

2013 AASRI Conference on Intelligent Systems and Control

Design and simulation of the Timed Triggered Protocol (TTP) using agents

Sofiane Zaidi^a, Fateh Boutekkekouk^{a,*}

^aUniversity of Oum-El Bouaghi, BP358, 04000, Algeria

Abstract

In this paper, we present our agent based approach for design and simulation of the basic services of the Time Triggered Protocol (TTP) for automotive distributed embedded systems such as the startup algorithm, the membership, the messages sending and reception, errors detection and handling. The proposed architecture is modeled as a multi-agent system and implemented in the JADE platform following mainly the so-called O-MaSE methodology. Using JADE, We developed an ontology that provides an automatic interpretation of frame fields.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).
Selection and/or peer review under responsibility of American Applied Science Research Institute

Keywords: Automotive distributed embedded systems, TTP, Agent, O-MaSE methodology, Simulation

1. Introduction

The increasing of safety real time control applications in the automotive field led to the creation of the time triggered protocol (TTP) [6]. The latter is a family of field bus protocols based on the Time-Triggered Architecture (TTA). TTA is a framework to develop distributed real time applications. It provides a logical bus connecting the host computers that implement the chosen application, and a set of services to organize the application in a fault-tolerant manner. Each host computer attaches to the system through a TTA controller;

* Tel.: +0-213-699-859663

E-mail address: fateh_boutekkekouk@yahoo.fr

the combination of a host and its controller is called a node. Nodes communicate over replicated shared media, called channels. TTP uses a Time Division Multiple Access (TDMA) technique to control the access to the communication media to enable collision-free bus allocation. The TDMA strategy permits to each node to periodically utilize the full transmission capacity of the bus for some fixed amount of time called a slot. A TDMA round is defined as a sequence of slots. The number of different TDMA rounds determines the length of a cluster cycle. Generally, TTP includes a set of basic services such as the startup, the membership, the clock synchronization, the messages sending, reception and errors detection and correction. In this paper, we investigate the idea of using the agent paradigm to design and simulate the TTP services. Since there is a big similarity between multi-agent systems and distributed embedded systems, it seems very appropriate to model a distributed embedded system as a multi-agent system. The impetus of this is to exploit the agent paradigm capabilities especially the complexity, the intelligence, the planning, and the adaptability aspects. In order to accomplish our objective, we followed the O-MaSE methodology to design the multi-agent TTP system since it provides many models that cover many aspects such as the organizational, the functional, the structural and the behavioral aspects. Of course we integrated the ontology aspect which is absent in the O-MaSE methodology. We used the JADE platform to implement and simulate our proposed system. The paper is organized as follows: section two reviews briefly the related work. Section three puts the light on the TTA. Section four describes the conceptual models of the basic services using the O-MaSE methodology. Section five presents our implementation using JADA platform before the conclusion.

2. Related work

In this section, we try to present briefly some pertinent works on TTP modeling and simulation. The work in [2] applies the object-oriented paradigm to design and implement a software simulator for the official TTP/C specification. The main methods and tools that were utilized during the development of this project were the object-oriented development process, a range of tools like the JUnit unit testing suite, and the Rational Rose UML modeling tool respectively. The author in [5] presents the SIDERA (SIMulation model for Dependable Real-time Architectures) for the simulation of various real-time protocol services like system startup, communication, clock synchronization, membership service, protocol error detection and handling. They provide a case study of the clock synchronization in distributed real time systems. According to the literature, we can state that there is a lack of works targeting the simulation of the TTP services using the agent paradigm. For this reason, we try in this work to show how we can exploit efficiently the agent paradigm for TTP basic services modeling and simulation.

3. The Time Triggered Architecture (TTA)

TTA is composed of one or more clusters, each cluster is composed of a set of interconnected nodes via a replicate shared media, each node is an autonomous entity which executes a part of the distributed application and exchanges messages with others nodes. The communication between the clusters is realized by the gateway. Each node consists of the host computer which runs the application and a communication controller which is based on the TTP protocol, the communication controller part which is composed of the protocol processor, the Message Descriptor List (MEDL), and the bus guardian, the interface between the controller and the host part is the Interface Communication Network (CNI), and the interface between the controller and the communication bus is the Logical Line Interface (LLI) [2].

4. Conceptual models of the TTP basic services

The TTP is a complex protocol for which the functional and oriented object methods are not able to deal with all aspects of this protocol. As mentioned before, we have been used the O-MaSE (Organization-based Multi-agent Systems Engineering) methodology [3] which is an extended version of MaSE that allows the design of the multi-agent organization. The main O-MaSE Models used in our MAS are, the goal Model, the organization model, the Role Model, the plan Model and the protocol Model.

4.1. The goal model

The main goal of TTP protocol is to guarantee a fault tolerant communication for the safety critical real time automotive applications. This global goal is noted Goal0. The later can be divided into the communication between nodes (Goal1), the fault tolerate communication (Goal2), and the time real communication (Goal3). Goal2 can be divided into the sub goals: the node fault tolerance (Goal2.1) and the communication medium fault tolerance (Goal2.2). Figure 1 shows a goal model of the TTP protocol.

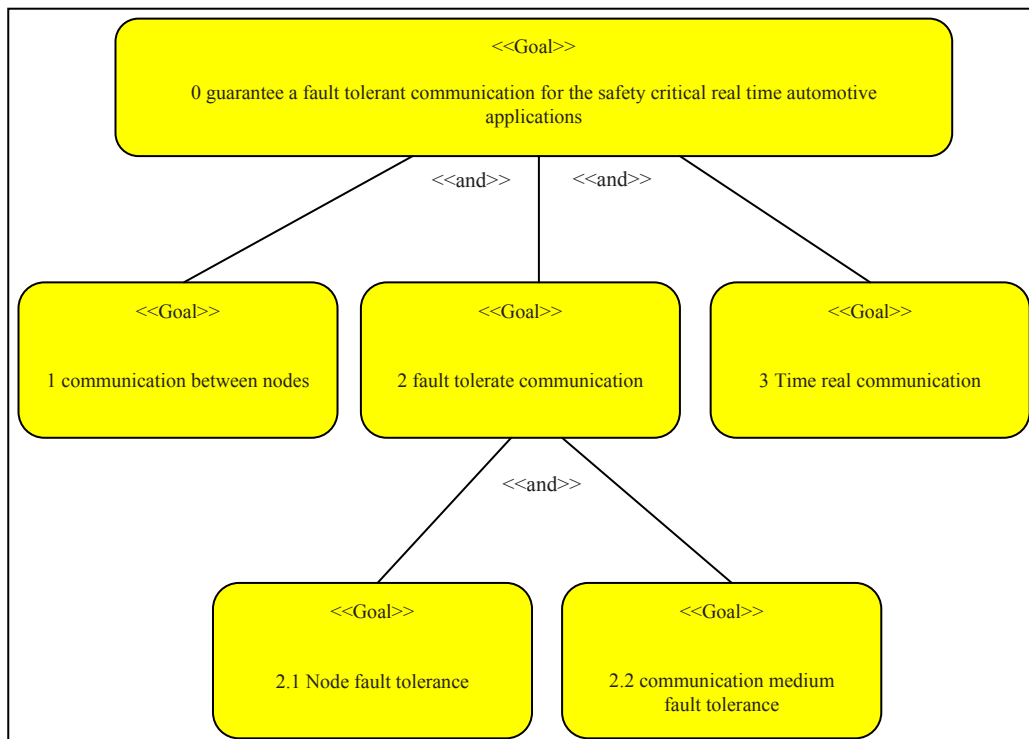


Fig. 1. Goal model of the TTP protocol

4.2. The organization model

The aim of this model is to identify the system (the organization) interfaces with the external actors. There are two abstraction levels of the organization in our proposed simulation: the macro organization that represents the cluster; the external actor in this level is the main simulator and the micro organization that

represents the node of the cluster; the external actors in this level are the communication medium, the scheduler and the simulator. Figure 2 shows a micro level Organization Model of TTA node.

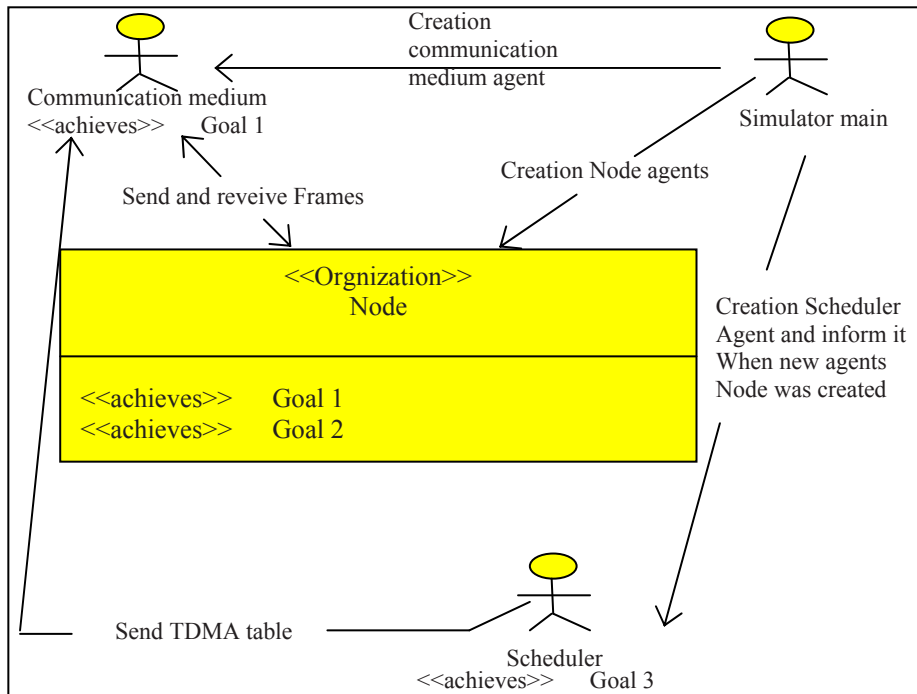


Fig. 2. Micro level Organization Model of TTA Node

4.3. The role model

This diagram represents the roles that can be played by the system agents. Each leaf goal in the goal model must be assigned to one or more roles in the role model that can achieve it. A role may achieve multiple leaf goals. We identified seven roles: transmission service to achieve the Goal1, Membership, clock synchronization, startup system and control access to the communication medium to achieve the Goal2.1, redundancy mechanism to achieve the Goal 2.1 and Goal 2.2, a determinist scheduling to achieve the Goal3. Figure 3 shows a role model of the TTP protocol.

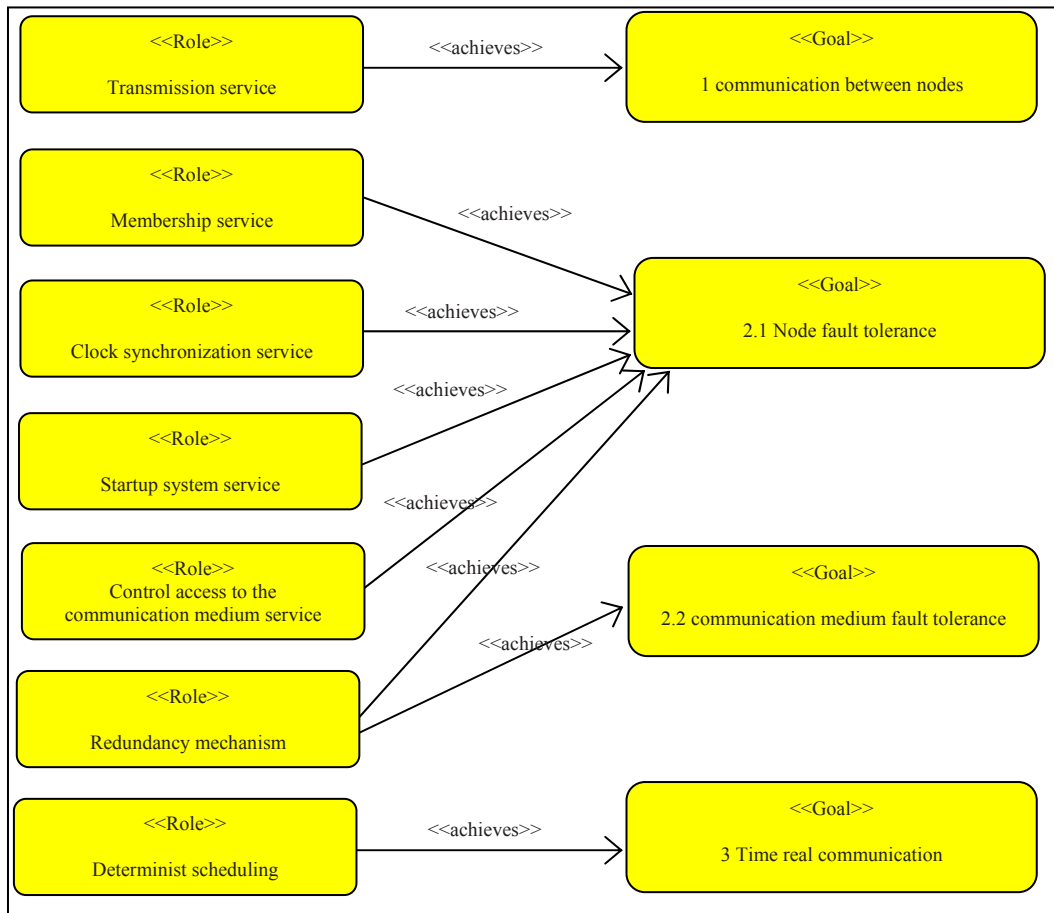


Fig. 3. Role model of TTP protocol

4.4 The Agent Class model

This model represents the system agent classes. Each role in the role model must be assigned to one or more agent classes that can play it. An agent class may play multiple roles. We identified three agent classes to play transmission service role: communication medium agent (we proposed that the medium communication is a bus), CNI interface agent and LLI interface agent. Membership agent to play the Membership role, synchronization agent to play the clock synchronization agent, protocol processor agent to play the startup system role, the TDMA and bus guardian agents to play the control access to the communication medium role, the scheduler agent to play the determinist scheduling role. The role redundancy mechanism is played by instantiate two agents of bus agent class and for each other’s agent classes. In the official version of the TTP protocol, the table TDMA is created by the developer of the system. It is a predefined static table. In our simulation, we proposed an automatic TDMA table creation by the scheduler agent. The simulator main informs the scheduler agent when it creates a new node. The scheduler agent must add a new node in this table and sends it to all cluster nodes via the communication bus agent. The scheduler has always a first slot of the round to send the new TDMA table.

4.5 The Agent Plan Model

The plan agent represents an algorithm agent for achieving his specific goal. Again, because there are nine different agents defined in the agent class model, we have to develop at least nine agent plan models; one for each agent. In the next of this paragraph we present only the plan model of the Bus communication agent, the startup agent, and the Membership agent. The medium communication agent must wait for frames reception. When the frame is received, this agent broadcasts the frame to all cluster nodes. Figure 4 shows the medium agent plan model. The startup agent plan model represents the general startup system strategy. Figure 5 shows the startup agent plan model. The Membership agent plan model represents the membership service phases.

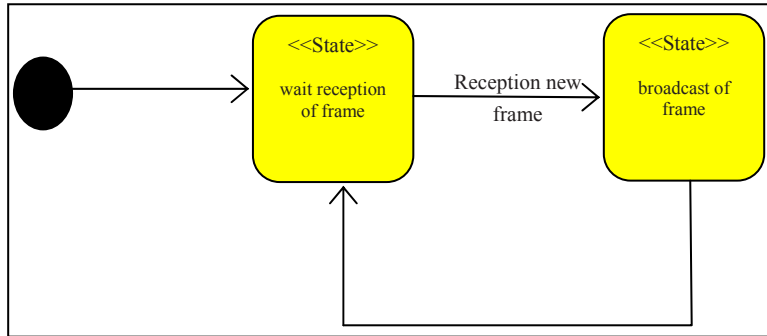


Fig. 4. Medium agent plan model

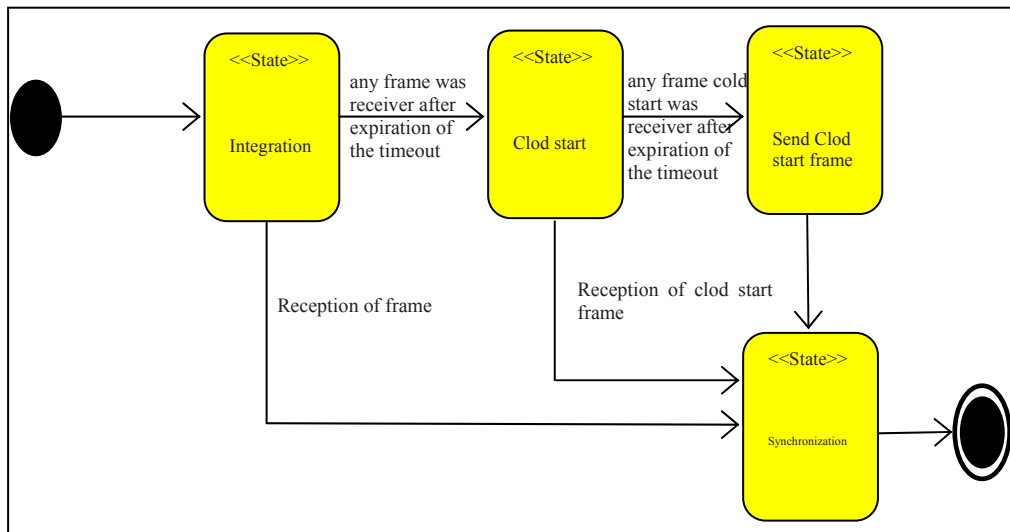


Fig. 5. Startup agent plan model

4.6 The Protocol models

This model defines the protocol in terms of messages passed between agents of node and the external node agents. We proposed two protocol models: the send protocol and the reception protocol models. In the next of

this paragraph we present only the send protocol model. This protocol begins when the TDMA agent informs the Protocol processor and the Bus guardian agents that the current slot is assigned to their node to start the sending of a frame. When the protocol processor receives this message it sends a request message to the CNI interface agent and waits for the received message. When this later is received, the processor protocol sends a request membership vector message to the membership agent, and waits for the membership vector. When this later is received, the protocol processor sends a clock synchronization message to the synchronization agent and waits for the clock current value. When this message is received, the protocol processor integrates this control information with the data of the frame and sends it to the LLI Interface agent. This later in turns sends a request authorization message to the bus guardian agent and waits. Finally, when the authorization message is received, the LLI interface sends a frame to the communication medium agent.

5. Implementation

We have used the JADE (Java Agent Development framework) [1] platform to implement our Multi-agent TTP protocol services. We choose JADE because it matches well O-MaSE methodology aspects such as the behavior of agent, the possibility of creating new communication protocols, the possibility of defining a new ontology. We integrated the Jade with Java eclipse to create our graphical interface and exploit the JADE platform notation to implement the TTP cluster organization, and the container JADE to implement the node organization. We proposed to define a new ontology that provides an automatic interpretation of the frame fields and used the JADE language to automatically convert the format of exchanged messages between agents.

6. Conclusion and perspectives

In this paper, we investigated the idea of using the agent paradigm to model and simulate the basic services of the TTP protocol which is based on the Time-Triggered Architecture (TTA). In our case, we followed the O-MaSE methodology to build the goal, the organization, the Role, the plan and the protocol Models. We used the JADE platform to define the frame ontology and simulate our TTP multi-agent system. As a perspective, we plan to model some automotive protocols (for instance CAN) which are based on the Event-Triggered Architecture (ETA) and the hybrid architecture (for instance FlexRay) using the agent paradigm.

References

- [1] Bellifemine F, Caire G, Trucco T, Rimassa G. JADE Programmer's guide. JADE 2.44, September, 2001.
- [2] Bradbury D. Simulation of a Time Triggered Protocol. Honour Thesis, Basser Department of Computer Science, Sydney University, 2000.
- [3] DeLoach S. A. Developing a multiagent conference management system using the O-MaSE process framework. AOSE'07 Proceedings of the 8th international conference on Agent-oriented software engineering VIII Pages 168-181, 2007.
- [4] Godary K. Validation temporelle de réseaux embarqués critiques et fiables pour l'automobile. Thèse d'état présentée devant l'Institut National des Sciences Appliquées de Lyon, France, Novembre, 2004.
- [5] Hanzlik A. SIDERA - a Simulation Model for Time-Triggered Distributed Real-Time Systems. International Review on Computers and Software (IRECOS), 2006.
- [6] Kopetz H, Grünsteidl G. TTP – A Protocol for Fault Tolerant Real-Time systems. IEEE Computer, pages 14-23, January, 1994.