



HAL
open science

Temporal verification of mobile publish/subscribe machine-to-machine communications

Ghada Gharbi, Nawal Guermouche, Thierry Monteil

► **To cite this version:**

Ghada Gharbi, Nawal Guermouche, Thierry Monteil. Temporal verification of mobile publish/subscribe machine-to-machine communications. IEEE International Symposium on a World of Wireless, Mobile and Multimedia (WOWMOM), Jun 2014, Sydney, Australia. 10.1109/WoW-MoM.2014.6919022 . hal-01228310

HAL Id: hal-01228310

<https://hal.science/hal-01228310v1>

Submitted on 16 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Temporal Verification of Mobile Publish/Subscribe Machine-to-Machine Communications

Ghada GHARBI
CNRS, LAAS

7, avenue du colonel Roche,
F-31400 Toulouse, France

Université de Toulouse, UPS, F-31400
Email: ggharbi@laas.fr

Nawal GUERMOUCHE
CNRS, LAAS

7, avenue du colonel Roche,
F-31400 Toulouse, France

Université de Toulouse, INSA, F-31400
Email: Nawal.Guermouche@laas.fr

Thierry MONTEIL
CNRS, LAAS

7, avenue du colonel Roche,
F-31400 Toulouse, France

Université de Toulouse, INSA, F-31400
Email: monteil@laas.fr

Abstract—With the proliferation of data, components and services on the internet, cyber-physical systems are growing up these last years. Machine-to-Machine (M2M) is an example of a cyber-physical system where mobile communication is enabled between sensors, gateways and application services. M2M is used in several domains and in particular, in critical ones such as the healthcare domain. In such domain, requirements in term of Quality of Service (QoS) and particularly time related requirements in mobile environments must be rigorously respected. It is important to define an approach to ensure and validate these systems against these requirements. In this paper, we are interested in Publish/ Subscribe M2M communication based model. Our goal is to define a formal approach that aims at designing and validating M2M systems where we consider M2M mobile sensors and time related requirements. We define a graph based model to characterize Publish/ Subscribe M2M systems and model checking mechanisms to enable verification techniques.

I. INTRODUCTION

The Machine-to-Machine communications is a large label. It can be described as any technology involving a large number of intelligent machines sharing information and making collaborative decisions without human assistance. M2M networks evolve a wide variety of real-time monitoring applications such as smart homes, industrial automation and remote healthcare. In such systems, several communications models are deployed to enable communication between machines [1]. To ensure asynchronous communication, Publish/ Subscribe pattern is a good alternative to enable a full decoupling in time and space. This model allows efficient, distributed and selective content data delivery to a big number of users which addresses the scalability issue of M2M networks.

M2M research is moving rapidly to support M2M communications requirements. The issue of Quality of Service (QoS) is high in the M2M field. In our study, we are interested in time related QoS properties. In fact, M2M applications use public telecommunications networks to transfer data between devices and servers where data transfer delays must be respected. Another issue is: M2M devices are mobile and change location and the network they use for communications can affect the temporal requirements. To cater for dynamic changes of temporal properties due to M2M devices mobility, we need to define monitoring and verification mechanisms to ensure temporal constraints.

The presented work aims to define a formal verification and validation framework to monitor and ensure temporal constraints within mobile Publish/ Subscribe M2M systems. Given a mobile M2M system, our goal is to analyze the events of this system. More particularly, we focus on the temporal properties analysis. In fact, we intend to verify that the events driven from the M2M based component system, satisfy hard temporal constraints throughout the execution of the M2M application whatever the telecommunication network in service to handle the data traffic generated by the M2M devices. To do so, we rely on an autonomic manager to monitor temporal related requirements and trigger alarms in case of their violation. The knowledge base of the proposed framework is composed of graph-based models to characterize structural and temporal properties of Publish/ Subscribe M2M systems and timed automata for temporal constraints checking. We conducted a healthcare case study to validate our approach where stringent transmission delay's requirements should be respected for the continuous remote monitoring of patients vital signs.

The reminder of the paper is organized as follows: In Section II, we present the approach we propose to monitor and verify M2M systems while considering time related requirements and mobile devices. In section III we describe the experimentation setup. Section IV concludes and gives some perspectives for future work.

II. APPROACH

In this section, we present the proposed framework. Our objective is to check if the features of a connection (e.g transmission times), verify temporal constraints. After monitoring the connection status between an M2M device and the M2M server at application level, particularly events related to M2M devices's mobility, an analysis step is fulfilled for temporal requirements checking. In case of possible violations, the M2M application end users will be notified via alarms. To do so, the proposed framework relies on an autonomic manager which is based on the IBM autonomic manager architecture reference [2]. The processing of our framework can be described as follows:

– Collecting the status information about the started connection between the M2M devices and the new network. This step is out of scope of this work.

- Modelling component based description of Publish/ Subscribe M2M architecture taking into account the monitored temporal properties. We rely on graph-based representation [3] to model the Publish/ Subscribe M2M system.
- Transforming the component based model to an event based one. In fact, our goal is to analyze the events of a mobile M2M system. More particularly, we focus on the timed properties analysis. We intend to verify that the events driven from the M2M based component system, satisfy hard temporal constraints throughout the execution of the M2M applications in a mobile context. So an event-based description is more suitable for the detection of temporal conflicts. We define a graph grammar with a set of transformation rules to process event-based model translation [3]. To compute model transformation, the GMTE (Graph Matching and Transformation Engine) [4] is used. Once the transformation is complete, we proceed to the verification of the non-violation of the temporal properties with the UPPAAL model checker [5]. Temporal requirements of data transmission between events publishers and subscribers are specified as CTL (Calculation Tree Logic) formulas and checked by UPPAAL.
- Considering the analyzer output and based on policies, the planner constructs the needed plans of actions to determine the appropriate alarms notifications related to M2M end users in case of temporal constraints violation.
- Finally, the executer checks the received actions' plans and create alarms. In the use case of e-Health, we can distinguish between an sms alarm sent to the patient(s), however an email transmitted to the medical team.

A. Graph grammar for mobile Publish/ Subscribe M2M systems

1) *Temporal related requirements for mobile Publish/ Subscribe M2M systems*: Given that our objective is to design a mobile Publish/ Subscribe M2M architecture where events take place within the required temporal delays, we define the following temporal properties for events' analysis and validation:

- *Transmission Time* or the end-to-end time: It measures the time which a sensed data takes to be transported from the source to the broker. We distinguish between Request Transmission Time (abbreviated TT_{Req}) and Response Transmission Time (abbreviated TT_{Resp}).
- *Execution Time*: It represents the time taken to perform processing. It is abbreviated ET.
- *Response Time*: It measures the time between the service call and its acquittal. For example, for a server, the response time represents the elapsed time between receipt of the request and when the response is returned. It is abbreviated RT.
- *Service Time*: It measures the required time to prepare an event to be sent to a set of subscribers. It is abbreviated ST.
- *Waiting Time*: It measures the waiting time for an event to be send to a group of interested consumers. We define a different waiting time (abbreviated WT) for each topic.
- *Routing Time*: The time needed to send notification for a group of subscribers. It is abbreviated RoT.
- *Arrival Rate*: It is the mean number of arrivals (events) per unit of time. We define a different arrival rate (abbreviated AR_P) for each producer by topic.
- *Monitoring Period*: It represents our time constraint which is abbreviated MP_T . We want to check, at run time upon

detection of displacement of the M2M device to a new network with different temporal properties, that the following temporal constraint remains valid: *a consumer requires to receive an event in a Period P of its issue. If the event is received after this period, it is no longer valid.* We note that a consumer can specify, according to its requirements, a different period for each topic on which it is subscribed.

2) *Proposed graph grammar for mobile Publish/ Subscribe M2M communications*: To describe software architectures, graphs have the advantage of providing both intuitive and formally based formalism to describe a wide range of architectural structures. Graph-based models guarantee the correctness by construction of a software architecture by providing formal means for consistency's preservation such as graph grammars and graph rewriting rules [3].

A system configuration can be modelled using attributed graph, where vertices represent entities or events and edges specify their relationships. In the following, we give a general definition of a graph grammar.

Definition: Graph Grammar

A graph grammar is defined as a system $\langle AX; NT; T; P \rangle$ where:

- AX is the axiom;
- NT is a set of non-terminal vertices;
- T is a set of terminal vertices; and
- P is a set of graph grammar productions.

An instance of a graph grammar is obtained starting from the axiom AX by applying a sequence of productions P and where there is not non-terminal vertices.

Since we are interested on events' temporal properties checking, we define a graph grammar and rewriting rules to transform a component view to an event-based description of Publish/ Subscribe architecture in the context of mobile M2M communications.

We study a Publish / Subscribe architecture composed by N producers, M consumers and a queues's server. For component-based model, we consider four types of components corresponding to *server*, *producer*, *consumer* and *topic*. We consider that each node is identified by a unique identifier and a set of structural properties (e.g: the machine where is deployed on) and temporal properties defined previously. These nodes are defined as non-terminal in the graph grammar.

We consider two types of links between components: 1) The first concerns the communication links that connect a consumer or a producer to a server, 2) the second concerns the relationship of deployment, publication and subscription linking a topic respectively to a server, producer and consumer. The communication edges are attributes with the following properties: Request Transmission Time and Response Transmission Time. As to the second category of links, we consider these temporal properties: the Wait Time (associated to each Topic deployed on the Server), the Arrival Rate (corresponding to the arrival rate of a producer to a topic), the Monitoring Period (associated to a required period specified by a consumer to receive an event).

The terminal nodes describe events. We consider that each node is identified by a unique identifier. We distinguish the following classes of events: "sending_evt", "reception_evt", "reception_ack", "query_processing", "determination_subscribers" (the determination of the

subscribers list to a particular topic), “sending_notification” (the operation of sending notification to the set of consumers subscribed to a given topic once an event has been published on), “reception_notification”. Each event node is characterized by a group of structural (e.g. the event id, the event’ associated topic) and temporal properties (e.g execution time, waiting time).

Two relationships classes between event nodes are distinguished: 1) The first concerns the temporal interval required to move from one event to another, 2) defines the temporal constraints between the transmission of an event and its reception. This class of links is characterized by a const attribute equal to “const_link” assigned with the corresponding monitoring period.

We define the following graph grammar taking into account terminal and non terminal terms’s characterization:

$GG = (G, NT, T, P)$ where:

G is the graph where nodes represent components.

$NT = N(id_S, \text{“Server”}, idMachine_S, ET_S, RT_S, RoT_S, ST_S), N(id_P, \text{“Producer”}, idMachine_P, ET_P, RT_P), N(id_C, \text{“Consumer”}, idMachine_C, ET_C, RT_C), N(id_T, \text{“Topic”}, TV), N(\text{“Temp”}),$

$T = N(id_{se}, \text{“sending_evt”}, id_P, TV), N(id_{re}, \text{“reception_evt”}, id_P, TV), N(id_{ra}, \text{“reception_ack”}, id_P, id_{se}, TV), N(id_{qp}, \text{“query_processing”}), N(id_{ds}, \text{“determination_subscribers”}, RoT_S, WT, TV), N(id_{sn}, \text{“sending_notification”}, TV), N(id_{rn}, \text{“reception_notification”}, id_C, id_{se}, TV, MP_T),$ and

$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$

The specifications that have been elaborated for the relationships to be initiated or for a node to be instantiated, as well as how to do it, are described by the transformations rules of the graph grammar.

B. Verification using UPPAAL model checker

As said previously, our objective is to validate temporal properties of M2M Publish/ Subscribe configurations. To do so, we rely on model checking techniques [6] and we use the UPPAAL model checker. We transform the obtained event-based graph instance to a timed automata using an XML parser that we have developed.

Once the UPPAAL description is provided, we proceed to generating the CTL formulas corresponding to the temporal properties. We seek to check for instance if there exists a final state associated to a consumer in which the time required to receive an event is less than the monitoring period specified by the consumer. To this end, we consider the event-based graph’ edges tagged with const_link attribute and the value of the associated monitoring period. An example of UPPAAL implementation of such temporal property is: $(E \langle \rangle \text{Process.final_c1} \ \&\& \ t \langle 400 \rangle)$ that means: a consumer c1 requires to receive an event in a Period P less than 400ms of its issue.

III. EXPERIMENTATIONS

The objective of this section is to show the feasibility and study the scalability of the proposed approach. We consider a healthcare emergency situation where a remote medical team requires a hard real-time transmission of medical data. We

started by a configuration deploying two producers (the electrocardiogram and the oximeter sensors), two topics (blood pressure, and cardiac tracing) and three consumers representing the attending doctor, the emergency care and the outpatient clinics’ monitoring applications. In this case, the medical team monitors one patient. To evaluate scalability, the number of monitored producers (sensors) has been increased from 2 to 50 which means that the number of monitored patients has risen. The experimental performance evaluation focuses on the execution time. The results, shown in table I, demonstrate that the execution time to handle verification is in the order of ms seconds. These results remain acceptable in particular if we consider that the case where 50 sensors are moving simultaneously is very infrequent.

	Producers	Execution Time (ms)
Execution 1	2	10
Execution 2	10	37
Execution 3	50	3720

TABLE I: Experimentation results

IV. CONCLUSION

In this paper, we have proposed a formal verification approach of mobile Publish/ Subscribe M2M communications considering temporal requirements. Our approach is based on the following steps: collection of the temporal properties of the started connection between an M2M device and the server once a M2M device localization changes, event-modeling of Publish/ Subscribe systems taking into account the monitored temporal properties, checking of temporal constraints related to data transmission and triggering alarms in the case of non-compliance of temporal related requirements. The architecture transformation is specified and ruled using graph rewriting models and the temporal properties are checked using model checking mechanisms. The models elaborated in this paper have been illustrated in a healthcare business case where temporal properties are a critical factor since there are many situations in which a communication should be towardly to save the life of the patient.

As future work, we plan to extend the proposed work by adaptation features to handle the violations. Moreover, we propose to enrich our temporal verification of Publish/ Subscribe M2M communications with additional constraints.

REFERENCES

- [1] D. Boswarthick, O. Elloumi, and O. Hersent, “M2M Communications: A Systems Approach”, Wiley, 2012.
- [2] “An architectural blueprint for autonomic computing”, White Paper, IBM, June 2006.
- [3] G. Rozenberg, “Handbook of Graph Grammars and Computing by Graph Transformations”, Volume 1: Foundations, World Scientific, 1997.
- [4] M.A. Hannachi, I. Bouassida Rodriguez, K. Drira, and S. Epomares Hernandez, “GMTE: A Tool for Graph Transformation and Exact/ Inexact Graph Matching”, In GbRPR, 2013, vol 7877, pp.71-80, Springer 2013.
- [5] J. Bengtsson, K. Larsen, F. Larsen, P. Pettersson, and W. Yi, “UPPAAL a tool suite for automatic verification of real-time systems”, In Hybrid Systems III, vol. 1066, pp. 232-243, LNCS. New York, 1996.
- [6] P. Bouyer, and F. Laroussinie, “Model Checking Timed Automata”, In Modeling and Verification of Real-Time Systems, pages 111-140. ISTE Ltd. - John Wiley & Sons, Ltd., 2008