SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

Mobile Agent Life Cycle Demystified using Formal Method

IMIANVAN Anthony Agboizebeta[1] and AKINYOKUN Oluwole Charles[2]
[1]*Department of Computer Science, University of Benin, Benin City. Nigeria*
[2]*Department of Computer Science, Federal University of Technology, Akure, Nigeria.*
tonyvanni@uniben.edu

ABSTRACT

Underlying technique for mobile agent development is often mystified. Existing research sometimes ignore unveiling the details of the mobility and autonomy of the agent system. This paper exposes using formal methods the technique of a mobile agent system using a life cycle. The system proposed will serve as takeoff springboard for mobile agent developers.

Keywords: Mobile Agent, Formal Method, Z-notations, Mobile Agent Life Cycle.

1 Introduction

Mobile agents are autonomous and intelligent programs that are capable of moving through a network, searching for and interacting with the resources on behalf of the network administrator. Mobile agent is an executive program that can migrate at times of its own choosing from one machine to another in a network. This means that a mobile agent is 'free' to travel to any place in the network. It can execute without requiring a link with or being controlled from the originating location (Imianvan, 2009).

Mobile agent could be activated and launched from one computer to another for the purpose of autonomously searching for and interacting with network resources on the network administrator's behalf (Imianvan and Akinyokun 2014). A conscious effort at developing a mobile agent for the assessment and evaluation of computer networks with emphasis on throughput, utilization and availability have been proposed in (Aderounmu, 2001). Mobile agent technology is useful in electronic commerce transactions, distributed information retrieval, and network management (Huy et al., 2005; Imianvan, 2009; Weina and Gaoyuan 2011; Djamel et al., 2012).

The following are examples of some existing mobile agent systems. The items inscribed in the brackets are the individual, or institution or organizations that developed them.

a. Agentspace (Alberto Silva).
b. Agent TCL (Dartmouth).
c. Aglets (IBM).
d. D' Agents (Dartmouth College).
e. Dicovery (University of Maryland).
f. JATLite (Standford University).
g. MARS (University of Modena).
h. Messengers (University of California).
i. Mobile Agent Platform (Universita' di Catania).
j. TACOMA: Tromso and Cornell Moving agent (University of Tromso).

DOI: 10.14738/tmlai.32.1025
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1025

Mobile agent like every other intelligent agent has the following features:

a. Autonomy.
b. Adaptiveness.
c. Collaborative behavior.
d. Character.

A specification written and approved in accordance with established (mathematical) notations is a formal specification. Z ('zed'), for instance is a formal notation based on set algebra and predicate calculus for the specification of computing systems. Z specification of systems employs the power of discrete mathematics. The Z notation is useful to organize and communicate thoughts within a design team (Diller 1994; Spivey 1998).

Since a formal specification is precise, if such a specification is wrong, it is easier to tell where it is wrong and correct it. Using formal notation increases the understanding of the operation of a system especially early in a design. It helps to organize the thoughts of a designer, making clearer, simpler designs possible. Formal specification provides a check that the system will behave as expected by the designer. The use of formal methods can help to explore design choices. Such methods aid the design team in reasoning about the operations of the system in clear terms before and during its implementation (Spivey, 1998).

This paper provides an attempt to demystify using formal methods (Zed notations), the operational life cycle of a mobile agent system.

2 The Mobile Agent Life Cycle

A mobile agent is an artificial life which is capable of birth (creation), survival (launching) and death (disposal). The processes of birth, survival and death are characterized by a sequence of logical steps called the mobile agent life cycle as presented in Figure 1.

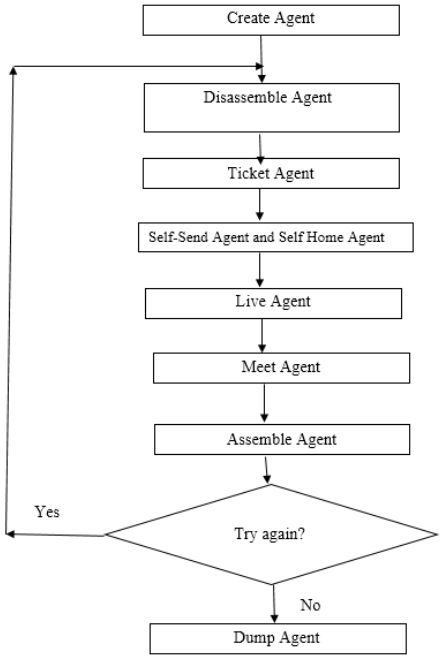

Figure 1: The Mobile Agent Life Cycle

The mobile agent life cycle presented in Figure 1 involve the following series of logical procedures (Imianvan, 2009).

a. CreateAgent.
b. DisassembleAgent.
c. TicketAgent.
d. SelfSendAgent and SelfHomeAgent.
e. LiveAgent.
f. MeetAgent.
g. AssembleAgent.
h. DumpAgent.

The creation of the mobile agent involves developing the functionality of the system and then adding it to the universal set of agent as presented in Figure 2.

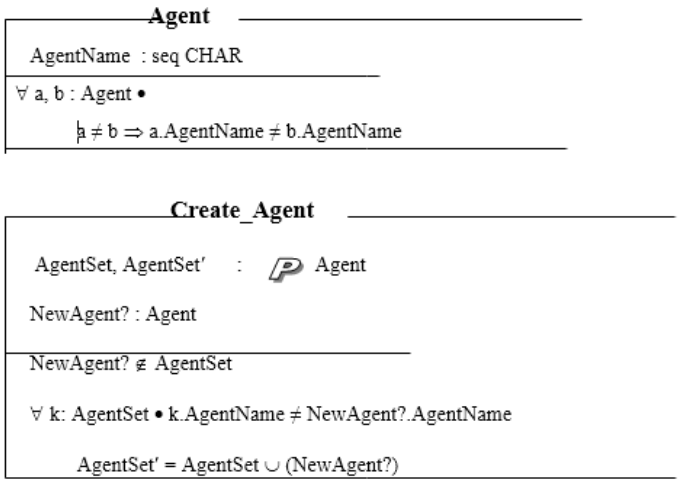

Figure 2: Formal Specification of the CREATE Agent Process

At the source, the mobile agent is decomposed into units that are transportable to the target workstations. The formal specification of the disassembling of the mobile agent is presented in the Z Schema of Figure 3.

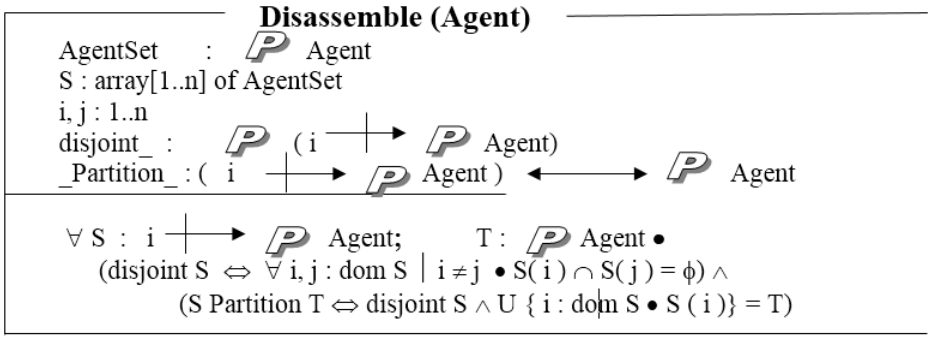

Figure 3: Z Schema Specification for Mobile Agent Disassembling.

Operational procedure of TicketAgent component of mobile agent system is presented in Figure 4.

```
1.  TicketAgent(agent, v, e)
2.      // agent is to be deployed to workstations.
3.      // v is set of vertices of the graph representing workstations.
4.      // e is set of edges of the graph representing distance between workstations.
5.      TargetFound ← false
6.      j ← 1
7.      loop i from 1 to number of workstations
8.          TeleAddress wi      // tag target addresses.
9.          LookUpPlaces wi     // used to autonomously view targets workstations.
10.         //  wi is target workstations
11.         if wi  not in v
12.             v ←  v U {wi}
13.         end if
14.         copy(v)     // collect or copy target address
15.         j ← j + 1
16.         if wi ≠ wj edgei,j  ← distance wj – distance wi
17.             e   ←  e U edgei,j
18.         end if
19.         copy(e)   // collect or copy distance between targets
20.         TargetFound  ← true
21.         Ticket (agent, wi)
22.     end loop
23. end TicketAgent
```

Figure 4:TicketAgent Algorithm

The following are sequence of Telescript commands specification to notify an agent (for example, Bandwidth Agent) of target workstations.

Teleaddress addr := here@LookUpPlaces.Address.Copy();

Ticket(BandwidthAgent, addr):

SelfHomeAgent operates as follows:

a. Use SELF-command to activate autonomy.
b. Use GO command to activate the agent movement.
c. Use HOME command to activate return home.

LiveAgent uses the game of life algorithm presented in (Akinyokun, 1997) to account for the resource of the computer network environment.

MeetAgent operates using the command.

agentToMeet := here@MeetingPlace.Meet (aPetition, nil);

The agent will normally interact with the host operating system of the target and its appendages or utility programs for network monitor and cyber clock for the purpose of assessing and evaluating network resource. The results obtained by mobile agent after a successful visit to a set of target workstations are assembled using the specification of Figure 5 for the purpose of reporting them for external analysis, interpretation, policy formulation and decision making.

ASSEMBLE(SubAgent)

Agent ::= BandwidthEvaluator
AgentSet, AgentSet′ : ℙ Agent
SubAgent : array[1..n] of AgentSet
i : 1..n

∀ SubAgent : i → ℙ Agent
AgentSet′ ← AgentSet ∪ {SubAgent(i)}

Figure 5:Z Schema description of ASSEMBLE Agent

The disposal of the mobile agent simply means removing the mobile agent from the universal set of agents. The process of the mobile agent disposal is presented in Figure 6. Telescript command used to dispose agent is DUMP.

Agent

AgentName : seq CHAR

∀a, b : Agent • a ≠ b ⇒ a.AgentName ≠ b.AgentName

DisposeAgent

AgentSet, AgentSet′ : ℙ Agent

Remove? : Agent

Remove? ∈ Agentset
∃₁ k : Agentset • k.AgentName = Remove?.AgentName
AgentSet′ = AgentSet \ {Remove?}

Figure 6: Specification of the Mobile Agent Disposal Process

3 Conclusion

Formal method (Zed notations) has been used as the operational engine for demystification of the life cycle of a mobile agent. The use of Formal Method is one more weapon in the armoury against making design mistakes. It is hoped that developers of mobile agent system will find the scheme useful.

REFERENCES

[1]. Imianvan Anthony Agboizebeta and Akinyokun Oluwole Charles (2014),

[2]. Formal Characterization of a Mobile Agent Operational Environment, Journal of the Nigeria Association of Mathematical Physics, Published by Nigeria Association of Mathematical Physics, Volume 26, March 2014, pages 467 – 475

[3]. Imianvan Anthony Agboizebeta (2009), "Development of a Mobile Agent System for Evaluating the Use of Bandwidth in a Computer Network", PhD Thesis, Federal University of Technology, Akure, Ondo State. Nigeria.

[4]. Aderounmu G. A. (2001), "Development of an intelligent Mobile Agent for Computer Network Performance Management", PhD Thesis, Obafemi awolowo University, Ile-Ife, Nigeria.

[5]. Huy Hoang To, Shonali Krishnaswamy, and Bala Srinivasan (2005), Mobile Agents for Network Management: When and When Not! , ACM Syposium on Applied Computing.

[6]. Akinyokun O. C. (1997), "Catching and Using the Virus", The Journal of the Institute of the Management of Information Systems (IMIS), London, United Kingdom, Vol. 7, No. 6, Pages 12-17.

[7]. Weina He and Gaoyuan Liu (2011), The application of mobile agent in e-commerce, 3rd International Conference on Advanced Computer Control (ICACC), 2011, HARBIN.

[8]. Djamel Eddine Menacer, Habiba Drias, Christophe Sibertin-Blanc (2012), MP-IR: Market-Oriented Mobile Agents System for Distributed Information Retrieval, Advances in Intelligent and Soft Computing, Volume 122, pages 379-390, 2012.

[9]. Diller A., (1994), Z : An Introduction to Formal Methods, (2nd edition), John Wiley and Sons

[10]. Spivey J. M. (1998), "The Z notation: A Reference Manual", Prentice Hall International, United Kingdom.

