

Artículo de investigación

Modeling and simulation of agent-oriented networks of internet of things

Modelado y simulación de redes orientadas a agentes de internet de cosas

Modelagem e simulação de redes orientadas a agentes da internet das coisas

Recibido: 9 de abril de 2018. Aceptado: 30 de mayo de 2018

Escrito por:

Bahareh Adineh ^{63*}

Abstract

The concept of the Internet of objects of the world is that these interconnected objects collect and analyze information on a regular basis and use it to begin action, plan, manage, and decide. Internet of things is a new evolution of the Internet, they offer more objects and they create their intelligence by creating or empowering decisions about the reality that information communication can have together and they have access to information that is aggregated by other objects or they introduce the components of complex services. The complexity of our industrial and economic software systems requires system modeling techniques to support reliable and expandable design. Therefore the purpose of this paper is to model and simulate object oriented Internet networks. In this paper, the G-net model is a kind of Petri net, and is used to support system modeling based on a set of independent modules with low interdependence. We have modified the base G-net model to define a so-called operating-oriented G-net. This model is considered as a general model for the design of the operating system.

Keywords: modeling, object oriented Internet, G-net

Resumen

El concepto de Internet de las cosas es que estos objetos interconectados recopilan y analizan información de forma regular y la utilizan para comenzar a actuar, planificar, gestionar y decidir. Internet de las cosas es una nueva evolución de Internet, ofrecen más objetos y crean su inteligencia creando o potenciando decisiones sobre la realidad que la comunicación de la información puede tener en conjunto y tienen acceso a la información que se agrega por otros objetos o introducen el componentes de servicios complejos. La complejidad de nuestros sistemas de software industriales y económicos requiere técnicas de modelado de sistemas para admitir un diseño confiable y ampliable. Por lo tanto, el propósito de este documento es modelar y simular redes de Internet orientadas a objetos. En este documento, el modelo G-net es una especie de red de Petri, y se utiliza para soportar modelos de sistemas basados en un conjunto de módulos independientes con baja interdependencia. Hemos modificado el modelo base G-net para definir una llamada G-net orientada a la operación. Este modelo se considera como un modelo general para el diseño del sistema operativo.

Palabras clave: modelado, Internet orientado a objetos, G-net

⁶³ MSc Software Engineering, Department of computer engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. E-mail: bahareadine9696@gmail.com



Resumo

O conceito da Internet das Coisas é que esses objetos interconectados coletam e analisam informações regularmente e usam-nas para começar a agir, planejar, gerenciar e decidir. Internet das coisas é uma nova evolução da Internet, eles oferecem mais itens e criar a sua inteligência, criando ou reforçando as decisões sobre a realidade que a comunicação de informações podem ser juntos e têm acesso à informação adicionada por outros objetos ou introduzir componentes de serviço complexos. A complexidade de nossos sistemas de software industriais e econômicos exige técnicas de modelagem de sistemas para suportar um design confiável e expansível. Portanto, o objetivo deste documento é modelar e simular redes de Internet orientadas a objeto. Neste documento, o modelo G-net é um tipo de rede de Petri e é usado para suportar modelos de sistema baseados em um conjunto de módulos independentes com baixa interdependência. Modificamos o modelo base G-net para definir uma chamada G-net orientada para a operação. Esse modelo é considerado como um modelo geral para o design do sistema operacional.

Palavras-chave: modelagem, Internet orientada a objetos, G-net

Introduction

Imagine a world in which billions of objects interact with each other over the public or private network protocols and share information. The concept of the Internet of objects of the world is that these interconnected objects collect and analyze information on a regular basis, and they use the planning, management, and decision-making process to begin operations.

Internet of things is a concept that takes into account the presence of diverse objects/ things with wireless and wired connections and unique and empowered interactive mapping designs and these objects/ things work together to create new applications/ services and achieve common goals. Development challenges are to create a smart and large world. The world that in a real, digital and virtual way all converges to create smart environments that creates smarter areas of energy, transportation, cities and many other areas. The goal of object Internet is empowering objects to connect in every time and place with everything or everybody that uses every path, network or service ideally. The Internet of things is a new evolution of the Internet, they offer more objects and they introduce their intelligence more by creating or empowering decisions about the reality that information communication can have with each other, and access to information that is aggregated by other objects, or they are complex components of the service (Vermesan et al, 2013).

The Internet of things is a network of physical objects that includes embedded technology for communicating and sensing or interacting with their own internal states or external environment and the intersection of wireless

efficient protocols, improved sensors, cheaper processors and a group of startups and companies for the necessary management and application software which eventually all these concepts create the main route of Internet of objects. The number of devices connected to the Internet surpassed the number of people on earth in 2011. And by 2020 the number of connected Internet devices is expected to be between 26 billion \$ and 50 billion \$. For each computer or desktop phone connected to the Internet, there will be 5 or 10 types of devices (gadgets) sold with native internet connections (Raymond, 2014).

Internet of things in general refers to many things around us from washing machines, refrigerators, televisions or ventilation systems, home and office lighting that are connected to the Internet and can be controlled and managed by apps in smartphones and tablets. The Internet of Things has also been very successful for many businesses and stores, and has shown that a clear future for this technology could be realized in the field of commerce. At the moment stores that use the Internet of things are not concerned about running out of their products because warehouse intelligent control system knows at any moment the inventory of all products and even makes orders.

Smart agents can be considered as active objects, or objects with intelligence status. However, smart agents differ in terms of communication mechanisms and decision-making capabilities with objects. The result is that the object oriented methodologies are not suitable for modeling agents. Specifically they do not directly support transferring Unicode messages and modeling independent behavior. Therefore, agent-oriented methodologies are

provided as guidance for the analysis and design of the agents. Examples of such works include AAll and Gaia methodologies. Both of these methodologies are object oriented.

Although many efforts have been made to develop agent-based systems, but the research on characteristics and design of such systems is scattered. As operating technologies have emerged as a good solution for industrial and economic applications, the need to ensure the resilience and reliability of such developing systems increases (Carlos et al, 1998).

Network-based approach to agent-oriented design

This section describes the network-based approach to the design of the agent.

G-net model

An accepted principle in software engineering is that a system can be decomposed into a set of independent units, so that each internal module hides its processing functions and modules communicate through well-defined interfaces. The G-net model strongly supports this principle (Perkuisch and Figueiredo, 1997). G-nets are object oriented developments of Petri networks which are actually a graphically defined model for graphic systems. A G-net system consists of a number of G-net, each of which shows a module or an object. A G-net consists of two parts:

1. A specific location called the general switch location;
2. An internal structure

GSP abandoned the module and plays as the only intermediary between the G-net and other modules. IS is a modified Petri network which shows minor design of the module. A GPS of G-net includes a set of G.MS methods which determine services or interfaces provided by the module and also a set of characteristics of G. AS which are status variables. In G.IS or internal structure of G-net Petri network locations show basics, while transfers with arcs show connections or relationships between those basic principles. The basics may define local actions or method calls. Method calls are represented by specific locations called sample switched places. If a basic principle receives a token, it activates and the activated basic principle can be implemented.

The G-net model supports server-client model and it is suitable for object oriented design. However it is not enough to design an agent because the G-net model does not directly support the following features.

1) The intelligent agents in the multi-agent HSP system are usually independently developed by the vendor, so having a common language and following shared protocols is essential.

2) The underlying operating relationship model is usually async and an agent can decide whether or not to perform the actions requested by other agents.

3) Usually agents are designed to determine their behavior based on individual goals, their knowledge and the environment. They can independently initiate their internal or external behavior at any time.

A framework for agent-oriented modeling

In order to support agent-oriented design, G-net must first be developed to support classroom modeling (Xu and Shatz, 2001). This can be done simply by interpreting a G-net as a model of the operating class, it also requires the prototype of the G-net to be defined in two steps.

- Generating an agent G.Aid unit identifier;
- Initialization the mental state of the obtained object agent

In addition at the class level, five specific modules have been introduced to make an agent independent and autonomous. Goal module, plan module, knowledge-base module are based on BDI agent model (Kinney et al, 1996). While environment module is an abstracted model of the environment like external model of agent world. While planner module represent the heart of the agent that can decide whether to ignore a received message, start a new conversation or continue the current conversation. In planner module approved design are archived and goal, plan and knowledge-base modules are updated after each conversation or environment changes. Internal structure of an agent-based G-net includes three parts: Incoming message, outgoing message and private utility. The section of incoming/ outgoing message defines a set of message processing units, which is labeled in figure 1 as action_i and it is used to process incoming/outgoing messages and it can use ISP-type modeling to recall definition methods in private-utility section.



Although both objects and agents use message transmission to communicate with each other, the message transmission for objects is a unitary form of method call while agents differentiate between different types of messages and repeatedly model these messages as a conversation and use complex protocols for conversation. In particular, these messages should satisfy standardized communication that defines the type and content of the message. In addition, the agents analyze these messages and can decide whether to do the requested action. Agent's communication is based on asynchronous message transmission. As the Asynchronous Message Transmission is much more basic than transmitting the synchronous message, we introduce the introduction of a new mechanism that is called the switch location of the message transmission, so as to directly support the transmission of the asynchronous message.

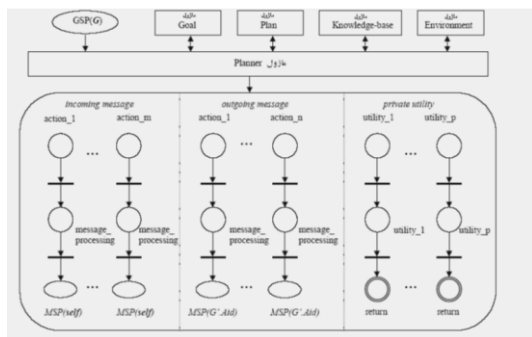


Figure 1- overall model of agent based G-net

The pattern for module planner is shown in figure 2. Goal, plan, knowledge-base, and environment modules are shown in four specific locations, each of them includes tokens that represent a set of goals, a set of beliefs and a model of the environment correspondingly. These four modules are connected via abstract transition to the planner module which is shown in figure 2 with a shaded rectangle. Make-decisions show abstract units of decision making or mental status update. In a more detailed stage than design, abstractions can be maintained in subnets with specific sequences of action for those activities.

Also a new unit is defined in planner module which is called autonomous unit that make an agent independent and autonomous. An autonomous unit includes a sensor, which is fired whenever certain conditions are fulfilled or new events are taken. If abstract transmission of sensor is fired, then the independent unit based

on current mental state of the agent decides whether to start a conversation or simply update its state of the mind. This work is done either by firing the start-a-conversation transfer or the automatic-update transfer after executing any action associated with the new-action location.

Note that the Planner module is purpose-oriented as well as event-oriented since the sensor can be triggered whenever a validated target arrives or a new event occurs. In addition, the Planner module is activated with the message because each time a message arrives, certain actions can begin. A message is displayed as a message token with an internal / external / method tag. A message token with internal tag displays a message that is sent by MSP mechanism or agent to itself or displays a new output message before sending it to other agents. While a message token with an external tag is an incoming message that comes from another agent. In both cases a token with internal/external tag should not be in method call invocation. On the contrary a message token with method tag shows that the token is currently in a method frequency invocation. When an incoming message/ method arrives with an external/ internal tag in its respective token, it is sent to appropriate MPU/Method defined in the agent's internal structure. If this invocation is a method, the method defined in utility method section will run from the internal structure and after that the token will return to the caller method for example an ISP from the caller unit. However, if this is an incoming message, the message is initially processed by an MPU defined in the Incoming message section of the agent's internal structure. The token tag then changes from external to internal before it is returned to the GSP host using MSP itself. Note that we have developed G-net to allow the use of the self key word to refer to the object itself. After the entry of a token in the GSP is labeled as internal, internal transfer can fire after firing make-decision abstraction fire. Note that at this point in time, there are tokens in certain locations of the Goal, Plan, and Knowledge-base, so the bypass transmission is disabled and cannot be fired. Any other essential activity can be performed in the next-action place before the call is canceled or continued. If the current conversation is ignored, the transfer is ignored; otherwise, the transfer continues to fire. If continue transfer is triggered, a new output in a form of a token with internal tag is sent to an appropriate MPU in output message section of internal structure of the agent. After the message

is processed by the MPU, the message is sent to the receiving agent using the mechanism and the token tag of the message varies from internal to external. In both cases, a token will be placed in the update-goal / plan / lb location, allowing the abstraction transfer of the update to fire. As a result, the Goal, Plan and Knowledge-base modules, if needed, are updated, and the agent's mental state may change.

To ensure that all decisions are made after the agent's mental status, for example, the last values in the Goal, Plan, and Knowledge-base modules, and similarly to ensure that the sensor always maintains the last state of the agent's mental state, we introduce the syn sync unit, which is modeled as a marked location with a conventional token.

References

- Iglesias, C., Garrigo, M., Centeno-Gonzalez, J. (1998). A Survey of Agent-Oriented Methodologies. Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Language. pp. 317-330.
- Kinny, D., Georgeff, M., and Rao, A. (1996). A Methodology and Modeling Technique for Systems of BDI Agents. IN Van de Velde and J. W. Perram, editors, Agents Breaking Away: Proceeding of the Seventh European Workshop on Modeling Autonomous Agents in a Multi-Agent World, 56-71, Springer-Verlag: Berlin, Germany.
- Perkuisch, A. and Figueiredo, J. (1997) G-net: A Petri Net Based Approach for Logical and Timing Analysis of Coplex Software Systems," Journal of Systems and Software, 39(1): 39-59, 1997.
- Raymond J. & Associates. (2014). The Internet of Things -A Study in Hype, Reality, Disruption, and Growth", online at <http://sitic.org/wp-content/uploads/The-Internet-ofThings-A-Studyin-Hype-Reality-Disruption-and-Growth.pdf>.
- Roch, S. and Strake, P. (1999) INA: Integrated Net Analyzer, Version 2.2, Humboldt-Universitat zu Berlin Institut fur Informatik.
- Vermesan, O, Friess, P., Guillemin, P., Sundmaecker, H. et al. (2013). Internet of Things Strategic Research and Innovation Agenda. Chapter 2 in Internet of Things – Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers. ISBN 978-87-92982-73-5
- Wooldrige, M. Jennings, M., and Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design, "International Journal of Autonomous Agents and Multi-Agent Systems, 3(3). pp. 285-312.
- Xu, H. and Shatz, M. (2001). A Framework for Modeling Agent- Oriented Software," To appear in the Proc. Of the 21st International Conference on Distributed Computing Systems (ICDCS). Phoenix, Arizona.

