CHAPTER 6

FAULT TOLERANT MOBILE AGENT SYSTEM

6.1 INTRODUCTION

A mobile agent (or simply agent) is a self-contained software element responsible for executing a programmatic process, which is capable of autonomously migrating through a network. An agent migrates in a distributed environment between logical “places” referred here to as agencies. When an agent migrates, its execution is suspended at the original agency, the agent is transported (i.e. program code, data, execution state and control information) to another agency in the distributed environment, and, after being re-instantiated at the new agency, the agent resumes execution Raimundo et al (2001).

The IEEE defines fault tolerance as (IEEE 1983):

(1) The ability of a system or component to continue normal operation despite the presence of hardware or software fault. Or
(2) The number of faults a system or component can withstand before normal operation is impaired. Or
(3) Pertaining to the study of errors, faults, and failures, and methods for enabling systems to continue normal operation in the presence of fault.

In the mobile agent system, the absence of execution of mobile agent can be considered as fault. In order to make the mobile agent system as a fault tolerant, the system has to continue to provide the normal operation when the failure of one or more mobile agent is not performing the assigned task due to
loss. For a reliable mobile agent application like mobile banking, it must be fault tolerant.

The basic idea of the volunteer algorithm is explained with the following steps:

1) if an agent is not currently committed to execute any plan, it will try to find a plan in which it can play a role

2) the agent will find out whether that role has been assigned to any other agents,

3) if that role has not been assigned to any other agent, the agent will volunteer to play that role and send a message to everyone else in the team Yu Zhang and Jianwen Yin (2006).

The Extended Volunteer Algorithm (XVA) (Marikkannu 2011) based on the volunteer algorithm is valid only inside the Context-Aware Environment. The additionally assigned roles are discarded once the mobile agent gets out of the environment. So the Mobile Agent, when it moves out of the environment can carry only the data that is concerned with its basic role. For applications such as mobile banking where the mobile host is not fixed and also mobile banking transaction must be exactly once, XVA is not suitable.

The mobile group concept was introduced to provide synchronization among a group of mobile agents Silva and Macedo (2000). A mobile group is an extension of the traditional concept of a process group that can directly support migrating processes as members of the group. With mobile groups, a migrating process has the ability to change its location in the distributed environment while belonging to a group. Mobile groups also provide message delivery guarantees and a sort of virtual synchrony. However, mobile groups provide these guarantees despite the mobility of their members. Furthermore,
they make process mobility not only visible for the group, but also consistently ordered with other group actions (such as process crashes, joins, and leaves). Due to the guarantees that they provide, mobile groups represent a suitable abstraction for the development of reliable agent-based applications Flavio Assis Silva and Raimundo Macedo (2001).

In a mobile group, agent communicates with each other by exchanging messages which are multicasted to the whole group. In order to preserve consistency among the state of group members, the group communication protocols guarantee certain properties such as atomic delivery (either all processes deliver a message or no one delivers it), message ordering guarantees (e.g., causal and total), and a sort of virtual synchrony where modifications on the group membership (caused by events such as process crashes and joins, etc.) are consistently ordered with respect to message delivery.

In mobile groups, a agent can migrate while belonging to the group. The mobile group system recognizes the movement action of the agent and considers movement as a single group operation.

A concept for coordinating groups of agents is described, but which is not fault tolerant Baumann and Radouniklis (1997). Approaches for reliable message delivery to mobile agents are proposed, but failures are not tolerated Murphy and Picco (1999). In another approach failures are considered, but the approach supports only unicast (peer-to-peer communication) Ranganathan et al (2000). A concept of mobile group is discussed with only fixed host, it does not deal with the mobile group for the mobile devices Flavio Assis Silva and Raimundo Macedo (2001).
6.2 FAULT TOLERANT ALGORITHM

The set of agents used in mobile banking will form a mobile group. Each agent represents a process of the group like authentication, balance checking, bill payment, fund transfer etc. If any agent fails, the coordinator create new agents to do the job of failed ones. Also, since mobile groups provide a consistent view of process locations, when an agent decides to visit a new location, it will not move to locations it knows were already visited by another agent of the group. If the coordinator fails, the other agents in the mobile group will eventually know about this. To select the new coordinator, the agent with the greatest process id in the group concept is used Flavio Assis Silva and Raimundo Macedo (2001). This concept may not always work properly. A suitable coordinator algorithm is needed to make the system more reliable.

A new coordinator can be elected by applying volunteer algorithm on the set of operational agents. The agency informs the failure of the coordinator to the agent directory. To select the new coordinator in case of failure, a modified volunteer based coordinator selection fault tolerant algorithm is proposed.

6.2.1 Volunteer Based Coordinator Selection Fault Tolerant Algorithm

Step 1: The agent directory sends a ready_to_accept message to all the agents in the mobile process group.

Step 2: If any agent responds with I_am_ready message within a specific time interval t then

    Set True to the ready_to_work message for that agent

Else
Set False to the ready_to_work message for agents not responded within time interval t
Step 3: If no_of_roles < 2 with low priority and ready_to_work = True then
   3a. Assign the additional role as coordinator
   3b. Add 1 to the no_of_roles
   3c. Reset the value of ready_to_work
Step 4: If there is no agent with the above criteria then
       Create new agent and assign it as the new coordinator

6.3 FAULT TOLERANT TRADE-OFF ANALYSIS

This section discusses about a trade-off analysis in order to determine the coordinator selection intervals satisfying a response time goal with the minimum possible cost Panagiotis Katsaros et al (2007). As shown in Table 6.1, a fault tolerant coordinator is selected after every 10 ms, 25 ms, 50 ms and 100 ms and the respective response time of the mobile banking system is given. This analysis has to be considered for other applications and also for other environment.
Table 6.1 Fault Tolerant Coordinator Selection At Various Intervals

<table>
<thead>
<tr>
<th>Number of user requests</th>
<th>After 10 ms</th>
<th>After 25 ms</th>
<th>After 50 ms</th>
<th>After 100 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>213</td>
<td>194</td>
<td>185</td>
<td>172</td>
</tr>
<tr>
<td>5</td>
<td>270</td>
<td>253</td>
<td>241</td>
<td>233</td>
</tr>
<tr>
<td>10</td>
<td>292</td>
<td>268</td>
<td>257</td>
<td>243</td>
</tr>
<tr>
<td>15</td>
<td>303</td>
<td>287</td>
<td>279</td>
<td>264</td>
</tr>
<tr>
<td>20</td>
<td>312</td>
<td>298</td>
<td>285</td>
<td>277</td>
</tr>
<tr>
<td>25</td>
<td>324</td>
<td>312</td>
<td>305</td>
<td>394</td>
</tr>
<tr>
<td>30</td>
<td>388</td>
<td>374</td>
<td>362</td>
<td>352</td>
</tr>
</tbody>
</table>

Figure 6.1 Performance Analysis of Fault Tolerant Mobile Banking System
As shown in the Figure 6.1, if the fault tolerant coordinator is selected after every 100 ms, the cost associated with the overhead is minimal. So that the minimum response time goal is achieved. But when the coordinator is failed in between, the cost associated for making the system to be fault tolerant is high. If the fault tolerant coordinator is selected after every 10 ms, the cost associated is high. The response time is also high. Based on the fault tolerant trade-off analysis, the interval for the selection of fault tolerant coordinator has to be decided. Using this interval value, the performance can be increased or decreased.

6.4 MOBILE AGENT APPLICATIONS

6.4.1 Introduction

Simulators are very powerful tools for developing mobile applications; they have unique features which enable them to provide a debugging environment. Based on the studies made by Karl Kurbel et al (2002) about the simulator toolkits available in the market to develop mobile applications, Nokia Mobile Internet Toolkit provides sophisticated features than Ericsson IDE 3.1.1 Application Designer and Openwave Software Development Kit (SDK) Edition 5.0.

6.4.2 Nokia Mobile Internet Toolkit 4.0

Nokia Mobile Browser Simulator 4.0 (NMB 4.0) is a mobile Internet browser SDK designed for use on a mobile device such as a mobile phone or PDA. NMB is the browser found on most Internet enabled Nokia phones. It contains support for Wireless Application Protocol (WAP), eXtensible Hypertext Markup Language – Mobile Profile (XHTML-MP), Cascading Style Sheet (CSS) and Multimedia Message Service (MMS). Mobile browsers are optimized so as to display web content in a most effective way for small
screens on portable devices. This software must be small and efficient to accommodate the low memory capacity and low-bandwidth of wireless handheld devices. A mobile phone simulator acts as a micro browser, which is a small-scale web browser that is designed specially for wireless devices.

6.4.3 Results

6.4.3.1 Registration process

To utilize the mobile banking service provided by the bank, the user has to register themselves with the bank with all relevant details such as customer code, password, name, date of birth, address, email-id, phone number as shown in the Figure 6.2.

Figure 6.2 Mobile Banking Registration Form Screenshot
Figure 6.3 shows the screenshot of the details required for bill payment facilities.
6.4.3.2 User login screen

When the user wants to avail mobile banking facility, he has to enter his login id and pin number or password. The source code and the corresponding login screen is as shown in the Figures 6.4 and 6.5.

![Figure 6.4 Nokia Mobile Internet Toolkit Login Source Code](image-url)
6.4.3.3 Mobile banking transaction

The user can select the transaction to do the required operations like fund transfer from his account number to his friend’s account number, Cheque book request, Bill payment and so on. The Figure 6.6 shows the transaction screenshot.
6.4.3.4 Mobile banking bill payment

The user can pay the electricity bill, telephone bill, mobile bill and insurance policy through mobile. The following Figures 6.7 and 6.8 show the screenshots of bill payment facilities.
Figure 6.7 Mobile banking Electricity Bill Payment screenshot1
6.4.3.5 Mobile banking fund transfer

The user can transfer amount from his account to friend’s account. The Figures 6.9 and 6.10 show the screenshots of fund transfer facility available in mobile banking.
Figure 6.9 Mobile Banking Fund Transfer Screenshot1
6.5 SUMMARY

To summarize, in the fault tolerant mobile banking system based on mobile group, the coordinator is selected at various intervals like after every 10 ms, 25 ms, 50 ms and 100 ms. A fault tolerant trade off analysis is used to select the interval. So that the response time goal is achieved.