SP. Moreover, we conducted a real MAS implementation by using ROS 2 (Robot Operating System) to verify the preliminary results of SR and SP agents' trust evaluation.

Categories and Subject Descriptors

D.4.5 [**Operating Systems**]: Reliability

General Terms

Measurement, Performance

Keywords

Trust, Internet of Agents, Real Implementation, ROS 2

1 Introduction and Motivation

MAS refers to systems composed of agents interacting with each other and performing actions to achieve a dedicated goal [12]. Since MAS can solve complex problems that remain demanding for a single agent, MAS has been applied to diverse fields, e.g., robotic controls and smart grids [2]. And recently, a trend of solutions is emerging that combine MAS and IoT (Internet of Things), which is called MaIoT (Multi-agent IoT) [6], MAS-based IoT [9], or IoA. On the other hand, as identified in [2], while a lot of features support MAS' adoption into IoT, such as resource utilization and reconfigurability, the real-time behavior, like service-related activities, is still insufficiently addressed. For this, some studies applied trust models to aid service assessment through a mathematical framework, which describes the real-time reliability of an agent by a probabilistic value in the range of [0, 1]. To date, numerous trust models have been proposed and developed to improve IoA applications: an Integrated Trust Establishment model (ITE) was proposed in [1], which utilizes a multi-criteria method to build a comprehensive assessment of a trustor's needs in IoA. A multi-agent subjectivity alignment (MASA) mechanism was designed in [14] to counteract biases of different agents and increase the accuracy of second-hand information fusion by using a regression technique. However, the evaluation of agents during the service process, i.e., agents as service providers or consumers, is not discussed in both [1, 14] of them. In this regard, authors in [7] introduced a centralized trust model in which Quality of Recommendation is collected to measure agents' trust accordingly. To integrate an overall monitoring scheme for both SP and SR, authors in [10] proposed a phase-based trust model considering both qualities of service provider and consumer (named 'rater' in this work) and output an overall trust score based on these two quality values. Furthermore, all the above-mentioned works are validated by simulation results, meaning that the real implementation is still missing. Indeed, according to [8], none of the surveyed trust models conducted implementation. From the above review, we can notice that agents' trust as service providers and consumers is rarely addressed, and the overall trust considering both sides is often insufficiently discussed. And more importantly, an implementation with real-world IoT devices is needed to validate the effectiveness and feasibility.

With the purpose of overcoming these limitations, we present in this paper a role-based trust model to evaluate the agents by their service-related activities, i.e., service provision and service rating. This work is based on our previous work [10], where the inter-agent communication and service process remain distributed, and the trust evaluation will be realized by a centralized trust manager. Furthermore, we implemented this model with a real MAS composed of aerial and ground robots to validate the preliminary results by using ROS 2. The rest of the paper is organized as follows: Section 2 defines the role-based agents and details their trust assessment. The implementation setup will be detailed in Section 3. The implementation results and performance analysis are presented in Section 4. Lastly, Section 5 draws the conclu-

[*] Contributed equally.

sion and outlines our future work.

2 Framework: Trust of Role-based Agents

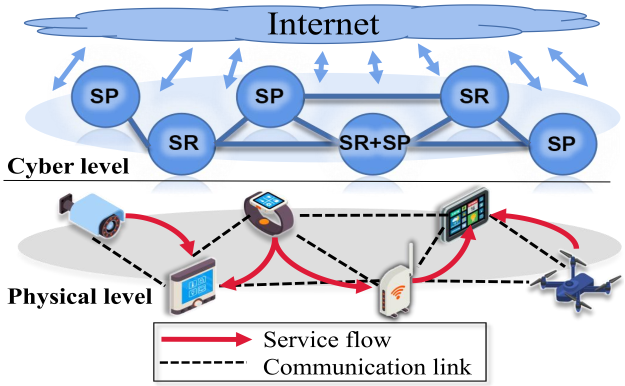

Figure 1. A motivating example of a service process in IoA, where agents communicate with each other and contribute to the service cooperatively according to their roles, i.e., SR, SP, or SR+SP.

As illustrated in Fig. 1, agents may provide or consume service, or both, depending on their capabilities and resource situation, as well as the service requirement. Thus, we define in this section role-based agents and introduce three algorithms evaluating their trustworthiness accordingly. We fix the timestamp for the current and last service t and $t-1$, a SR agent i, SP agent j, and SR+SP agent k until the end of the section.

-SR agent: To give feedback reporting the service quality, service consumers are supposed to rate the services they receive, and in this case, they become SR in service evaluation. As dishonesty-biased feedback either damages well-behaved SP agents' trustworthiness or increases misbehaving ones' reputation, *ToR* given by Algorithm 1 can be used to identify the dishonest service raters by punishing the gap between the SR agents' feedback and the average level.

Algorithm 1 Assessing the SR Agent: **T**rust **o**f service **R**ater agent (*ToR*)

```
    Input: $ToR_i^{t-1}$, $f^t$        Output: $ToR_i^t$
    Parameters: $\lambda \in [0.5, 1[$

        $ToR_i^t = \lambda \cdot QoR_i^t + (1-\lambda) \cdot ToR_i^{t-1}$,        (1)
where Quality of Rater
        $QoR_i^t = 1 - \frac{1}{|SP^t|} \sum_{j \in SP} |f_{ij}^t - \bar{f}_j^t|^{1/2}$
    ▷ $SP^t$ refers to all SP agents at time $t$ and $\bar{f}_j^t$ is the average of $j$'s notes at time $t$
```

-SP agent: Services performed by SP agents should also be counted in service evaluation, not only the quality of their current services but also the stability. *ToP* value computed by Algorithm 2 not only inspects feedback of the current service provided by the SP agents but also verifies their stability of giving service. In ρ calculation, the normalized sinc function is chosen since it is continuous at point 0, maps $[0,1]$ onto $[0,1]$, and has inflections that can penalize the great Δf and unsatisfactory f. In such a manner, the unstable behaviors over time will be punished by ρ, and the only opportunity for SP agents to gain reputation is to provide high-quality services in a consistent manner.

Algorithm 2 Assessing the SP Agent: **T**rust **o**f service **P**rovider agent (*ToP*)

```
    Input: $ToP_j^{t-1}$, $f^t$        Output: $ToP_j^t$
    Parameters: $\lambda \in [0.5, 1[$

        $ToP_i^t = \lambda \cdot QoP_j^t + (1-\lambda) \cdot ToP_i^{t-1}$,        (2)
where Quality of Provider
        $QoP_j^t = \frac{1}{|SR^t|} \sum_{i \in SR} \rho_{ij}^t \cdot ToR_i^t \cdot f_{ij}^t$
for
        $\rho_{ij}^t = \mathrm{sinc}(1 - f_{ij}^t) \cdot \mathrm{sinc}(\Delta f_{ij}^t)^{\Delta t}$
▷ $\Delta t$ represents the difference between $t$ and $t-1$, $SR^t$ refers to all SR agents at time $t$, and $\Delta f_{ij}^t = |f_{ij}^t - f_{ij}^{t-1}|$
```

-SR+SP agent: Indeed, agents may be capable enough of playing SR+SP roles for one service process in a way that they rate service A and give service B to other agents, as it does not make sense that one agent performs a service to itself and then rate this service, especially this will strongly encourage self-promoting misbehavior as discussed in [11].

Algorithm 3 Assessing the SR+SP Agent: **T**rust **o**f **A**gent (*ToA*)

```
    Input: $ToR_k^t$, $ToP_k^t$        Output: $ToA_k^t$

        $ToA_k^t = \alpha_k^t \cdot ToR_k^t + (1-\alpha_k^t) \cdot ToP_k^t$,        (3)
for
        $\alpha_k^t = \frac{|R_k^t|}{|R_k^t| + |P_k^t|}$
▷ $ToR_k^t$ and $ToP_k^t$ are given by (1) and (2), where $i$,$j$=$k$. $R_k^t$ and $P_k^t$ refer to services rated and provided by agent $k$ until $t$.
```

Algorithm 3 is employed for SR+SP agents as it enables trust evaluation in both SR and SP sides, i.e., it takes both *ToR* and *ToP* into account. In *ToA* computation, we set α factor viewing the contribution workload of SP and SR to weigh the *ToR* and *ToP* values. Besides, it should be noted that the terms 'SR' and 'SP' are used to describe agents' roles in one service process, which means SR or SP agents can switch roles in different service processes over time (unlike SR+SP agents perform two roles in one service process), and their trust can also be computed by Algorithm 3.

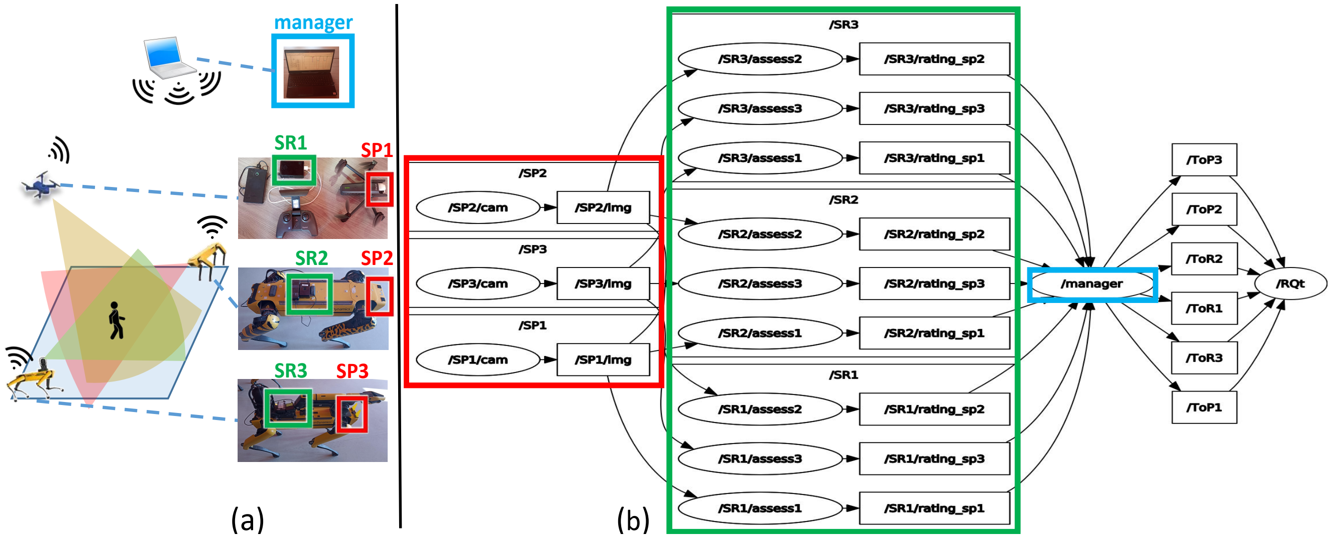

Figure 2. Real MAS implementation by using ROS 2, where SR, SP, and trust manager are highlighted by corresponding colors: (a) Considered scenario and implemented hardware; (b) Software-level architecture generated by RQt

3 Implementation

As illustrated in the left part of Fig. 2 (a), a three-robot scenario is considered for implementation, where robots accomplish a common mission using their cameras to cooperatively monitor a human. The image transmission frequency is fixed at every 500ms. As the preliminary implementation of the designed trust model, a HOG (Histograms of Oriented Gradient) [3] human recognition algorithm from OpenCV is adopted by each Raspberry Pi card to return the probability describing the existence of the target human, i.e., each SR (Raspberry Pi card) evaluates three SPs (cameras). In such a manner, a 3-SR and 3-SP case is built, and the above-mentioned probability will be taken into trust computation as SRs' feedback. Besides, we set λ=0.5 since the historical record and current behavior are considered equally important. Next, we are going to detail the implemented hardware and software.

3.1 MAS Setup: Implemented hardware

Implemented robots, as shown in Fig. 2 (a), consist of one aerial drone (Anafi, ™Parrot) and two quadruped ground robots (Spot, ™Boston Dynamics). Each ground robot is equipped with a Raspberry Pi card, and a remote control is connected to one Raspberry Pi for the drone. Besides, a laptop is deployed as the trust manager. All Raspberry Pi cards and the manager run Ubuntu 22.04 and ROS 2 Humble, and we set a 5GHz Wifi access point to enable communication between them.

3.2 Service Process on ROS 2

Compared with other Robotics Software Frameworks (RSF), ROS 2, an open-source software platform for robotics based on DDS (Data Distribution Service) [5], is best suited to multi-agent robotic systems and for data exchange [4]. For this reason, we considered ROS 2 for the implementation. The software architecture is depicted in Fig. 2 (b) by a ***Node Graph***, which is composed of nodes, topics, and namespaces. The namespaces correspond to the involved 3 SRs and 3 SPs. Each node is an executed process: the first node **`cam`** retrieves images from robots' cameras; Each SR contains 3 **`assess`** nodes that return the feedback assessing 3 SPs. Before calculating *ToR* and *ToP* values, the manager will realize an approximate synchronization of nine ratings (feedback) produced by the nine **`access`** nodes. After that, the node **`manager`** computes the trust of role-based agents by employing algorithms in Section 2. Finally, 3 *ToR* and 3 *ToP* values will be output by the **`manager`**. While one robot and the equipped Raspberry Pi card can be regarded as an individual SR+SP agent to conduct *ToA* calculation, in our preliminary implementation, we only evaluate SR and SP roles separately.

4 Preliminary Results

Via RQt, agents' trust values are illustrated in Fig. 3 and Fig. 4, where **On-Off Attack** (OOA) and **Bad Mouthing Attack** (BMA) are launched, respectively. We can notice that in both figures SRs and SPs are working properly at the beginning, where *ToP* and *ToR* values are close to 1.

We imposed the target human moves quickly to create environmental perturbation, once at 18 and once at 30 seconds, which explains several changes in trust values before the 40s in Fig. 3. Then, between 50 and 90 seconds, the camera SP3 of the quadruped robot was dedicated to performing OOA, where it switches between good and bad over time. It can be seen the red curve representing the OOA attackers' *ToP*3 decreases to 0 while *ToP*1 and *ToP*2 are also slightly lowered. As the gap between the attacker and the well-behaved ones is sufficiently large, the OOA attacker can be identified.

One other type of attack is tested and visualized in Fig. 4, where SR3 is fixed as the BMA attacker between 20 and 45 seconds to rate 0.5 for all received services, no matter how the SPs really perform. The attacker aims to ruin the reputation of good SPs by rating them negatively. The red curve representing the BMA attacker's *ToR*3 decreases to 0.4, where *ToR*1 and *ToR*2 remain at 0.6. On the one hand, this figure clearly shows that the BMA attacker SR3 can be distinguished from normal SRs. On the other hand, all well-behaved SPs are influenced harmfully in a way that their *ToP* values drop to a low level below 0.5. This is because the 1/3 malicious rater case reaches the limit of the Byzantine problem, in a larger-scale MAS with more SRs and SPs, such negative effects caused by dishonest SR will be significantly reduced.

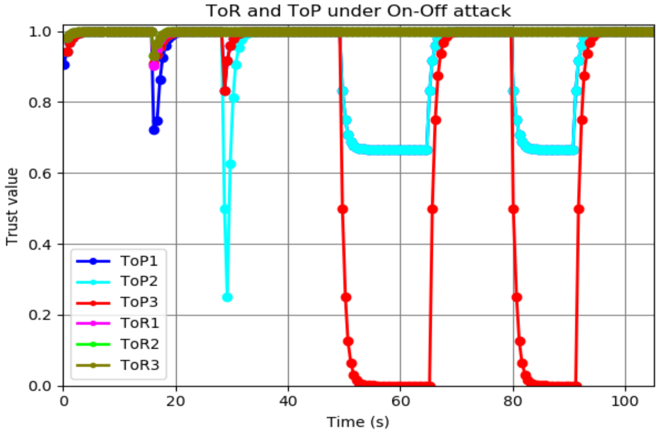

Figure 3. Changes in *ToR* and *ToP* in the presence of OOA, launched by SP3 at the 50s. Before that, *ToP* and *ToR* values converge to 1 but do not remain entirely stable due to environmental perturbation (e.g., shooting angles and lighting).

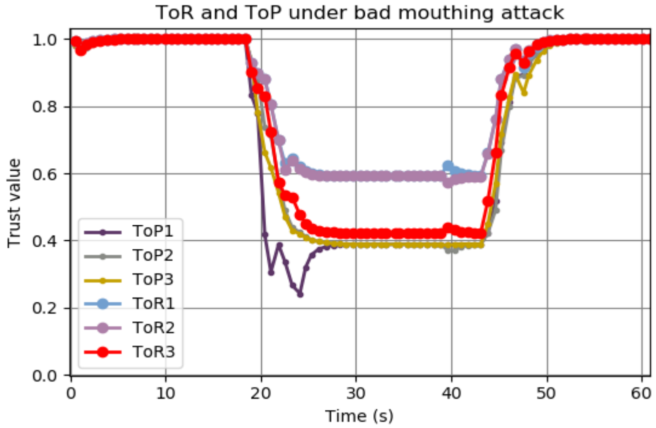

Figure 4. Changes in *ToR* and *ToP* in the presence of BMA, launched by SR3 at the 18s. While the BMA attacker can be identified, the SPs' *ToP* values are largely influenced in a negative manner.

5 Conclusions and future work

In this paper, we first presented our role-based trust model aiming to determine the trust of role-based agents according to their real-time behaviors during the service process. And then, we applied this model in a real-world robotic MAS by using ROS 2, including three robots and three Raspberry Pi cards to build a 3-SR and 3-SP scenario. The first results of implementation show the feasibility of our proposed framework in real robotic MAS and the resilience against two attacks, namely OOA and BMA. We can also observe that the negative influence from the malicious SR side is much more than the malicious SP side. For future work, we plan to study the assessment of SR+SP agents, whose trustworthiness can be computed by Algorithm 3. Furthermore, we are also interested in extending the IoA size by involving more agents to investigate the scalability of the proposed model and also apply it to a more mobile scenario with robots. As recommended in [10], an access control mechanism that allows agents to enter and leave flexibly for their own interests can also be considered in future work.

6 References

[1] A. Aref and T. Tran. An integrated trust establishment model for the internet of agents. *Knowledge and Information Systems*, 62(1):79–105, 2020.

[2] D. Calvaresi, M. Marinoni, A. Sturm, M. Schumacher, and G. Buttazzo. The challenge of real-time multi-agent systems for enabling iot and cps. In *Proceedings of the international conference on web intelligence*, pages 356–364, 2017.

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition (CVPR '05)*, volume 1, pages 886–893, San Diego, United States, June 2005. IEEE Computer Society.

[4] P. Iñigo-Blasco and et al. Robotics software frameworks for multi-agent robotic systems development. *Robotics and Autonomous Systems*, 60(6):803–821, June 2012.

[5] S. Macenski and et al. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66), May 2022.

[6] J. C. Nieves, D. Andrade, and E. Guerrero. Maiot-an iot architecture with reasoning and dialogue capability. In *Applications for Future Internet: International Summit, AFI 2016, Puebla, Mexico, May 25-28, 2016, Revised Selected Papers*, pages 109–113. Springer, 2017.

[7] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent. Trust management system design for the internet of things: A context-aware and multi-service approach. *Computers & Security*, 39:351–365, 2013.

[8] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera. Towards trustworthy internet of things: A survey on trust management applications and schemes. *Computer Communications*, 160:475–493, 2020.

[9] V. Stefanova-Stoyanova and I. Stankov. Multi-agent systems (mas) in the area of iot and using a model with distributed shared memory system (dsm). In *2020 XXIX International Scientific Conference Electronics (ET)*, pages 1–4. IEEE, 2020.

[10] R. Su, A. R. Sfar, E. Natalizio, P. Moyal, and Y.-Q. Song. Pdtm: Phase-based dynamic trust management for internet of things. In *2021 International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2021.

[11] R. Su, A. R. Sfar, E. Natalizio, P. Moyal, and Y.-Q. Song. Ensuring trustworthiness in ioit/aiot: A phase-based approach. *IEEE Internet of Things Magazine*, 5(2):84–88, 2022.

[12] M. Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

[13] H. Yu, Z. Shen, and C. Leung. From internet of things to internet of agents. In *2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing*, pages 1054–1057. IEEE, 2013.

[14] L. Zeynalvand, J. Zhang, T. Luo, and S. Chen. Masa: Multi-agent subjectivity alignment for trustworthy internet of things. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2013–2020. IEEE, 2018.