

Adaptive Trading of Cloud of Things Resources

Ahmed Salim Alrawahi
and Ahmad Lotfi

*School of Science and Technology
Nottingham Trent University
Nottingham, UK*

ahmed@alrawahi.org, ahmad.lotfi@ntu.ac.uk

Kevin Lee

*School of Information Technology
Deakin University
Melbourne, Australia*

kevin.lee@deakin.edu.au

Abstract—Cloud of Things (CoT) consists of heterogeneous Cloud and Internet of Things (IoT) resources. CoT increasingly requires adaptive run-time management due to the CoT dynamism, environmental uncertainties and unpredictable changes in IoT resources. Adapting to these changes benefits particularly trading of CoT resources where the adaptability of traded resources and applications remains a challenge. Run-time changes in CoT trading environments can impact vital aspects including resource allocation, resource utilisation and application performance. This paper adopts monitoring, analysis, planning and execution (MAPE) model from autonomic computing to support adaptations when trading CoT resources. This is achieved by applying the MAPE model to systematically capture and identify changes in CoT environment. Based on the identified adaptations, an adaptive model is proposed to react to these changes.

1. Introduction

The rapid developments in Cloud Computing and Internet of Things (IoT) enable the emerging Cloud of Things (CoT) paradigm. CoT integrates both technologies in a complementary manner facilitating integrated computing capabilities as services. However, CoT has several challenges including the heterogeneity of its resources, the unpredictable dynamism of its application and uncertainties of the surrounding environment. These challenges become more complex when trading CoT resources.

CoT dynamism involves constrained resources, complex application requirements, unpredictable resource mobility, uncertain resource availability, and network topology changes due to failures. Static approaches in CoT infrastructure deployment and management seem unfeasible. While the potential of adaptivity support can be justified, proposing such support for trading CoT resources remains complex and sophisticated.

Trading CoT resources describes the process of commoditising CoT resources by offering them to be consumed by multiple applications. This involves mapping resources from multiple providers to applications from multiple consumers based on the application requirements, perform resource allocation based on the mapping and scheduling of

the allocated resources. Addressing the challenge of adaptivity support is complicated due to the following reasons. 1) Changes in resource state and/or application requirements may occur before, during or after the trading process has initiated. 2) Unified monitoring of changes in CoT resources is not clear due to the heterogeneity of CoT resources and their constrained energy capabilities. 3) Adapting to scalable requirements of CoT application requires sophisticated coordination between resources, application and the marketplace.

To address the above-mentioned challenges, this paper adopts monitoring, analysis, planning and execution (MAPE) model from autonomic computing community [1] to support adaptations when trading CoT resources. MAPE functions capture the changes, investigate them, propose the required adaptations and execute them. The aims of this paper are; 1) Analyse the requirements for adaptive CoT, 2) Propose MAPE-based adaptivity model for trading CoT resources, 3) Systematically identify run-time adaptations when trading CoT resources, and 4) Discuss how the proposed model achieves adaptive trading of CoT resources.

The remainder of this paper is organised as follows. Section 2 presents background on trading CoT resources and autonomic computing along with several motivating scenarios. The related work is reviewed in Section 3. Section 4 provides an analysis of adaptations in trading CoT resources. Section 5 proposes MAPE-based model to support adaptive trading of CoT resources. Conclusions and future work are discussed in Section 6.

2. Background

This section provides a brief background on trading CoT resources, autonomic computing and presents several motivation examples.

2.1. Trading Cloud of Things Resources

CoT is a large-scale computing infrastructure by nature. It can potentially consist of few dozens to billions of connected resources. This scalable infrastructure currently requires dedicated hardware deployment, application development, infrastructure management and maintenance.

The required up-front investment hinders many stakeholders from adopting the technology [2]. One reason for this challenge is the limited shared access to IoT physical resources by multiple applications. The limitation is caused by the constrained power and computing capabilities of IoT resources, particularly when compared with Cloud resources. Improving shared access to CoT resources have the following advantages. The costs of CoT are likely to be reduced as resources are reused by multiple applications. Increasing the reusability of CoT resources are expected to reduce the existing costly replication of similar CoT infrastructures. These factors may lead to overall higher rates of CoT adoption. One potential approach to the improve shared access is by commoditising CoT resources.

Trading CoT resources appears where infrastructure deployers offer their resources to be consumed as a service by application developers. Commoditising CoT resources into services provides positive isolation between infrastructure deployment and application development. This segregation improves the focus of providers on the management and maintenance of their infrastructure regardless of the consumer applications. Similarly, consumers focus on their application development minimising hardware-specific considerations. The approach of trading CoT resources is successfully used in similar paradigms to CoT such as Cloud Computing, Grid and Fog computing.

Realising trading CoT resources requires adaptive techniques to ensure the following. 1) Changes in CoT resources, applications or the surrounding environment are captured and analysed, 2) Adaptations to the captured/analysed changes are planned, and 3) The proposed adaptations are executed according to the plan. Adaptivity support in trading CoT resources is vital for providers and consumers. Resource providers are required to react to changes in their infrastructure (e.g. power/communication outages) to ensure the availability of resources and provide the required maintenance. Similarly, application developers -consumers- require the adaptivity support to ensure their application requirements are fulfilled.

2.2. Autonomic Computing

Autonomic computing refers to the ability of computing systems to manage themselves based on predefined goals set by administrators [3]. It aims to dynamically adapt the system behaviour to unpredictable changes in which the system complexity is optimised. This optimisation is achieved by using systematic adaptive models that enable systems to adapt themselves to context changes. One of the commonly used autonomic models is MAPE that is illustrated in Figure 1 [1].

MAPE model is composed of three main components. These are the managed resources, the touch-points, and the autonomic manager and the managed resources. The managed resources are the software and hardware components to be managed autonomically. In the context of trading CoT resources, these are the marketplace system, the CoT resources and the CoT applications. The second component

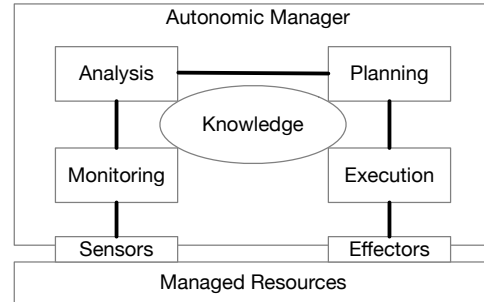


Figure 1. MAPE Model, see [1]

of the MAPE model is touch-points. They are manageability interfaces to collect details (e.g. events, log files) about changes in the managed resources and implement adaptations accordingly. Sensors collect details from the managed resources while effectors enable state adaptations to the managed resources.

The autonomic manager is the third component that represents the autonomic control loop. This loop consists of the following four functions. Monitor, Analyse, Plan and Execute. Monitor function utilises sensors to collect details about changes and prepares them for analysis by identifying patterns of change causes. Analyse function investigates the reported changes to understand the current state of monitored resources and proposes a change request if necessary. Plan function structures the adaptations needed to alter the managed resources and create the change plan for the execution function. Execution function implements the adaptations plan using the effectors. Updating knowledge can be part of the adaptations execution. The knowledge is a shared data repository among the four functions of the control loop.

2.3. Motivating Scenarios

CoT is emerging to meet the requirements of various real-world applications. A wide range of CoT applications requires CoT resources to be operable in continuously changing environments under unpredictable circumstances [4]. Adaptations to CoT dynamism can be categorised into three types as follows. 1) Environmental changes, 2) resources changes, and 3) application changes. In all types, middleware and system architecture have to adapt to the new changes.

Scenario 1. Environmental changes such as heavy rains, lightning, fog and snow are likely to have a direct and indirect effect on CoT resources. Consequently, the performance of CoT applications is likely to be affected. For instance, traffic monitoring application bids for motion-detection cameras with specific area coverage. The marketplace allocated the required cameras to the application for the required time. During the lease time, thick fog starts and the allocated cameras are hindered. The requested area coverage is significantly reduced. Changes happened after the resources have been already allocated. The marketplace

aims to capture such external forces and plan adequate adaptations. This may include suspending resources utilisation and resuming later or releasing the allocated resources back to the marketplace for a later round of trading. The goal of such adaptations is to ensure the availability of allocated resources to consumer applications during the utilisation period without considerable changes.

Scenario 2. Changes to resources have various causes and effects on CoT systems. Causes may include a power outage, connectivity interruption and network interference. This scenario considers additional changes to the resource location that have a direct effect on CoT application. This is demonstrated by a CoT marketing application that uses footfall sensors and smartphones sensors to track pedestrians around a specific business area. The application coordinates between all resources to push advertisements to the clients' smartphone within the target area. Pedestrians may walk out of the targeted area where the feed from smartphones become useless for the application. Adaptation in this scenario aims to ensure the availability of enough number of smartphones within the required area. In the case of an increasing number of 'out of area' resources, the marketplace can propose a new set of additional mobile resources within the target area.

Scenario 3. As stated in the other two scenarios, the performance of CoT applications can be affected by environmental forces as well as changes to functionalities of resources. This scenario considers the requirement changes in CoT application itself. Applications often require the existing resource allocation to scale up or down based on changing requirements. This can be achieved in a dedicated infrastructure to the application while it is challenging in a trading environment. For instance, an environmental monitoring application detects a pollution spike and require more resources to provide more accurate monitoring. The goal of adaptations, therefore, is to dynamically enable the application requesting additional resources and releasing them back to the marketplace based on its requirements.

3. Related Work

Adaptations to changes in IoT and Cloud systems benefit vital aspects including system performance, reliability and dynamism. This section reviews related work to adaptivity in CoT systems.

3.1. Adaptive CoT Systems

An early attempt to address adaptivity requirements of CoT is made in [5]. The study proposes a middleware to enable smartphones acting as adaptive gateways between IoT and Cloud services. Another early try by [6] presents a two-level model to manage resources at Cloud and IoT layers autonomously. To aid the development of adaptive IoT-based systems, a methodology is introduced in [3]. The method combines patterns from autonomic computing and cognitive computing to capture different-level adaptations. The captured changes are then modelled to enable adaptive

IoT architectures. Another MAPE based model is introduced in [7] to describe adaptations in IoT devices and data they produce. The study also proposes vocabularies needed to describe the relations between IoT resources and the surrounding environment.

Adaptivity in CoT architectures is also studied. An adaptive architecture is proposed in [8] to reconfigure edge resources based on changes in the workload. The architecture aims to improve performance, maximise energy efficiency, and minimise the latency of IoT applications. Another architecture that autonomously manages IoT sensors is developed in [9]. It integrates machine learning and rule specification to support adaptations in healthcare applications. In [10], an architecture is proposed to adapt to the run-time context changes. The architecture addresses adaptations to the unexpected disappearance of IoT resources.

A decentralised architecture is introduced in [11]. The run-time monitoring method of resource utilisation aids the architecture adaptability to extend the service life of power constrained resources. Another adaptive architecture is proposed in [12] for manufacturing applications. The architecture goal is to automate reactions to changes in manufacturing environments using IoT devices. Two variants of a CoT architecture are presented in [13]. The architecture dynamically scales up and down by adding and removing IoT nodes based on the changes in incoming data to the system.

The adaptability of CoT platforms is also studied. A platform to support adaptations in CoT deployments and configuration is introduced in [14]. The platform adapts to the relocation changes of IoT resources among IoT gateways. FarmBeats [15] is an adaptive IoT platform for agriculture applications. The platform addresses the adaptivity requirements for weather-related power and Internet interruptions. A platform for CoT manufacturing applications is proposed in [16]. It supports sensing the changes in production requirements, analyses them and integrates the new requirements into the new production plans. Another platform that address changes to data sensing and collection is presented in [17]. The platform implements two adaptive techniques to control data sensing and collection in systems with limited 3G plans.

A different adaptivity approach is taken in [18] to use an optimisation based model in monitoring changes in IoT environments. The model aims to optimally capture the changes and identify them as monitored metrics. It also adapts to environmental events by re-optimising the frequency of monitored metrics. Another optimisation-based adaptive framework is presented in [19]. The framework employs the online convex optimisation algorithm to enable fog resources adapting to the varying IoT user requirements. A low-cost framework for CoT monitoring is proposed in [20]. The framework captures run-time changes in data streams and dynamically adjust the frequency of data monitoring and dissemination. As a result, lowering the energy consumption, network traffic and provide efficient data processing.

3.2. Gap Analysis

As discussed in the previous section, the state of the art research provides various adaptivity techniques for CoT architectures and platforms. Due to the heterogeneity of CoT resources, adaptivity-related challenges are still emerging [21]. This includes how to efficiently reacting to changes in CoT resources and applications while they are traded. Recent blockchain based solutions for trading CoT resources are presented in [22] and [23]. Both solutions address resource management aspects while contributions to adaptability aspects of trading CoT resources are still unclear.

In order to fill that gap, this paper aims to achieve the following. 1) Conduct requirements Analysis for adaptive CoT, 2) Propose autonomic-based adaptivity model for trading CoT resources, 3) Systematically identify run-time adaptations when trading CoT resources, and 4) Discuss how the proposed model achieves adaptive trading of CoT resources.

The adoption of MAPE framework form autonomic computing in CoT context is justified as follows. MAPE model has been successfully used to perform self-adaptation in Cloud, IoT and edge systems [24]. Dynamism in trading CoT resources is inherited from both Cloud and IoT where MAPE model is already tested. It also has some similarities to adaptive grid scheduling as presented in [25]. Therefore, this work is inspired by [24], [25] and applied to the domain of trading CoT resources.

4. Adaptations in Trading CoT Resources

As discussed earlier in Section 2, adaptations in CoT respond to either external forces (e.g. environmental) or in the execution environment (e.g. resources and application). The scope of this paper is limited to adaptivity in resources and applications only.

The adaptivity support for trading CoT resources would typically be implemented on a marketplace system such as [26] to provide a full control loop of the system changes and adaptations to these changes. MAPE loop can take different patterns based on the system decentralisation. The adaptability of the trading system can be supported by one of the following patterns [24].

- **Centralised.** The adaptive control loop is performed centrally through the marketplace. This pattern is illustrated in Figure 2A.
- **Information Sharing.** This pattern is presented in Figure 2B where the marketplace components are fully distributed addressing localisation requirements of applications. In this pattern, the monitoring function is responsible for coordinating with other regional peers.
- **Regional Planning.** This pattern also shows a distributed marketplace. Each local planning function coordinates with one another to propose the adaptations plan for local or global changes. This pattern is depicted in Figure 2C.

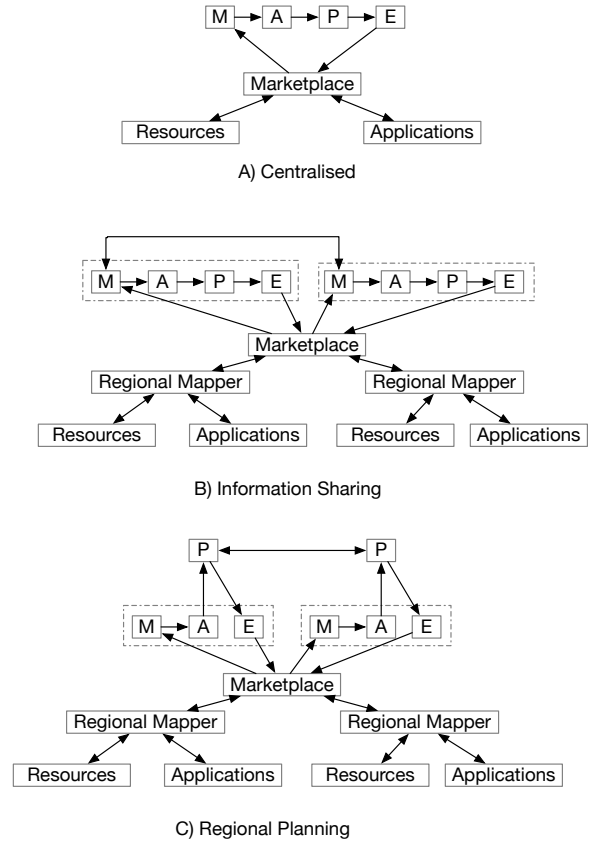


Figure 2. Adaptive CoT Resource Trading Patterns, inspired by [24]

4.1. Resource Adaptations

Resource adaptations describe the required reactions to resource changes during or after resource mapping and allocation processes by the marketplace system.

- **Resource Replacement** CoT resources are heterogeneous and dynamic. Resources can join and disappear spontaneously due to several reasons (e.g. battery depleted, communication outage). Adaptations have to consider any potential faults and topology changes and replacement resources with similar properties to the lost ones.
- **Resource Removal** resources can be removed if they are malfunctioning, in particular when affecting other resources or application functionality. Resources can also be removed if they are over-allocated to an application and the removal does not compromise the application performance.
- **Resource Mobility.** Mobile resources can move out of the application coverage area. Adaptations are needed to find alternative resources, maintain connectivity and minimise data loss.
- **Restart Resource.** Infrastructure maintenance may require adding additional hardware, replacing existing components such as the battery or changing the operating properties. Restarting the resources can be nec-

essary to apply such changes.

- **Communication Interference.** Dense populated areas may have multiple CoT deployments. Such replication can cause interference among wireless resources. Changing the transmission channel and maximise the transmission power may become useful.

4.2. Application Adaptations

Application adaptations refers to the reactions required to address the run-time changes of applications. These include the following.

- **Changes in Application Requirements.** Application requirements are likely to be changed after resources have been mapped and allocated to the application. This can be due to the increase or decrease in demand for application services. These adaptations consider requesting new resources from the marketplace to scale-up or releasing underutilised resources to scale-down. Changes in application requirements may include requesting an increase or decrease of the resource consumption time. Therefore, adaptation to the scheduling may be required.
- **Increasing Redundancy of Allocated Resources.** Time-sensitive CoT applications (e.g. emergency services) may require running their application services in different resources simultaneously. Adaptations may include increasing the number of parallel resources that can be utilised by the application simultaneously.
- **Decreasing Redundancy of Allocated Resources.** In trading environments, CoT applications have different priorities when bidding for resources. Emergency services application has a higher priority than a marketing application. Potential adaptations to the resource allocation may be required to release some of the paralleled resources already allocated to a lower priority application to join the application with higher priority.
- **Migrate Tasks Across Resources.** Cost of resources may change over a period of time and become infeasible for some applications at a given time. Useful adaptation is to move some tasks of the application to different resources with similar properties within the application budget.

5. MAPE-Based Adaptivity Model for CoT Marketplace

Due to the heterogeneity of CoT resources and the complexity of CoT application, the need for an autonomic-based adaptive model for efficient trading of CoT resources is justified. This section presents the proposed MAPE-based model to capture the potential adaptations discussed in Section 4 and proposes adequate plans to execute them. The model is illustrated in Figure 3.

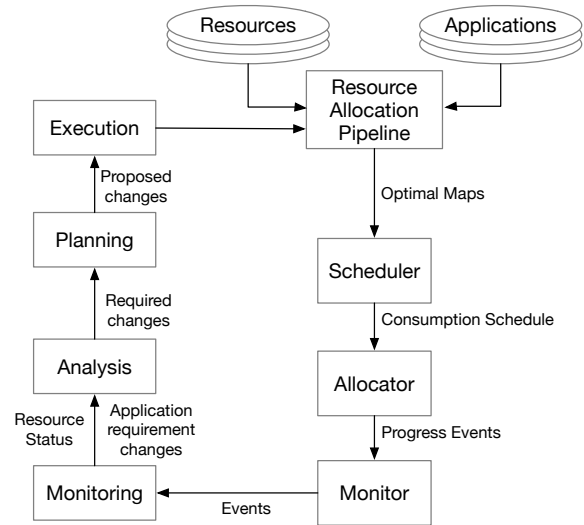


Figure 3. Overview of The Proposed Model

5.1. Monitoring

As shown in Figure 3, monitoring the resource allocation pipeline is vital for capturing important details about CoT resources and applications. The following events are monitored to identify run-time in changes resources and applications.

- M.1 Resource State.** This can be detected by a connectivity check or by monitoring the resource feed to the application.
- M.2 Resource Location.** Resource location can be compared with the request area coverage of the application. This can identify out of area resources.
- M.3 Wireless Communications.** A list of all similar wireless protocols that use the same network channel.
- M.4 Available Resources.** A list of changes in the available resources, mapped resources but not allocated yet and already allocated resources.
- M.5 Level of Resource Utilisation.** This measures and log the processing, memory and network usage of a resource.
- M.6 Mapping Process.** List of resources mapped to applications or log of the ongoing mapping process.
- M.7 Utilisation Schedule.** Records of resource utilisation lease time and the progress of resource utilisation.
- M.8 Application Requirements.** Log of all changes to the application requirements made after requesting resources from the marketplace.

Monitored details are collected, aggregated, correlated and filtered by the monitoring function via the sensors and then prepared to be transferred to Analyse function.

5.2. Analysis

Analysing the monitored events is critical to the system due to the complexity of data analysis that aims to identify potential issues or/and required adaptations.

- A.1 Resources Unavailability.** Analysing events M.1, M.2, M.4 and M.6 may indicate resource unavailability. Further analysis may show the root cause of the issue. For instance, a mobile resource has moved away from its coverage area or the resource power run out.
- A.2 Unrealistic Utilisation Schedule.** Applications that can not meet their scheduling deadlines may imply unrealistic scheduling or frequent changes in application requirements. This can be observed by analysing M.5, M.7 and M.8.
- A.3 Over-utilised Resource.** Resource over-utilisation occurs when one or more of the resource capabilities is being under heavy usage for a certain time. Analysing events of M.5 may provide early detection of potential resource bottleneck or failure.
- A.4 Under-utilised Resource.** Similarly, investigating the details from M.5 may reveal a low level of resource utilisation or idle resource status.
- A.5 No Optimal Mapping.** Trading CoT involves mapping resources to applications. Mapping is not always possible because of possible unmatched application requirements or constrained application budgets. Analysing M.6 is an important aspect of the trading process.
- A.6 Resources availability.** Opposite to A.1, looking at details monitored earlier in M.4-M.7 can detect available resources that are not mapped to applications or resources to become available shortly due to the end of other application consumption.

Analysing the monitored events is crucial to determine the required adaptations. In case of adaptations are required, a change request is sent from the analyse function to plan function to describe such adaptations.

5.3. Planning

The Plan function is responsible for the systematic formation of the required adaptations. The planning process is based on the outcomes of the analyse function. The following plans are discussed.

- P.1 Propose Alternative Resource Allocation.** Based on the analysis of A.1, A.5 and A.6, a proposal of alternative resource allocation is made to cope with any application performance issues caused by resource unavailability.
- P.2 Make New Schedule.** This plan responses to A.2 problem where an unrealistic deadline for resource consumption is set. Rescheduling the affected application tasks and resources.
- P.3 Improve Balanced Redundancy.** To address the challenge of imbalanced resource utilisation analysed in A.3 and A.4, improving the redundancy of allocated resources is necessary.
- P.4 Trigger New Mapping Process.** No optimal mapping analysed in A.5 implies the inability of the marketplace to match CoT applications to appropriate resources. A new matching process should be triggered promptly.

After the required adaptations are planned, the plan function passes them to the execute function.

5.4. Execution

The execution function uses the touch-point effectors to perform the required adaptations to CoT resources and applications.

- E.1 Perform Alternative Resource Allocation.** In response to the proposed plan in P.1, the plan is executed by triggering the available resources for a new round of resource allocation.
- E.2 Execute The New Schedule.** The proposed schedule in P.2 is adopted by the marketplace system where it has optimised resource utilisation times.
- E.3 Balance The Load on Resources.** The plan to improve resource redundancy in P.3 is executed to balance the load on resources. This aims to minimise future wasting or loading resources.
- E.4 New Mapping Process.** Proposed in P.4, the new mapping process can be triggered by terminating unsuccessful mapping process that is underway before it ends and initiates a new process with tuned optimisation metrics.

The execution function involves updating the knowledge of the MAPE model. The knowledge is used to improve the adaptations considered by the autonomic manager at each iteration of the control loop.

5.5. Discussion

Adaptivity support for trading CoT resource environments is vital. This support should be considered as a requirement for the marketplace system architecture. The MAPE model provides the flexibility that CoT environments need. This is evidenced by the ability of MAPE in monitoring, analysing, planning and executing the required adaptations of CoT resources and applications.

Another benefit of using the MAPE model is the straightforward integration with the marketplace system architecture. This is achieved by using the touch-point sensors and effectors to control the adaptivity support of CoT resources and applications submitted to the marketplace for trading. Furthermore, MAPE provides CoT systems with the necessary adaptivity support across different distributed layers. This is illustrated in Figure 2 showing the patterns of adaptation.

6. Conclusions and Future Work

This paper focuses on the challenge of adaptivity support for trading CoT resources. The paper adopts the MAPE model from the autonomic computing community to structure and perform the required adaptations in CoT trading environment. The proposed model shows how changes to CoT resources and applications can be captured, analysed, planned and executed. The adaptivity support across decentralised layers are considered using three different adaptivity patterns.

The scope of this work is limited to theoretical proof-of-the-concept. Planned future work will involve expanding the

proposed model to react to more complex states including external forces to the CoT environment. Experimental integration of the proposed model into a marketplace system architecture is planned to enable experimental evaluation and analysis of the model.

References

- [1] IBM, "An architectural blueprint for autonomic computing," *IBM White Paper*, vol. 31, pp. 1–6, 2006.
- [2] S. Akkermans, W. Daniels, G. Sankar R, B. Crispo, and D. Hughes, "Cerberos: a resource-secure os for sharing iot devices," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 2017, pp. 96–107.
- [3] E. Mezghani, E. Exposito, and K. Drira, "A model-driven methodology for the design of autonomic and cognitive iot-based systems: Application to healthcare," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 3, pp. 224–234, 2017.
- [4] H. Fouad and I. S. Moskowitz, "Meta-agents: Managing dynamism in the internet of things (iot) with multi-agent networks," in *2018 AAAI Spring Symposium Series*, 2018.
- [5] R. Golchay, F. L. Mouël, S. Frénot, and J. Ponge, "Towards bridging iot and cloud services: proposing smartphones as mobile and autonomic service gateways," *arXiv preprint arXiv:1107.4786*, 2011.
- [6] S. Distefano, G. Merlino, and A. Puliafito, "Enabling the cloud of things," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2012, pp. 858–863.
- [7] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "Iot-o, a core-domain iot ontology to represent connected devices networks," in *European Knowledge Acquisition Workshop*. Springer, 2016, pp. 561–576.
- [8] A. Munir, P. Kansakar, and S. U. Khan, "Icfiot: Integrated fog cloud iot: A novel architectural paradigm for the future internet of things," *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 74–82, 2017.
- [9] D. Scheunemann, V. Tabim, A. Yamin, J. Lopes, and C. Geyer, "Exploring autonomic sensors processing in the internet of things," in *2018 3rd International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*. IEEE, 2018, pp. 1–6.
- [10] F. Alkhabbas, R. Spalazzese, and P. Davidsson, "Architecting emergent configurations in the internet of things," in *2017 IEEE International Conference on Software Architecture (ICSA)*. IEEE, 2017, pp. 221–224.
- [11] J. Mocnej, W. K. Seah, A. Pekar, and I. Zolotova, "Decentralised iot architecture for efficient resources utilisation," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 168–173, 2018.
- [12] C. Alexakos, C. Anagnostopoulos, A. Fournaris, C. Koulamas, and A. Kalogeras, "Iot integration for adaptive manufacturing," in *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE, 2018, pp. 146–151.
- [13] A. C. Marosi, A. Farkas, and R. Lovas, "An adaptive cloud-based iot back-end architecture and its applications," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 2018, pp. 513–520.
- [14] D. Pizzolli, G. Cossu, D. Santoro, L. Capra, C. Dupont, D. Charalampos, F. De Pellegrini, F. Antonelli, and S. Cretti, "Cloud4iot: A heterogeneous, distributed and autonomic cloud platform for the iot," in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2016, pp. 476–479.
- [15] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An iot platform for data-driven agriculture," in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 515–529.
- [16] M. Dimitris, V. Ekaterini, and V. Zogopoulos, "An iot-based platform for automated customized shopping in distributed environments," *Procedia CIRP*, vol. 72, no. 1, pp. 892–897, 2018.
- [17] M. Rahman, A. Rahman, H.-J. Hong, L.-W. Pan, M. Y. S. Uddin, N. Venkatasubramanian, and C.-H. Hsu, "An adaptive iot platform on budgeted 3g data plans," *Journal of Systems Architecture*, 2018.
- [18] S. Tata, M. Mohamed, and A. Megahed, "An optimization approach for adaptive monitoring in iot environments," in *2017 IEEE international conference on services computing (SCC)*. IEEE, 2017, pp. 378–385.
- [19] T. Chen, Q. Ling, Y. Shen, and G. B. Giannakis, "Heterogeneous online learning for thing-adaptive fog computing in iot," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4328–4341, 2018.
- [20] D. Trihinas, G. Pallis, and M. Dikaiakos, "Low-cost adaptive monitoring techniques for the internet of things," *IEEE Transactions on Services Computing*, 2018.
- [21] P. P. Ray, "A survey on internet of things architectures," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.
- [22] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2019.
- [23] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted internet of things," *IEEE Transactions on Industrial Informatics*, 2019.
- [24] H. Muccini, R. Spalazzese, M. T. Moghaddam, and M. Sharaf, "Self-adaptive iot architectures: an emergency handling case study," in *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*. ACM, 2018, p. 19.
- [25] K. Lee, R. Sakellariou, N. W. Paton, and A. A. Fernandes, "Workflow adaptation as an autonomic computing problem," in *Proceedings of the 2nd workshop on Workflows in support of large-scale science*. ACM, 2007, pp. 29–34.
- [26] A. S. Al Rawahi, K. Lee, J. Robinson, and A. Lotfi, "An evaluation of optimisation approaches in cloud of things resource trading," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2018, pp. 208–215.