A Novel Strategy Approach for Agent-based Resource Management System

[1]STEVEN K.C. LO, [2]LU WANG, [3]HUAN-CHAO KEH, and [3]YI-HONG LIN
[1]Department of Information Management
Jinwen University of Science & Technology
NO. 99,An-Chung Rd. Hsin-Tien, Taipei County, TAIWAN 23154, R.O.C.
*E-mail：kclo@just.edu.tw
[2]Graduate Institute of Management Sciences
Tamkang University
151 Ying-chuan Road, Tamsui, Taipei County, TAIWAN 25137, R.O.C.
[3]Department Of Computer Science and Information Engineering
Tamkang University
151 Ying-chuan Road, Tamsui, Taipei County, TAIWAN 25137, R.O.C.

Abstract: - The purpose of resource management is based on the limit resource and unknown users which retrieve the resources at any time to increase the resource exploitation rate, flexibility and availability. In this study, we propose four types of strategy including Expansion (when user loads are large and available resources are not limiting), Constraint (when user loads are large and resources are limiting), Modulation (when user loads are small and excess resources are held), and Supervision (when user loads are small and available resources are also small). The four types of strategy are abbreviated as MSCE. The MSCE strategy of resource management delivers efficient control and management of available resources, allowing for adaptations to changes in the system environment and changes in user requirements. The Monitor agent is responsible for monitoring the resource management system, controlling the rate of resource consumption while simultaneously keeping track of user loads. By employing the Monitor agent in this fashion, the system's resources have greater availability and their control becomes more flexible and efficient.

Key-Words: - _ Multi-agent; MSCE; Resource Management System

1. Introduction

The meaning of the term "agent" has been considered by a number of scholars to date [2, 3, 19, 24], and the concept has been implemented within a variety of systems [1, 6, 7, 12], including communication systems, coordination systems, and cooperation systems [4, 5, 14]. An agent is a software object specifically designed to process complex tasks under dynamic conditions. An agent is capable of reacting, even in an unstable environment, by automatically executing specific tasks in accordance with predefined parameters, such as those communicated by other agents as part of their behavior.

The purpose of resource management [11, 15, 16] is based on the limit resource and unknown users which retrieve the resources at any time to enhance the resource exploitation rate, flexibility and availability. And, the dynamic resource control becomes more crucial subject to reduce the resources and reuses the resources for human activities. How to achieve this goal is this study research direction and integrates the Multi-agent [17, 20, 23, 25] interoperability mechanism to automatic switch the strategy that is the key success factor (KSF). Hence, in this study, we integrate the Multi-agent interoperability [10] mechanism and applied the resource management strategy to achieve efficient resource management.

The rest of this study is organized by 5 sections. Section 2 describes the strategies of resource management. In this section, we propose the four strategies including Expansion, Constraint, Modulation, and Supervision (MSCE) and introduces the MSCE algorithm to dynamic control the resources. In section 3, we present a Resource Management System to illustrate how the

Multi-agent interoperability mechanism is used and applied the MSCE strategy. Eventually, it summarizes the overall integrated the Multi-agent interoperability mechanism and MSCE strategy into resource management system.

2. The Strategy of Resource Management

Resource management strategies for multi-agent systems tend to be focused upon the flexible and efficient control of resources. The resource pool is a crucial concept to the operation of a multi-agent system; management of the resource pool includes the collection of available resources, including newly created or newly released resources from the control cycle, as depicted in Fig. 1. In this figure, when the system is in the ready state, it retrieves the resources from the resource pool as per the strategy policy, and subsequently employs a queue data structure to control the use of the obtained resources using a first-in first-out (FIFO) sequencing pattern. In addition, the upper and lower bound parameters are used to dynamically and flexibly increase or decrease the quantity of obtained resources.

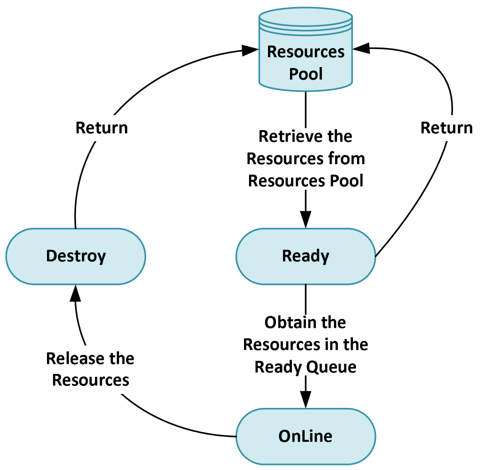


Fig. 1: The resources control cycle

When resources are required, the system obtains them from the Ready queue, which is the first location in the pipe from which it can draw resources. Once retrieved from the Ready queue, the resource's status is set to OnLine. The resource control cycle defines the system's use of resources from initiation to shutdown, performing the functions of Expansion, Constraint, Modulation, and Supervision (MSCE).

Herein, we propose four strategies (autonomous [13, 18], automatic switching according to the volume of available resources and the user load at a given point in time), including Expansion, Constraint, Modulation, and Supervision. Fig. 2 depicts the entry and exit conditions of each strategy upon a 4-quadrant diagram.

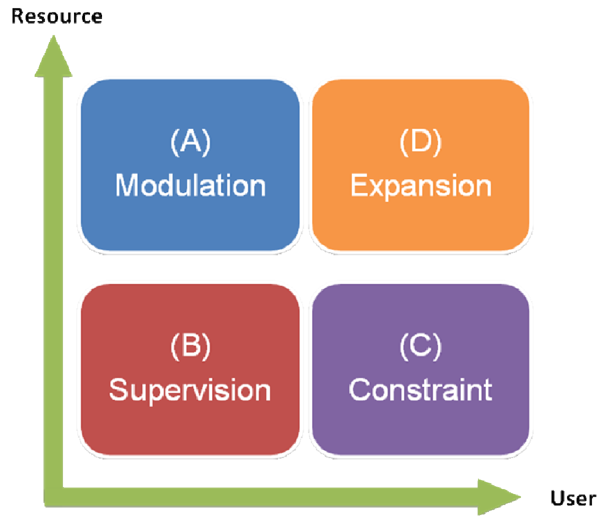


Fig. 2: The strategy diagram

Employing the MSCE approach, the system automatically changes its strategy based upon the user load and available volume of resource. A given process will enter into one of the strategic states from an initial listening state. This is triggered via a communication protocol such as TCP or UDP when a process is first created, as depicted in Fig. 3.

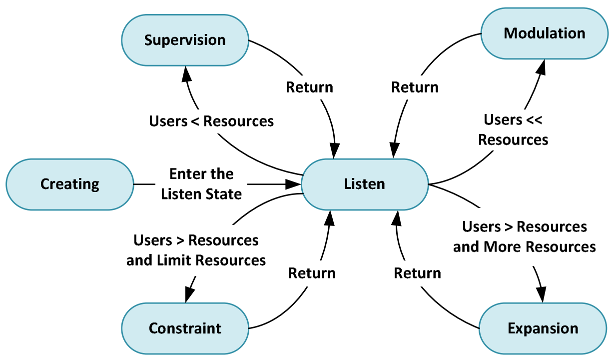


Fig. 3: The MSCE strategy Communication State Diagram

(A).*Modulation*: in the scenario where the volume of available resources is greater than the resource requirements (i.e., the consumption rate is manageable), to the system will, from the Ready state, release extra resources to the resource pool. The returned resource quantities are determined according to predefined policy parameters. The Modulation function, which implements this process, is defined below:

```
Function Modulation (){
    Move Resources;
    Adjust Policy;
}
```

(B).*Supervision*: the Supervision strategy is employed to monitor the user load and resource volumes in increase request scenarios. The Supervision function is defined below:

```
Function Supervision (){
    Monitor User Qty;
    Monitor Resources Qty;
}
```

(C).*Constraint*: in scenarios where the total volume of resources is insufficient to address a user request, the system will manipulate the rate of resource consumption, restricting the performance of the user's activities. The Constraint function is defined below:

```
Function Constraint (){
    Monitor User;
    Adjust Policy;
}
```

(D).*Expansion*: when operating under the Expansion strategy, the system will manage resources that have already been requisitioned by the system, or resources that remain free. When executing this strategy, the resources are maintained at a sufficient level to just satisfy the request, adjusting the upper bound parameter as necessary. The Expansion function is defined as follows:

```
Function Expansion (){
    Add Resources;
    Adjust Policy;
}
```

●MSCE Algorithm:

```
While (true){
    if (User < Resources){
        if (Resources>>User)
            Modulation ();
        else
            Supervision ();
    }
    else {
    if (has more Resources)
        Expansion ();
    else
        Constraint ();
    }
}
Function Modulation (){
    Move Resources;
    Adjust Policy;
}
Function Supervision (){
    Monitor User Qty;
    Monitor Resources Qty;
}
Function Expansion (){
    Add Resources;
    Adjust Policy;
}
Function Constraint (){
    Monitor User;
    Adjust Policy;
}
```

The MSCE algorithm is illustrated the MSCE strategy implementation in more detail. In this MSCE algorithm, the Users and Resources are the key parameters to decide entered particular strategy and control the resources exploitation rate.

●Definition 2.1:

If the resources cannot be separated and user are only capable of retrieving one resource unit. Then the Exploitation rate is calculated as follows:

$$Exploitation\_Rate = \frac{\sum_{i=1}^{m} \text{USER } i}{\sum_{i=1}^{n} \text{RESOURCE } i} \quad (1)$$

where USER and RESOURCE are greater than 0.

If the Exploitation_rate has a maximum value of 1.00, thus any value exceeding 1.00 is treated as 1.00.

●Definition 2.2:

This definition enforces an upper bound, lower bound and parameter t in the resource management calculations. The formula for parameter *t* is (3). In this scenario, resources are inseparable and users are only capable of retrieving one resource. The Exploitation_rate is calculated as follows:

$$Exploitation\_Rate = \frac{\sum_{i=1}^{m} \text{USER } i}{\sum_{i=1}^{n} \text{RESOURCE } i * t} \quad (2)$$

where USER and RESOURCE are greater than 0.

If the Exploitation_rate has a maximum value of 1.00, thus any value exceeding 1.00 is treated as 1.00.

$$t = Time\_Duration / Time\_Usage \quad (3)$$

where *t* is a positive integer.

●Case Study:

We now demonstrate examples, explicating definitions 2.1 and 2.2. Definition 2.1 uses a traditional resource management approach, in which one user is only capable of occupying a single resource unit at any point in time; Definition 2.2 refers to the application of the MSCE strategy; dynamic control of resource management, which increases overall resource availability and flexibility. Setting both the total resource quantity and the user load to 10, we perform the resource management calculations in Table 1. In addition, we provide an example employing definition 2.2: parameter t is determined for two different scenarios, resulting in values of 6 and 8, shown in Table 2 and Table 3. In Tables 2 and 3, the t parameter is applied only after the resource volume exceeds 6 (upper bound)

The difference between definition 2.1 and 2.2 is apparent in Table 4, which compares the results of Tables 1, 2, and 3 (the shaded column values). Note that the parameter *t* does affect the total resource availability, as depicted in Fig. 4. This figure presents a comparison of the resource consumption rate between the traditional and MSCE strategies of resource management. In this case study, the parameter *t* influences the curve of the resource consumption rate, in series II and III (Fig. 4).

Table 1. The Exploitation Rate of Definition 2.1 (I)

User	Resources									
	1	2	3	4	5	6	7	8	9	**10**
1	1.00	0.50	0.33	0.25	0.20	0.17	0.14	0.13	0.11	**0.10**
2	1.00*	1.00	0.67	0.50	0.40	0.33	0.29	0.25	0.22	**0.20**
3	1.00*	1.00*	1.00	0.75	0.60	0.50	0.43	0.38	0.33	**0.30**
4	1.00*	1.00*	1.00*	1.00	0.80	0.67	0.57	0.50	0.44	**0.40**
5	1.00*	1.00*	1.00*	1.00*	1.00	0.83	0.71	0.63	0.56	**0.50**
6	1.00*	1.00*	1.00*	1.00*	1.00*	1.00	0.86	0.75	0.67	**0.60**
7	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00	0.88	0.78	**0.70**
8	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00	0.89	**0.80**
9	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00	**0.90**
10	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	1.00*	**1.00**

*: The resource(s) cannot be separated, so the exploitation rate is 1.00

Table 2 The Exploitation Rate of Definition 2.2 (t=6) (II)

User	Resources									
	1	2	3	4	5	6	7	8	9	**10**
1	1.00	0.50	0.33	0.25	0.20	0.17	0.14	0.13	0.11	**0.10**
2	1.00*	1.00	0.67	0.50	0.40	0.33	0.29	0.25	0.22	**0.20**
3	1.00*	1.00*	1.00	0.75	0.60	0.50	0.43	0.38	0.33	**0.30**
4	1.00*	1.00*	1.00*	1.00	0.80	0.67	0.57	0.50	0.44	**0.40**
5	1.00*	1.00*	1.00*	1.00*	1.00	0.83	0.71	0.63	0.56	**0.50**
6	1.00*	1.00*	1.00*	1.00*	1.00*	0.17	0.14	0.13	0.11	**0.10**
7	1.00*	1.00*	1.00*	1.00*	1.00*	0.19	0.17	0.15	0.13	**0.12**
8	1.00*	1.00*	1.00*	1.00*	1.00*	0.22	0.19	0.17	0.15	**0.13**
9	1.00*	1.00*	1.00*	1.00*	1.00*	0.25	0.21	0.19	0.17	**0.15**
10	1.00*	1.00*	1.00*	1.00*	1.00*	0.28	0.24	0.21	0.19	**0.17**

*: The resource(s) cannot be separated, so the exploitation rate is 1.00

Table 3. The Exploitation Rate of Definition 2.2 (t=8) (III)

User	Resources									
	1	2	3	4	5	6	7	8	9	**10**
1	1.00	0.50	0.33	0.25	0.20	0.17	0.14	0.13	0.11	**0.10**
2	1.00*	1.00	0.67	0.50	0.40	0.33	0.29	0.25	0.22	**0.20**
3	1.00*	1.00*	1.00	0.75	0.60	0.50	0.43	0.38	0.33	**0.30**
4	1.00*	1.00*	1.00*	1.00	0.80	0.67	0.57	0.50	0.44	**0.40**
5	1.00*	1.00*	1.00*	1.00*	1.00	0.83	0.71	0.63	0.56	**0.50**
6	1.00*	1.00*	1.00*	1.00*	1.00*	0.13	0.11	0.09	0.08	**0.08**
7	1.00*	1.00*	1.00*	1.00*	1.00*	0.16	0.13	0.11	0.10	**0.09**
8	1.00*	1.00*	1.00*	1.00*	1.00*	0.17	0.14	0.13	0.11	**0.10**
9	1.00*	1.00*	1.00*	1.00*	1.00*	0.19	0.16	0.14	0.13	**0.11**
10	1.00*	1.00*	1.00*	1.00*	1.00*	0.21	0.18	0.16	0.14	**0.13**

*: The resource(s) cannot be separated, so the exploitation rate is 1.00

Table 4 The Comparisons of I, II, and I Exploitation Rate

Type	User (Resources=10 and User=10)									
	1	2	3	4	5	6	7	8	9	10
I	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
II	0.1	0.2	0.3	0.4	0.5	0.1	0.12	0.13	0.15	0.17
III	0.1	0.2	0.3	0.4	0.5	0.08	0.09	0.1	0.11	0.13

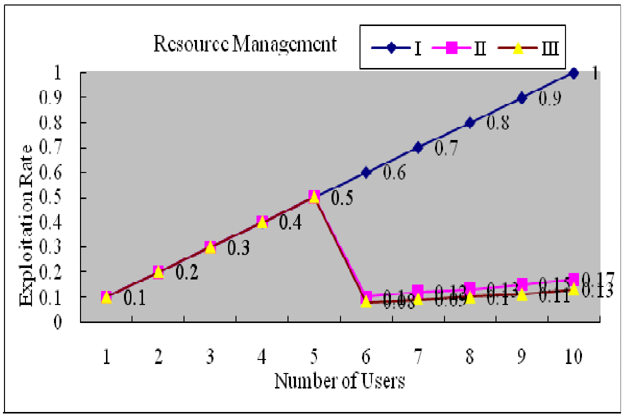


Fig. 4: The Exploitation Rate Comparison Chart

3. Implementation

This section uses a Resource Management System to demonstrate the implementation of a multi-agent mechanism employing agent behaviors and the MSCE strategy. The system architecture [8, 21, 22] of the resource management system is as shown in Fig. 5, including three main components: Management, Authentication and User. Each component has embedded agents[9], including a Policy Agent, Monitor Agent and Communication Agent.

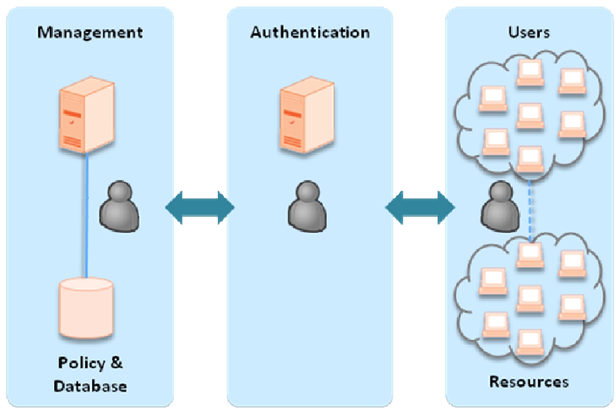
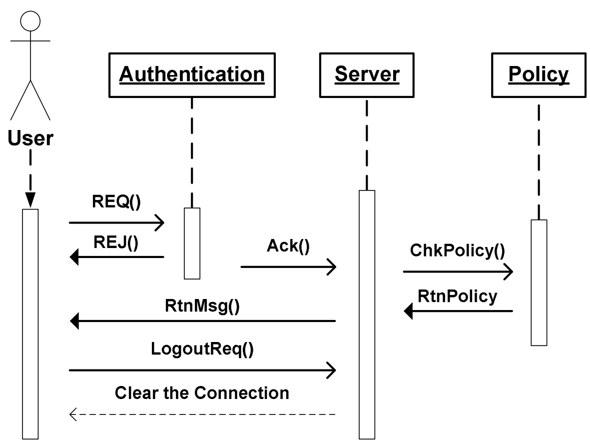


Fig. 5 System Architecture of Resource Management System

3.1 General Communication Procedure

The purpose of this procedure is to provide the user with the ability to retrieve resources following authentication, which is facilitated by the Resource Management System, as shown in Fig. 6.



REQ: Request
REJ: Reject
ChkPolicy: Check Policy
RtnPolicy: Return Policy
Ack: Acknowledge
RtnMsg: Return Message
LogoutReq: Logout Request

Fig. 6: General Communication Procedure

➤**Procedure Description:**

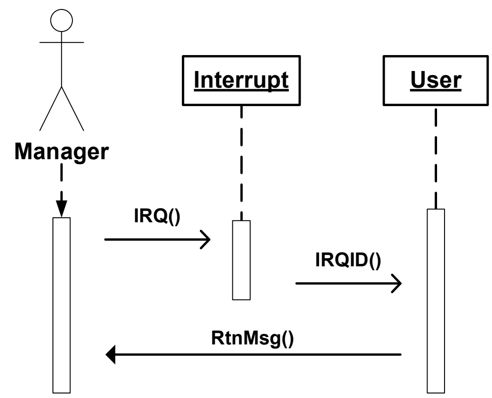
A. The user first sends a request (REQ) to the Authentication site when they access the Resource Management System. An ACK (acknowledgement) message is passed on to the Server site if the authentication is successfully validated; otherwise, a REJ (reject) message is returned to user.

B. The Server site determines the policy parameters via the ChkPolicy and RtnPolicy functions and subsequently returns a message to user via the RtnMsg function.

C. The LogoutReq function is used to clear the connection with Server when the user logs out of the Resource Management System. When this occurs, the resources are released back to resources pool.

3.2 Force Interrupt Procedure

The force interrupt procedure allows the system manager to directly execute an interrupt request, as shown in Fig. 7. The resources associated with a given task are released by the Resource Management System following a delay that is determined by the value of a predefined policy parameter (WAIT), which in this case has a value equal to zero. Once an event is triggered that results in the release of resources, those resources are immediately returned to the resource pool and, at the same time, the resource status is changed accordingly.



IRQ: Interrupt Request
IRQID: Interrupt Request ID
RtnMsg: Return Message

Fig. 7: Force Interrupt Procedure

➤**Procedure Description:**

A. The system manager sends an interrupt request (IRQ) to a particular user based on the MSCE strategy.
B. When an interrupt request is received by a user (IRQID), the system confirms the predefined policy parameters and executes the specified instruction, releasing resources to the resource pool.
C. The return message is retrieved by the system manager when the WAIT value is equal to zero.

3.3 Communication Agent

The Communication agent is responsible for obtaining resources for the user and performing authentication with the Server site, using the end-user's user ID and password. Once Resource Management System has completed authentication, it then retrieves the predefined policy parameters; if the authentication fails, the user's activities are immediately terminated, as shown in Fig. 8. Furthermore, the Communication agent begins to communicate with the Policy agent once the authentication has been completed. If the Resource Management System makes any changes to the policy parameters, they are subsequently applied.

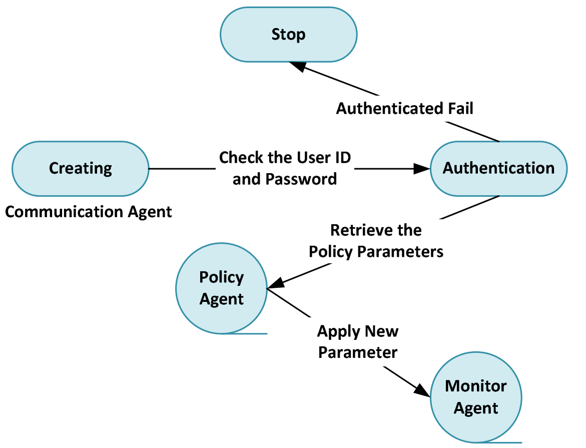


Fig. 8: Communication Agent State Diagram

3.4 Monitor Agent

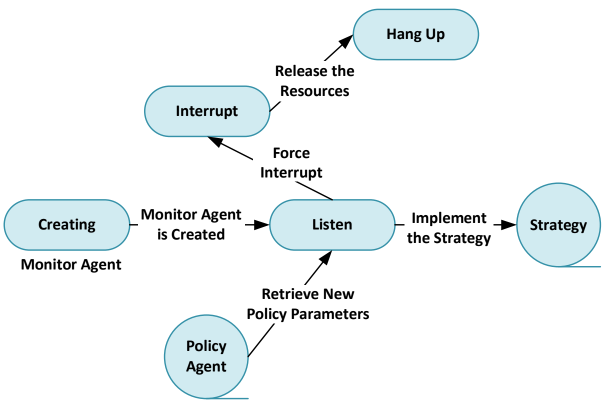


Fig. 9: Monitor Agent State Diagram

The purpose of the Monitor Agent is to monitor the Resource Management System's activities or events and to apply the MSCE strategy, dynamically adapting to changes in the system environment. Once the Monitor agent has been created, the system enters the Listen state, awaiting commands from the Policy agent or a Force Interrupt event (Fig. 9). The life cycle of the Monitor agent extends from the Resource Management System's initialization to shut down. In addition, this agent is responsible for maintaining the system activity logs.

3.5 Policy Agent

The Policy agent's function is to define dynamic policy parameters for the users, implementing the MSCE strategy, as shown in Fig. 10. Once the Policy agent has been created, the system enters the Initialize state, loading the initial values of policy parameters, initializing the Monitor agent and beginning to affect the resource management strategy. The Communication agent and Monitor agent will read the values of the policy parameters as necessary in the conduct of their tasks.

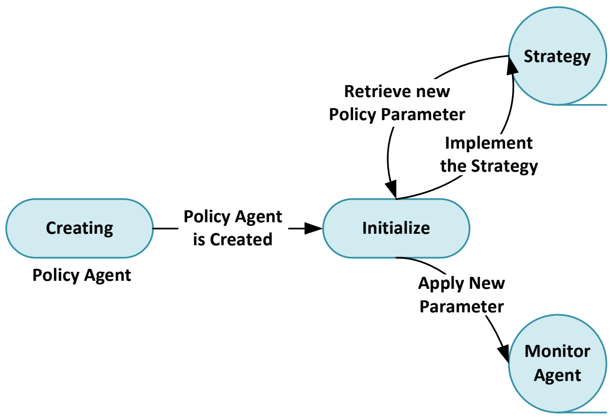


Fig. 10: Policy Agent State Diagram

3.6 System Application

We now examine a case study: the functionality of a Computer Lab RMS (Resource Management System) that has a graphical user interface (GUI) as shown in Fig. 11. This system provides the administrator with the ability to manage and control computer lab resources. The main functions of this RMS are displayed on the left hand side of the GUI, including Login, Setup, Policy, Monitor, Interrupt,

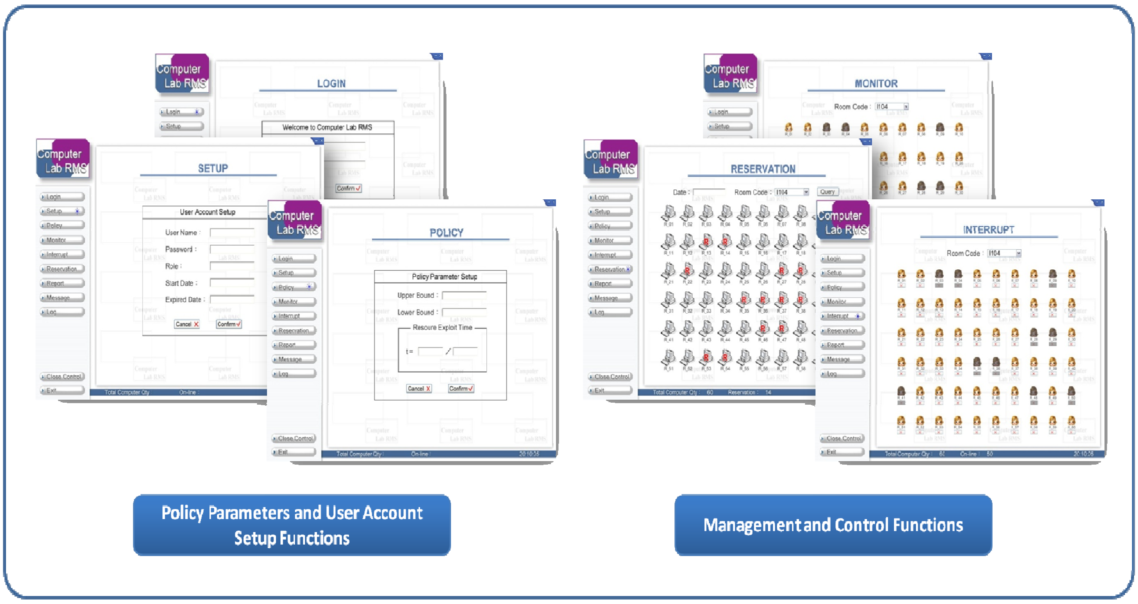


Fig. 11: Computer Lab RMS system functions and GUI

Reservation, Report, Message, Log, Close Control, and Exit.

Login: this function provides the system administrator with the ability to enter a username and password, thus authenticating them with the RMS system. Upon successful authentication, the system then returns an OK message, logging the user in; if authentication fails, the user is provided with an error message.

Setup: this function provides the system administrator with the ability to create a new user account, including an initial date of access, expiry date and role.

Policy: this function allows for the entry of upper and lower bound parameters, as well as a t parameter. The t parameter value is crucial to the operation of the MSCE strategy (refer to definition 2.2 and section 3.5).

Monitor: this function provides the system administrator with the ability to monitor the status of activity resources in real time (refer to section 3.4).

Interrupt: this function provides the system administrator with the ability to force the interruption of activities, releasing their associated resources back to the resource pool (refer to section 3.2).

Reservation: the reservation function provides the user with the ability to reserve computer lab resources over the Internet. Using this function, the system administrator can query the system to determine the reservation status of computer lab resources.

Report: the Computer Lab RMS system also provides a number of reports that can be viewed by the system administrator, including the computer lab usage list and the status of computer lab resources.

Message: this function provides the system administrator with the ability to send a message to a specified destination resource (computer) in the lab. This function only supports text based messages.

Log: the Computer Lab RMS log records detail lists of activities in the lab, including the usage of lab resources, forced interruptions and transmitted messages.

Close Control: this function is used to release control of a lab resource.

Exit: this function is used to exit the Computer Lab RMS system.

4. Conclusions

The key feature of a multi-agent system is the automatic execution of predefined, or user triggered procedures, in the background. This study has detailed the characteristics of agents operating in a multi-agent system, as well as a conceptualization of flexible resource management, employing dynamic switching between 4 strategies, including Expansion (when user loads are large and available resources are not limiting), Constraint (when user loads are large and resources are limiting), Modulation (when user loads are small and excess resources are held), and Supervision (when user loads are small and available resources are also small).

We have compared the traditional approach to resource management, as identified in definition 2.1, with the MSCE strategy, as identified in definition 2.2. The t parameter affects the rate of resource turnover. When the value of t is high, the rate of resource consumption is lower than would be the case under the traditional resource management methodology. To achieve this, the system employs a monitor agent (Section 3.4), which simultaneously controls the rate of resource consumption and monitors user loads.

The MSCE strategy reduces the requirement for human intervention as well as the variable cost of tax execution; this strategy also applies to many different fields, including the management of resources in call centers, computer labs (Section 3), etc. In Fig. 12, the MSCE strategy is applied in an implementation of a call center, employing a procedure that embeds the outlined agent behaviors. Agent (1) and Agent (2) receive requests from the PSTN (phone network) or Internet (computer network) and assign servants (On-Line resources) to deliver the requested services.

The volume of available resource quantities and the agent objectives affect the behavior of the agents, causing them to respond by employing behaviors, depending on the context. Agent (3) controls the on line resource usage; if the volume of available resources is insufficient to serve the requested call center services, the system automatically employs these agent behaviors to send a message to Agent (4). Agent (4) controls the set of ready resource and mediates communication between Agent (3) and Agent (5). If Agent (4)'s resources are sufficient, the system simply increases the number of available servants (i.e., On-Line resources) and sends a message to Agent (3). If Agent (4)'s resources are insufficient to address the request, Agent (4) communicates with Agent (5) to obtain more resources from the resource pool. This call center procedure demonstrates the dynamic management of resources in response to variations in the system environment, employing the agent behaviors of the MSCE strategy. In employing MSCE, it is evident that the management of resources becomes easier and more flexible.

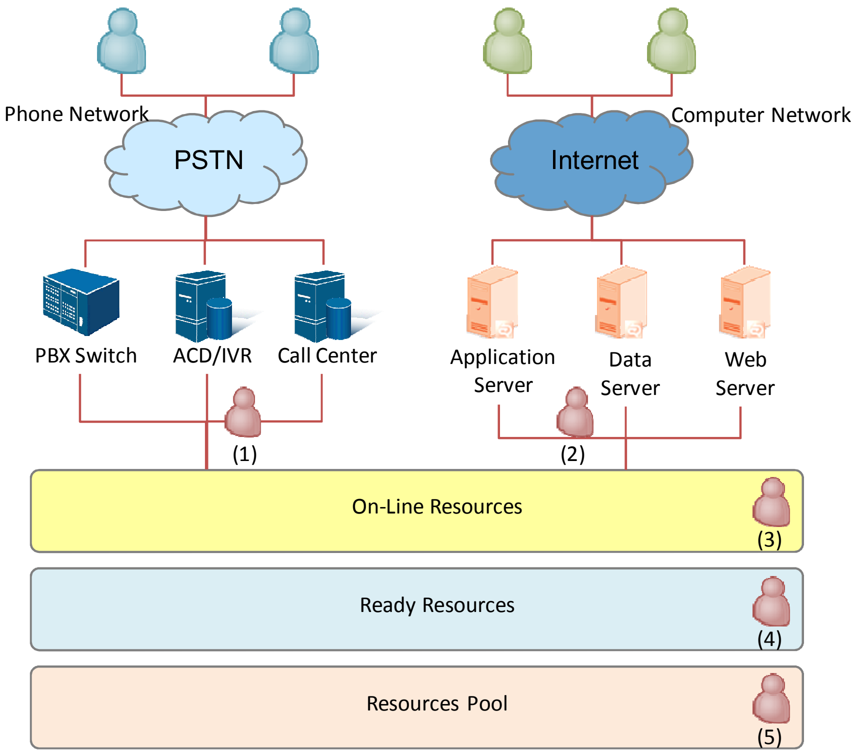


Fig. 12: Call Center with MSCE

References

[1] J.-W. Baek, H.Y. Yeom, A timed mobile agent planning approach for distributed information retrieval in dynamic network environments, *Information Sciences*, 176: 2006, 3347– 3378.

[2] Nazaraf Shah, Rahat Iqbal, Anne James, Kashif Iqbal, Exception representation and

management in open multi-agent systems, *Information Sciences*, 179 : 2009, 2555– 2561

[3] Pasquale De Meo a, Alfredo Garro b, Giorgio Terracina c, Domenico Ursino, Personalizing learning programs with X-Learn, an XML-based, "user-device" adaptive multi-agent system, *Information Science*s, 177: 2007, 1729– 1770

[4] P. Davidsson, M. Boman, Distributed monitoring and control of office buildings by embedded agents, *Information Sciences*, 171: 2005, 293– 307.

[5] S. Rahimi, J. Bjursell, M. Paprzycki, M. Cobb, D. Ali, Performance evaluation of SDIAGENT, a multi-agent system for distributed fuzzy geospatial data conflation, *Information Sciences*, 176: 2006, 1175– 1189.

[6] Sergio Ilarri, Eduardo Mena, Arantza Illarramendi, Using cooperative mobile agents to monitor distributed and dynamic environments, *Information Sciences*, 178: 2008, 2105– 2127

[7] S.M.T. Yau, H.V. Leong, A. Si, Distributed agent environment: application and performance, *Information Sciences*, 154: 2003, 5– 21.

[8] Eleni Stroulia and Maurine P. Hatch, An Intelligent-Agent Architecture for Flexible Service Integration on the Web, *IEEE TSMCC*, 33(4): 2003, 468-479.

[9] O'Hare, G.M.P., M.J. O'Grady, C. Muldoon and J.F. Bradley, Embedded agents: A paradigm for mobile services, Int. J. Web Grid Services, 2: 2006, 379-405.

[10] Giancarlo Fortino, Alfredo Garro and Wilma Russo, Achieving Mobile Agent Systems interoperability through software layering, *Information and Software Technology*, Vol.50, No.4, 2008, 322-341.

[11] Kalogeraki, V., Melliar-Smith, P. M., Moser, L. E., and Drougas, Y., Resource management using multiple feedback loops in soft real-time distributed object systems, *Journal of Systems and Software*, Vol.81, No.7, 2008, 1144-1162.

[12] Li, C., and Li, L., An agent-oriented and service-oriented environment for deploying dynamic distributed systems, *Computer Standards & Interfaces*, No. 24: 2002, 323-336.

[13] Liviu Panait and Sean Luke, Cooperative Multi-Agent Learning: The State of the Art, *Autonomous Agents and Multi-Agent Systems*, Vol. 11, No.3, 2005, 387 - 434.

[14] LIU, J. and YAO, C., Rational competition and cooperation in ubiquitous agent communities, *Knowledge-Based Systems*, 17, 2004, 189-200.

[15] Lu, Y., Xiao, N., and Yang, X., Scalable Resource Management System for High Productive Computing, *In Proceedings of the the Third Chinagrid Annual Conference (Chinagrid 2008) CHINAGRID. IEEE Computer Society,* Washington, DC, 2008, 331-337.

[16] Magana, E. and Serrat, J., Distributed and Heuristic Policy-Based Resource Management System for Large-Scale Grids, *In Proceedings of the 1st international Conference on Autonomous infrastructure, Management and Security: inter-Domain Management A. K. Bandara and M. Burgess, Eds. Lecture Notes In Computer Science, vol. 4543. Springer-Verlag, Berlin, Heidelberg,* 2007, 184-187.

[17] Mehdi Dastani and Jorge J. Gomez-sanz, Programming multi-agent systems, *The Knowledge Engineering Review*, Vol.20, No.2, 2005, 151-164.

[18] Nicoletta Fornara, Francesco Vigano and Marco Colombetti, Agent communication and artificial institutions, *Autonomous Agents and Multi-Agent Systems*, Vol.14, No.2, 2007, 121-142.

[19] Noureddine Boudriga and Mohammad S. Obaidat, Intelligent Agents On The Web, *A Review. Computing in Science and Engineering*, 06(4): 2004, 35-42.

[20] Ottens, K., Hernandez, N., Gleizes, M., and Aussenac-Gilles, N., A Multi-Agent System for Dynamic Ontologies, *J. Log. and Comput*, 19, 5, 2009, 831-858.

[21] Silva, C., Castro, J., Tedesco, P., Araujo, J., Moreira, A., and Mylopoulos, J., Improving the architectural design of multi-agent systems: the tropos case, *In Proceedings of the 2006 international Workshop on Software Engineering For Large-Scale Multi-Agent Systems SELMAS '06. ACM, New York*, NY, 2006, 107-113.

[22] Steven K.C. Lo, Huan-Chao Keh, Yi-Hung Lin and Wang Jo-Chi, Integrated the Intelligent Agent Behavior Model and Billing Service into Communication System, *Information Technology Journal* 8 (5): 2009, 668-677.

[23] Tweedale, J., Ichalkaranje, N., Sioutis, C., Jarvis, B., Consoli, A., and Phillips-Wren, G., Innovations in multi-agent systems. *J. Netw. Comput.* Appl. 30, 3, 2007, 1089-1115.

[24] Yulian Fei and Wenjuan Chen, A Multi-agent, Multi-object and Multi-attribute Intelligent Negotiation Model, *In Proceedings of the Fourth international Conference on Fuzzy Systems and Knowledge Discovery - FSKD. IEEE Computer Society*, Washington, DC Volume 03, 2007, 440-446.

[25] Zambonelli, F., Jennings, N. R., and Wooldridge, M., Developing multiagent systems: The Gaia methodology, *ACM Trans. Softw. Eng. Methodol*, 12, 3, 2003, 317-370.

[26] Kotz, D. and S. Robert Gray, Mobile agents and the future of the Internet, *ACM SIGOPS Operat. Syst. Rev.*, 33: 1999, 7-13