Journal of Information Processing Vol.22 No.4 660–668 (Oct. 2014)

[DOI: 10.2197/ipsjjip.22.660]

Regular Paper

Design and Evaluation of Energy-consumption-aware Evolutional Agent System for Portable Devices

Wenpeng Wei[1,a)] Akiko Takahashi[2,b)] Tetsuo Kinoshita[1,3,c)]

Received: August 26, 2013, Accepted: May 17, 2014

Abstract: Towards the improvement for service provision through portable devices which have limited resources, we propose a new design of energy-consumption-aware evolutional agent system. In the proposed design, platform specific parameters of local resources are taken into account to calculate the energy continuance index (ECI) of portable devices for controlling service provision in order to reduce power consumption of the portable platforms. Furthermore, we implement a multimedia communication system based on the proposal in order to evaluate its effectiveness. Results from the comparison with a conventional design show that the proposal is able to provide adequate services without affecting user experiences.

Keywords: energy consumption awareness, multi-agent system, multimedia communication system, evolutional agent system, portable device

1. Introduction

With the rapid development and popularisation of smart portable devices, it is becoming common to use various network services on portable devices nowadays such as navigation services on smart phones. However, those portable devices are usually used in unstable and dynamic environments with certain limitations such as variation in network bandwidth. In addition to that, not only the limitations of the environments, but also the limitations of the devices themselves such as limited battery power. For instance, users of multimedia sharing services like YouTube have to always keep an eye on the battery charge of their devices while enjoying the contents to make sure the playback will not be suspended.

A number of researches trying to improve the situation mentioned above have been undertaken in recent years. Those research efforts can be divided into two main streams. One of them is the works that attempt to reduce local resource consumption of the portable devices by optimising local processing, for instance, a smart phone application manager trying to extend battery life by exiting background processing. Due to the lack of ability to control service provisions outside the portable devices, those works show only limited effect in the case that the numerous system resources are consumed by the services which are provided by independent service providers. The other one is the efforts on the service provider side to prepare optimised services based on the situations of users and their devices. Nevertheless, it is obviously difficult, if not impossible, to predict all possible situations and prepare suitable services for all users in advance.

To solve the problems mentioned above, this research aims to provide appropriate services to portable devices flexibly by considering the situations of users and their devices while the users are enjoying the services. The remainder of this paper is organised as follows. Related works and main challenges are discussed in the next section. In Section 3, the proposal is presented in detail. An experimental system is described and the results discussed in Section 4. Finally, Section 5 presents important conclusions.

2. Related Works and Challenges

2.1 Related Works

There are quite a number of researches aiming at reducing the energy consumption of portable devices. A mechanism which sleeps the processors of portable devices for short periods to save battery power is suggested in Brakmo et al. [1]. While in Qiu et al. [2], an algorithm of task scheduling to aggressively reduce energy consumption is provided. Both approaches mentioned above are limited for portable devices and lack of ability to control service provision when portable devices are using network services.

On the other hand, automatic and dynamic construction of services according to the situations of service users and environments has been an active research area for a long time. In the last decade, a number of researches on autonomous service provisioning systems have been trying to provide appropriate services depending on the situations of the systems. Users of those systems are required to construct and to control the service provisioning system dynamically to receive adequate services.

[1] Graduate School of Information Sciences, Tohoku University, Sendai, Miyagi 980–0812, Japan
[2] Department of Information Systems, Sendai National College of Technology, Sendai, Miyagi 989–3128, Japan
[3] Research Institute of Electrical Communication, Tohoku University, Sendai, Miyagi 980–0812, Japan
[a)] wei@k.riec.tohoku.ac.jp
[b)] akiko@sendai-nct.ac.jp
[c)] kino@riec.tohoku.ac.jp

Regarding multi-agent based autonomous service provision systems, methods for constructing the autonomous overall system using a combination of software agents that collaborate with each other autonomously have been proposed [3], [4]. Those methods construct agent organizations with mutual interactions among autonomous agents to achieve autonomous system characteristics. Depending on QoS degradation and environmental changes, the systems change parameters and reconstruct the agent organization during service provisioning. The QoS degradation and environment changes serve as triggers in those systems which function reactively.

An agent organization model to construct a system that recovers autonomously when trouble occurs is reported in Oyenan et al. [5]. Knowledge for the control of agents is required by the reported organization model: the condition of the agent organization has to be monitored. A multi-agent system is constructed systematically by operating agents autonomously under the model. In case some agent is aborted or some trouble which does not satisfy the system's objective during service provision, the reconstruction of agent organization is performed to recover service provision autonomously. Nonetheless, service suspension is difficult to avoid by this model.

A multi-agent system architecture incorporates a policy to manage applications, resources and service provision is proposed in Tesauro et al. [6]. The proposal is able to construct systems based on user requirements, to recover from system failures and to optimise systems. However, in case that the proposed system handles every system component using the same policy, the overall system becomes overloaded easily.

There are also a number of researches aiming to construct multi-agent systems which are able to adapt to changeable unstable environments. Disparate models for that purpose have been introduced in various publications [7], [8]. The former provides a meta-model focusing on organisational concepts such as roles in a multi-agent system. The latter gives a concrete model for the actual modelling agent organizations of the application systems. In those approaches, the adaption of agent behaviour is realised by changing the roles which agents play dynamically after some predefined conditions are perceived. In Luckey et al. [9], a useful tool to define those important predefined conditions is provided. By extending standard UML use cases, the proposed Adapt Cases enable the explicit modelling of those conditions with domain specific means early in the design phase of software engineering process. It is noteworthy that the approaches mentioned above perform adaptive actions after the system falls into some undesirable situations. Even if the systems recover after that, user experiences are noticeably affected especially under an unstable environment such as wireless network.

Addressing the problems mentioned above, in our previous researches [10], [11], a method that is able to predict system troubles in order to maintain service provision by adopting the Evolutional Agent System (EAS) [12] has been proposed. Instead of monitoring undesired conditions of the applications as the triggers for reconstruct agent organizations, the proposed approach focuses on the characteristic of agent behaviour and application requirements to obtain the current system potential as system margin for providing required services. Based on the system margin, the proposed approach controls multi-agent system proactively considering the "control possibility" and "control effect" for reducing the decrement of QoS to the absolute minimum while providing services by reconstructing agent organizations. This approach avoids QoS deterioration effectively for systems which consist of immobile platforms. Therefore, it is natural to apply the succeed approach to portable devices. However, there are several challenges due to the unique limitations of portable devices mentioned above.

2.2 Challenges of Evolutional Agent System for Portable Devices

For portable devices which are used in unstable environments, it is obvious that autonomous service provisioning systems are suitable for users to experience adequate quality. While adapting our previous experience to portable devices, we meet the challenges that can be described by the following problems.

(P1) Without considering system power storage and other local resources, device battery life is shortened by additional control processing.

(P2) The extra control process reduces user experiences while enjoying provided services due to the limited local resources.

In the context of portable devices in this research, user experience shows if the provided service is fully enjoyed on the client side. For an instance, if a video is streamed from the server at 30 fps but on portable client it is played only at 15 fps, we consider that user experience is noticeably affected. If the video is served at 15 fps and played at the same fps, we consider the user experience is not affected. On the other hand, the user requirement is the functions with user required qualities.

Different from immobile devices, portable devices have certain limitations. Those limitations have to be considered while applying EAS concept to systems which involve mobile devices. Limited battery capacity is one of those limitations. In an EAS, besides the agent system which provides the required services, there is an extra evolution mechanism to improve the whole system. It is obvious that the extra computation of the evolution mechanism consumes extra energy of the system. That might not be a serious issue for immobile devices but for the devices with limited power supply it means shorter battery life which is an undesirable condition. Another limitation is the limited processing power of portable devices. On those devices, extra controlling process consumes additional computation resources and effects user experience.

Therefore, against those challenges we propose a new design of energy-consumption-aware EAS for portable devices. The proposed design customises and extends previous EAS model to overcome the limitations of portable devices. In brief, the proposal provides solutions to the challenges mentioned above as follows.

(S1) A service provision control mechanism considering device energy consumption and local resource. In addition to QoS parameters, local resources parameters are also taken into account while service is being provided to reduce the influence on device battery life.

(S2) A system architecture which considers user experiences during service provision. Instead of users and their devices, software agents on service provider side deal with the extra controlling.

Instead of performing control process of evolution mechanism on portable devices, the proposed design performs those process using immobile platforms of the whole system as much as possible to minimise the effect on portable devices. The details of the proposal are described in next section.

3. Proposal

3.1 Customised Model of Evolutional Agent System

An Evolutional Agent System (EAS) is a multi-agent system which has the ability to control and reconstruct itself actively to recover QoS after analysing the system environment, not merely passively responding to changes. We applied the EAS concept to construct an autonomous service provision system in the previous research [10]. In our EAS model, required service is provided by the cooperation of some autonomous software components called "agents". Those software agents perform certain functions autonomously and communicate to each other to cooperate for delivering more sophisticated services to users. Based on that successful experience, in this research we customise the EAS model from Ref. [10] by explicitly modelling energy consumption related characteristics of agent behaviour and the environment to overcome the limitations of portable devices. The customised EAS model is described in a syntactic manner as the following.

Definition 1. An Evolutional Agent System, denoted as an EAS, is designed based on ne, R, S, E and P;

$$EAS =< ne, R, S, E, P > .$$

Where ne signifies the name of the EAS, R denotes a set of requirements given to EAS, S is the EAS structure, E stands for the environment in which EAS operates, and P is the EAS property.

The EAS usually consists of several agent workplaces and one agent repository running on distributed platforms. Detailed elements of an EAS in this research are described as the following definitions.

Definition 2. R includes $req_k(k = 1, 2, ..., N_r)$ and req_k is determined as a required service function $req\text{-}f_k$ and the required quality $req\text{-}f\text{-}q_k$ of the $req\text{-}f_k$, as shown below.

$$R = \{req_k | req_k =< req\text{-}f_k, req\text{-}f\text{-}q_k >; k = 1, ..., N_r\}$$

Here, N_r is the total number of the required functions.

Definition 3. S is an organization of an EAS. It consists of AG, STR and $AL\text{-}AG$. S is

$$S =< AG, STR, AL\text{-}AG >,$$

where AG denotes a set of EAS member agents, STR stands for the structure of an agent organization and $AL\text{-}AG$ signifies a set of alternate agents which member agents in AG can be replaced by.

Firstly, AG includes $ag_i(i = 1, ..., N_a)$ which is determined with the following: identifier na_i; a set of services $Sv\text{-}ag_i$ which can be provided by the ag_i.

$$AG = \{ag_i | ag_i =< na_i, Sv\text{-}ag_i >; i = 1, ..., N_a\}$$

Here, N_a is the total number of EAS member agents. Member agents AG are problem solving agents which run in agent workplaces to provide required services.

$Sv\text{-}ag_i$ includes sva_i, which is determined as a provided function $func_i$, the provided quality $func\text{-}q_i$ of the $func_i$ and relative energy consumption $func\text{-}ec_i$ of providing $func_i$ as follows:

$$Sv\text{-}ag_i = \{sva_i | sva_i =< func_i, func\text{-}q_i, func\text{-}ec_i >; \\ i = 1, ..., N_s\}.$$

Here, N_s is the total number of functions agent ag_i has.

Secondly, $AL\text{-}AG$ includes $ag_j(j = 1, ..., N_{alt})$ as the following.

$$AL\text{-}AG = \{ag_j | ag_j =< na_j, Sv\text{-}ag_j >; j = 1, .., N_{alt}\}$$

Here, N_{alt} is the total number of the alternate agents. Alternate agents $AL\text{-}AG$ are stored in the agent repository which is running on top of a immobile platform such as a PC server.

Finally, STR includes $rel_{ij}(i, j = 1, ..., N_a)$ which denotes the relation r_{ij} between ag_i and ag_j. Here, ag_j and ag_i are included by AG, r_{ij} is included by Rel which is the set of inter-agent relations defined in EAS, respectively.

The STR is

$$STR = \{rel_{ij} | rel_{ij} =< r_{ij}, ag_i, ag_j >; \\ ag_i, ag_j \in AG; r_{ij} \in Rel\}.$$

All the agents in the EAS are stored as alternate agent in the agent repository at first. After a requirement is issued to the EAS, appropriate agents are initiated from the agent repository to the agent workplaces as member agent to construct agent organizations which provide the user required services. Those member agents are selected in a design process as following.

Definition 4. In a design process, the EAS is feasible and S is determined when all requirements and functions are compared and matched. It can be described by the following expression:

$$\begin{aligned} (\forall k) req_k &= < req\text{-}f_k, req\text{-}f\text{-}q_k > \in R, \\ (\exists i, j) ag_i &= < na_i, Sv\text{-}ag_i > \in AG; \\ sva_j &= < func_j, func\text{-}q_j, func\text{-}ec_j > \in Sv\text{-}ag_i; \\ req\text{-}f_k &\leq func_j; \\ req\text{-}f\text{-}q_k &= func\text{-}q_j. \end{aligned}$$

Using this expression, S is determined.

By that design process, agents that have necessary functions to realise the user requirements are selected as member agent. In case that multiple agents are suitable for one certain requirement, the agent with highest quality of function is selected. After the process, the EAS is ready to provide required services using the agent organization constructed by member agents AG.

P is the operating property of the EAS. It is determined by the evolutional mechanism of the EAS. If the system can recover its performance, then the evolutional mechanism operates the system actively and dynamically based on the operating status of the multi-agent system and the information of system environment. In this study, we design the evolutional mechanism as the knowledge and abilities of a meta-agent (mag). It manages P as internal

information. The details of evolutional control based on system margin are described in Ref. [10]. In addition to that, to overcome certain limitations of portable devices, the hardware information of distributed platforms is also necessary as follows.

Definition 5. E is the information of the behavioural environment of the EAS system. In this research, E consists of the hardware information HW_i of each distributed platform:

$$E = \{HW_i | i = 1, ..., N_{dp}\},$$

where N_{dp} is the total number of distributed platforms.

With that information, the EAS is able to be aware of the energy consumption and potential of the whole system as discussed in the next section.

3.2 Energy-consumption-aware Design of EAS

To design an energy consumption aware control mechanism, we propose the Energy Continuance Index ECI as an indicator of the estimated continuance of system energy storage for each distributed platform. ECI is one of the detectable properties of EAS which are represented by the set P.

By calculating the energy consumption related data which is obtained from platform hardware, it is possible to be aware of the impact of the current service on the energy storage of the platform. ECI is defined as a numerical value to measure that impact which is directly proportional to current battery charge of the platform and is inversely proportion to the energy consumption speed.

Definition 6. The Energy Continuance Index ECI is obtained using platform hardware information HW_i as the follows.

$$ECI = \{eci | eci =< eng\text{-}strg_i, ECF_i >; eng\text{-}strg_i \in HW_i; ECF_i \subseteq HW_i; i = 1, ..., N_{dp}\}$$

where $eng\text{-}strg_i$ denotes the current energy storage of the platform, ECF_i is a set of energy consumption factors of the platform as follows:

$$ECF_i = \{ecf_{ij} | ecf_{ij} \in HW_i; j = 1, ..., N_f\},$$

where N_f is the number of energy consumption factors of the platform.

In this study, for simplicity we define eci as the following.

$$eci_i = \frac{N_f \times eng\text{-}strg_i}{\sum_{j=1}^{N_f} (\alpha_{ij} \times ecf_{ij})},$$

where α_{ij} is the weight of each factor and N_f is the total number of ecf_{ij}. The range of α_{ij} is $(0, 1]$ and for each ecf_{ij} a specific α_{ij} is defined. $eng\text{-}strg_i$ is a numerical value which represents energy storage, the battery charge for instance, in the range of $[0, 1]$. ecf_{ij} denotes the energy consumption factor which influences the energy storage using a value from $[0, 1]$. Therefore, eci_i is a numerical value which is larger than or equal to 0, where a value of 0 shows that the platform has no energy in the storage to operate anymore, while a value of positive infinity means that the platform has minimal energy consumption and is expected to provide maximal battery life that hardware allows. In practice, $eng\text{-}strg_i$ and ecf_{ij} can be various over disparate platforms. A concrete instance is shown in next section.

ECI is updated and monitored frequently at regular intervals during service provision. The architecture of EAS with energy consumption awareness is presented in **Fig. 1**. The agent workplace is the behavioural platform which contains the member agents ($ag_i \in AG$) to provide user required services. The alternate agents ($ag_i \in AL\text{-}AG$) are managed by the meta-agent (mag) in the agent repository which constructs agent organizations in response to user requirements. mag also retrieves platform specific hardware information HW_i from each distributed platform and keeps updating and monitoring ECI.

Using the architecture described above, the agent behaviour to archive energy consumption awareness is designed as follows. During service provision, if a ECI degradation, i.e. acute energy consumption which may seriously affect system power storage, is detected by mag, it dynamically reconstructs the agent organization by replacing corresponding member agents with alternative agents which have equivalent functions with same QoS and lower relative energy consumption as described next.

Definition 7. The agent organization S is reorganised if

(1) ECI degradation is detected:

$$eci + \Delta eci < \theta,$$

(2) appropriate alternative agent is available:

$$\begin{aligned}
(\exists i) ag_i &\in AG,\\
(\exists j) ag_j &\in AL\text{-}AG,\\
&\text{where}\\
(\forall k) sva_k &= < func_k, func\text{-}q_k, func\text{-}ec_k > \in Sv\text{-}ag_i,\\
(\exists q, l) sva_q &= < func_q, func\text{-}q_q, func\text{-}ec_q > \in Sv\text{-}ag_j;\\
req_l &= < req\text{-}f_l, req\text{-}f\text{-}q_l > \in R;\\
func_k &= func_q = req\text{-}f_l;\\
req\text{-}f\text{-}q_l &\leq func\text{-}q_q;\\
func\text{-}ec_k &> func\text{-}ec_q.
\end{aligned}$$

Here, θ is the threshold.

In other words, for a certain member agent, the alternative agents, which have the same functions as the member agent and meet the required function quality and consume less power than the member agent are considered as appropriate. In the case of that multiple alternative agents are suitable, the one with the highest function quality is selected.

In summary, ECI is introduced as an additional measurement besides application domain related QoS to enable the whole system to consider both at the same time. Furthermore, unlike the QoS as a performance measurement, ECI predicts the system energy continuance in the future and provides the ability to take actions in advance. Usually, it appears as a trade-off between energy continuance and performance. In this case, ECI is also used as the tool for find appropriate balance in the trade-off.

By implementing those models as behaviour knowledge of software agents in the EAS, energy consumption awareness is realised. This proposed model is based on multi-agent system, and is able to be applied to multi-agent based applications. An

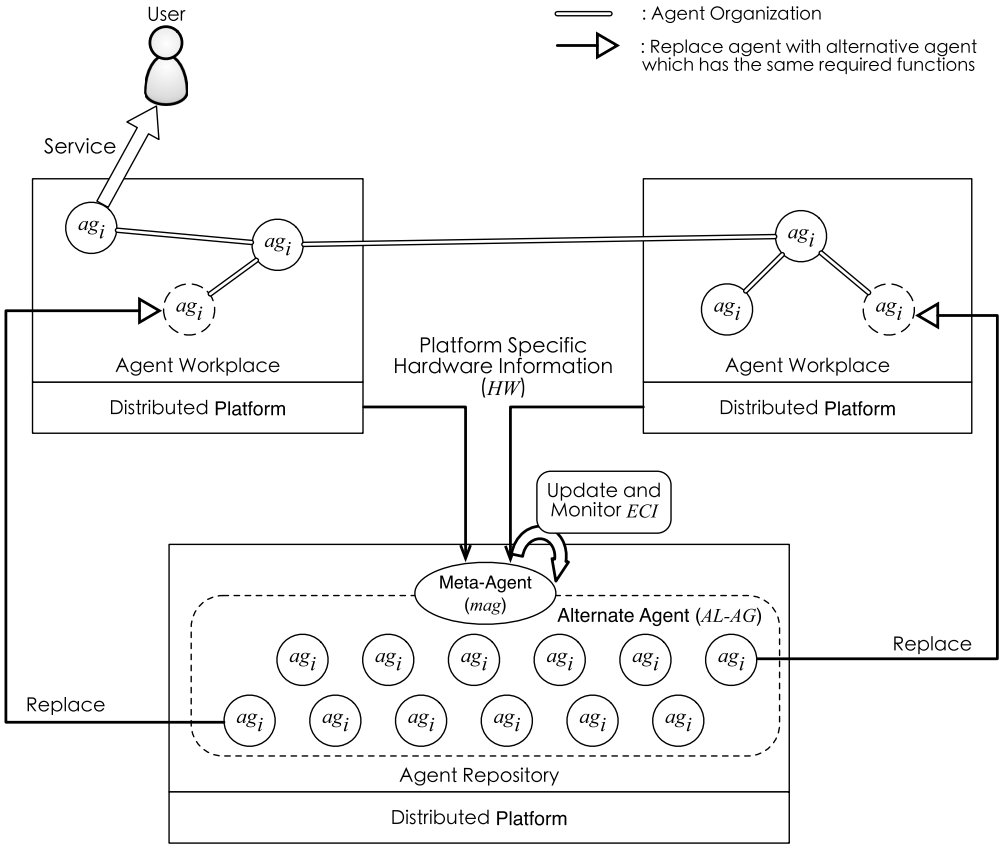

Fig. 1 System Architecture of Energy-consumption-aware EAS.

application example is described in the next section.

4. Experiment and Evaluation

4.1 Application to Multimedia Communication System

To evaluate our proposal, we applied it to a Flexible Multimedia Network Middleware (FMNM) [13], which is an agent-based multimedia communication system providing adequate multimedia communication services dynamically based on user requirements. In this research, the experimental application contains a live video server and one or more portable devices as clients. The server captures live video with a web camera and encodes the video while streaming it to the clients to play it back. As the previous research of FMNM, the quality of the provided service in terms of video frame rate (fps) and video resolution are used as QoS parameters. Every QoS parameter is implemented as a component of design specifications and the parameters of user requirement's function and its quality are implemented as *req-f* and *req-f-q*. The agents in the system are as follows:

video capturing agent The agent controls the web camera and passes the video raw data from the camera to the video encoding agent. The controller and driver of the web camera on the video server is implemented as the knowledge of this video capturing agent.

video encoding agent The agent encodes the video raw data using different video encoders with variable parameters to pass to the video streaming agent. All the video encoders available on the video server are wrapped and implemented as the knowledge of this video encoding agent.

video streaming agent The agent controls network communication to send and to receive encoded video data from video encoder agent to video agent. The networking related controllers and drivers of both server and client are implemented as the knowledge of this video streaming agent.

video agent The agent receives encoded video data from video streaming agent and decodes the data using corresponding video decoder to play back the video on the display. All the video decoders and the video player on the client are implemented as the knowledge of this video agent.

In addition, each of the agents also has the knowledge for communication with each other. Specifically, for the video encoding agent and the video agent, there are disparate agents for different video encoders/decoders with varying parameters. The agents for each type have the same functions but with different internal implementations. They perform the same task such as encoding a video with different QoS and energy consumption depending on the concrete encoder/decoder implementation and parameters. During system initiation, user requires a multimedia communications service which is able to play a remotely captured live video on a Android tablet device at a required frame rate and resolution. The system selects appropriate member agents from the agent repository according to the design process defined in Definition 4. In this process, the agents combination which is able to provide required services at highest QoS is selected for constructing agent organization.

After constructing the agent organization which is able to provide user required service, the experimental prototype system continuously provides streaming services for live video captured from a web camera of a PC to an Android tablet. For the An-

droid tablet, in addition to the conventional QoS parameters such as video frame rate, local resource parameters are also involved as HW. In this prototype system, those local resource parameters include the following: CPU load (%), memory usage (%), display brightness (%), wireless network signal strength (%), network traffic (kB) and battery charge (%).

$$HW =< cpu\text{-}load, mem\text{-}usg, disp\text{-}brt, wifi\text{-}sigl, nw\text{-}traf, batt\text{-}chrg >$$

During the required service provision, the meta agent mag in the agent repository receives HW update from the table every second and uses it to calculate the energy continuance index ECI of the tablet. For the use case in this experiment, the battery power of the tablet is mainly used in three parts as following.

(1) Network part: to receive video data through the wireless network.
(2) Processor part: to decode the video data using CPU calculation.
(3) Display part: to playback the video on the display.

Specifically, for the network part, the power consumed for wireless communication is inversely proportional to the signal strength of the networks. More power is needed for a connection with relatively weak signal. In summary, there are three factors of energy consumption ecf of the tablet. Also the tablet is the only portable device in this experimental system for the calculation of ECI. Therefore, ECI of the system is defined as follows.

$$ECI = \frac{3 \times batt\text{-}chrg}{cpu\text{-}load + disp\text{-}brt + f\text{-}scale\left(\frac{nw\text{-}traf}{wifi\text{-}sigl}\right)}$$

Here, $f\text{-}scale$ is a function to scale the network part factor to the same numerical range of the other factors to guarantee they have the same influence on the ECI. For simplicity, we use 1 as the weight α for all the factors. In this way, the mag is able to be aware of the energy continuance index of the portable tablet device.

If an ECI degradation of the system is detected by the mag, it looks for alternative agents which have the same required functions as member agents but consume less energy in the set $AL\text{-}AG$ according to Definition 7. The relative energy consumption of a certain function is represented by $func\text{-}ec$ element of sva. For example, encoding video in high resolution consumes more resources and more power than encoding video in low resolution. In this study, the value of $func\text{-}ec$ is 1 for encoding video in full resolution and 0.8 for 80% resolution.

In case that appropriate alternative agent exists in the repository, corresponding member agent is replaced by it to recover the ECI. i.e., to reduce the energy consumption of the portable device in order to recovery expected battery life. In this process, as a trade-off to reducing the rate of energy consumption , the QoS is usually reduced. However, the reduced QoS is still able to satisfy user requirements otherwise the reorganization will not be performed.

The experimental application described above is implemented using ADIPS/DASH agent framework [14] and Java language. The live video server is running on a Unix-based desktop PC while the clients are implemented using tablets running Android OS. The server and clients are connected using wireless network.

4.2 Experiment, Results and Discussion

To evaluate the feasibility and effectiveness of our proposal, using the experimental application described above, two comparison experiments have been performed. The application system to compare provides the same video streaming service to portable devices. However, without adopting the proposed EAS model, the conventional system lacks the ability to consider energy consumption of portable devices. The conventional system simply monitors only the current QoS during service provision. If current QoS is lower than required, the system tries to adjust the streaming parameters, such as bit rate and resolution, to maintain QoS. For instance, in case that video at 30 fps is required, if the current fps is lower than that due to the poor network situation, the conventional system tries to recover video frame rate by reduce the streaming bit rate. Nevertheless, the conventional approach based on QoS only is not enough for the situation that both QoS and energy consumption are needed to be considered, such as the system contains portable devices. The following experiments show the differences between conventional approach and proposal.

4.2.1 Experiment 1: Feasibility Evaluation

This experiment is set up to evaluate the feasibility of the proposal. As shown in **Fig. 2**, a live video captured by a web camera of a PC is continuously streaming to an unplugged Android tablet using the multimedia communication service based on FMNM. The requirements to the service are live video streaming at over 15 fps and 0.25 times resolution. The details of video encoding agents in the application using proposal are shown in **Table 1**. Through this experiment, the network situation is stable and the video playback fps on the portable device is always at 30.

The tablet starts to use the service and plays the video stream

Fig. 2 Snap shoot of experiment 1.

Table 1 Details of alternative video encoding agents.

func	*func-q* (video resolution scale)	*func-ec*
Encode	1	1
Encode	0.75	0.75
Encode	0.6	0.6
Encode	0.4	0.4
Encode	0.25	0.25
Encode	0.1	0.1

Table 2 Result of experiment 1: comparison of portable device battery life.

	conventional method	our proposal
battery life (hours)	3.78	6.74
increment	n/a	78%

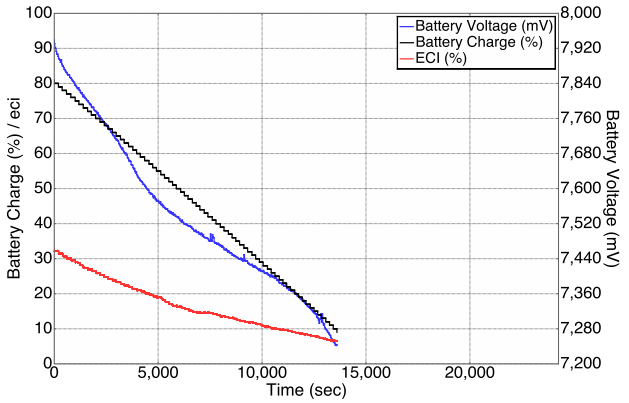

Fig. 3 Result of experiment 1: the *ECI* and battery information of conventional method.

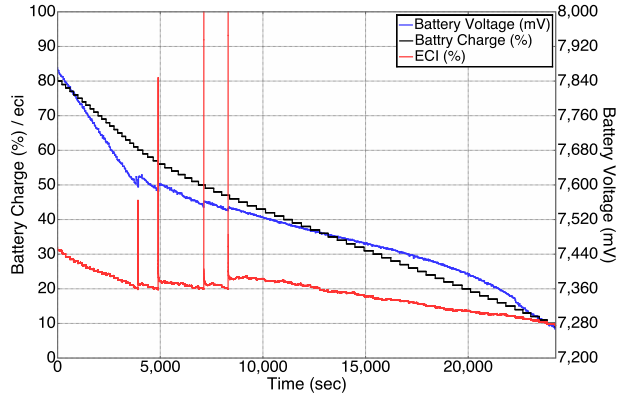

Fig. 4 Result of experiment 1: the *ECI* and battery information of proposed method.

when it has 80% battery charge and stops when the battery charge is less than 10%. Identical experiments are repeated for the conventional method from previous work which only considers QoS parameters and proposed method of this research. The time spend in that operation is recorded as system battery life which is the main evaluation aspect. The local resource parameters such as battery voltage and battery charge are also collected during the service provision to help analysing experiment results. The comparison of portable device battery life is shown in **Table 2**. The details of battery is presented in **Fig. 3** and **Fig. 4**, where the x-axis shows time and y-axis shows battery voltage, battery charge and ECI of the tablet *ECI*, respectively.

First of all, as shown in Table 2, while the conventional method does not consider energy consumption of the portable devices, our proposal significantly saves battery power to increase battery life of the device. In the experiment using conventional method, it takes 3.78 hours to drain the tablet battery power from 80% to 10% during the multimedia communication service provision. By using our proposal, for providing the same service with maintaining the same QoS represented by video frame rate, the same battery energy lasts for 6.78 hours which is a 78% increment of battery life.

The detailed battery information and the *ECI* obtained from the experiment using the conventional method is shown in Fig. 3. During the service provision, with the decrement of the battery power which is represented by battery voltage, the battery charge decrease linearly and fast. The *ECI* decreases rapidly after service provision starts and stays low during the experiment.

In the experiment using proposed method, we set the ECI threshold θ to 20 as the trigger to reorganise the service provision agent organization. After service provision starts, with the decrement of the battery voltage, the battery charge decreases linearly and the *ECI* decreases rapidly as the same as the experiment using convectional method.

However, when the *ECI* reaches the threshold, the reorganization is performed by replacing the video encoding agent in member agents using alternate agent which uses another video encoder. The new video encoding agent encodes video with 75% resolution and requires less energy to provide the multimedia communication service. After the reorganization, a recovery of the battery voltage is confirmed from the result figure. The *ECI* recovers over the threshold and the battery charge decreases at a lower rate which is represented by the slope of the battery charge curve.

After the first reorganization, the *ECI* reaches the threshold again and system reorganises one more time to reduce the energy consumption of the portable device by changing the video resolution to 60%. The reorganization repeats four times in the experiment, and reduce QoS from 100% video resolution to 25% video resolution to keep the *ECI* over the threshold to save the battery energy of the tablet. When the *ECI* reaches the threshold at the fifth time, the current QoS is 25% video resolution, and there is no available alternative agent, which is able to provide the same encoding function with the required QoS, in the agent repository. Therefore no reorganization is performed and the *ECI* falls down under the threshold and battery charge decrease linearly.

Each time the reorganization is performed, the video streaming is paused for less than a second till the new encoder agent starts functioning. At that moment, the CPU load and network usage are at a relatively low level. As a result of that, the *ECI* increases and decreases momentarily as the peaks shown in Fig. 4.

It is noteworthy that in the process of reorganization, there is the trade-off between battery life and QoS of the provided service. The balance of that trade-off is controlled by the *ECI* threshold θ. The higher θ is, the earlier reorganization is performed i.e. the more energy consumption is reduced, the more QoS is reduced and vice versa. In practice, service providers are encouraged to simulate multiple times to investigate the appropriate θ values for particular applications.

In this research, the result of experiment 1 shows that the system functions correctly as designed. The rate of energy consumption is reduced by the control of the evolution mechanism

Table 3 Result of experiment 2: battery life (hours) of tablet 1.

count	conventional method	proposed method
1	3.46	4.65
2	3.43	4.59
3	3.51	4.64
4	3.43	4.52
5	3.41	4.50

Table 4 Result of experiment 2: battery life (hours) of tablet 2.

count	conventional method	proposed method
1	4.07	5.13
2	4.21	5.25
3	4.07	5.15
4	3.98	5.11
5	3.96	5.11

Table 5 Result of experiment 2: average battery life increments of tablet 1 and tablet 2.

average batter life (hours)	tablet 1	tablet 2
conventional method	3.45	4.06
proposed method	4.58	5.15
increment	32.75%	26.68%

autonomously before the battery charge of the tablet actually reaches low level. The QoS is maintained at the level which is able to satisfy requirements to the system. Therefore, proposed design is feasible for the system which contains portable devices as the example.

4.2.2 Experiment 2: Effectiveness Evaluation

This experiment is set up to evaluate the effectiveness of the proposed approach. The experiment system is similar to experiment 1, but using two Android tablets together simultaneously. The tablets in experiment 2 are different from the one in experiment 1 which has a larger battery. The video streaming provider continuously streams live video to both tablets in the system. For both the conventional method from the previous research and proposed method in this research, the identical experiment has been performed 5 times. The tablets are fully charged before every time of the experiment and play the live video back until the batteries are empty and devices are automatically shutdown. The battery life results of tablet 1 and tablet 2 are shown in **Table 3** and **Table 4**, respectively.

From the result tables, it can be observed that even under the identical experiment setting, the battery life of the devices varies in noticeable differences. In this particular experiment, the difference is more than half an hour for both conventional and proposed method. It is reasonable because the battery lives of devices depend on both hardware part and software part of the devices. However, the average battery life extension of both tablet 1 and tablet 2 represented in **Table 5** shows that proposed method effectively increases battery life of portable devices which have different hardware. In this particular experiment, the average increments of device battery life is around 30%. The proposed method is generally effective to reduce the energy consumption of portable devices to realise an optimised battery life. It is useful for the situation that the system contains heterogeneous portable devices which is usually the case in the real world. Similar effect can be expected for different types of portable devices in the system.

To summarise, once the appropriate ECI threshold θ has been decided, the proposed method is guaranteed to provide required services at best QoS that is possible while considering not only the QoS parameter but also the energy consumption of portable devices in the system. Through the experiment, we can confirm that by the reorganizations of the autonomous agents, the energy consumption rate of the portable devices is reduced before the battery charge reaches low level and a longer battery life is achieved. Therefore the solution to (P1) is provided. Also, during the experiment, the QoS is maintained to satisfy the requirements to the systems with the minimal effect on user experience of enjoying the service, therefore the solution to (P2) is provided.

5. Conclusion

Facing the situation that network services are being used on portable devices under unstable environment with variable limitations, this research aimed to provide adequate services to the portable devices flexibly by considering the situations of users and their devices while enjoying the services. As described in this paper, we proposed a new design of energy-consumption-aware Evolutional Agent System for portable devices. The proposal considers platform specific parameters of local resources in addition to QoS parameters to control the service provision by autonomous agent reorganizations. Moreover, to evaluate the feasibility and the effectiveness of our proposal, we implemented a multimedia communication system based on the proposed design. By the experiment results, through comparison to the conventional method, we confirmed that our proposal is able to provide adequate multimedia communication services with minimum effect to user experiences.

In the experiments of this research, the energy consumption factors are prefixed statically while calculating ECI. However, in the real world, those factors are possibly changing dynamically and unpredictably. For instance, the served video is able to be decoded with GPU instead of CPU, or the user decides to use speakers instead of earphones. In unpredictable dynamic situation, it is important to adapt the calculations of ECI to current energy consumption factors in real time. Furthermore, the adaption is preferred to be performed autonomously without troubling users. The proposed EAS model is suitable for the dynamic adaption of ECI calculation, however the practical method to realise it in the system still remains as a future challenge.

Acknowledgments This research was partly supported by a Grant-in-Aid for Young Scientists, Japan Society for the Promotion of Science, 22700087, 2013.

References

[1] Brakmo, L.S., Wallach, D.A. and Viredaz, M.A.: μSleep: A technique for reducing energy consumption in handheld devices, *Proc. 2nd International Conference on Mobile Systems, Applications, and Services* (*MobiSYS '04*), p.12, ACM Press (online), DOI: 10.1145/990064.990069 (2004).

[2] Qiu, M., Chen, Z., Yang, L.T., Qin, X. and Wang, B.: Towards Power-Efficient Smartphones by Energy-Aware Dynamic Task Scheduling, *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, pp.1466–1472, IEEE (online), DOI: 10.1109/HPCC.2012.214 (2012).

[3] Jennings, N.R.: On agent-based software engineering, *Artificial Intelligence*, Vol.117, No.2, pp.277–296 (online), DOI: 10.1016/S0004-3702(99)00107-1 (2000).

[4] Takahashi, A., Suganuma, T. and Kinoshita, T.: Dynamic Construction Scheme of Multimedia Processing Components Based on Multiagent Framework, *IPSJ Journal*, Vol.45, No.2, pp.366–376 (2004). (online), available from ⟨http://ci.nii.ac.jp/naid/110002712090/en/⟩

[5] Oyenan, W.H. and DeLoach, S.A.: Design and Evaluation of a Multiagent Autonomic Information System, *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (*IAT'07*), pp.182–188, IEEE (online), DOI: 10.1109/IAT.2007.56 (2007).

[6] Tesauro, G., Chess, D.M., Walsh, W.E., Das, R., Segal, A., Whalley, I., Kephart, J.O. and White, S.R.: A Multi-Agent Systems Approach to Autonomic Computing, *International Conference on Autonomous Agents and Multiagent Systems*, pp.464–471, IEEE Computer Society (online), DOI: 10.1109/AAMAS.2004.23 (2004).

[7] Juan, T. and Sterling, L.: A meta-model for intelligent adaptive multiagent systems in open environments, *Proc. 2nd International Joint Conference on Autonomous Agents and Multiagent Systems* (*AAMAS '03*), p.1024, ACM Press (online), DOI: 10.1145/860575.860777 (2003).

[8] Sansores, C. and Pavón, J.: An adaptive agent model for self-organizing MAS, *Proc. 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol.3, pp.1639–1642, International Foundation for Autonomous Agents and Multiagent Systems (2008). (online), available from ⟨http://dl.acm.org/citation.cfm?id=1402821.1402945⟩

[9] Luckey, M., Nagel, B., Gerth, C. and Engels, G.: Adapt Cases: Extending Use Cases for Adaptive Systems, *Proc. 6th International Symposium on Software Engineering for Adaptive and Self-managing systems* (*SEAMS '11*), p.30, ACM Press (online), DOI: 10.1145/1988008.1988014 (2011).

[10] Takahashi, A., Abe, M., Horikawa, N., Wei, W. and Kinoshita, T.: Design and Evaluation of System Margin in Evolutional Multiagent System, *The 1st International Workshop on Smart Technologies for Energy, Information and Communication*, pp.69–79 (2012).

[11] Takahashi, A., Abe, M., Wei, W. and Kinoshita, T.: Proactive Control Method Based on System Margin in Evolutional Agent System, *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp.64–68, IEEE (online), DOI: 10.1109/WI-IAT.2012.201 (2012).

[12] Takahashi, A., Suganuma, T., Abe, T., Iwaya, Y. and Kinoshita, T.: A behavioral characteristics model for flexible distributed system, *20th International Conference on Advanced Information Networking and Applications - Volume 1* (*AINA'06*), Vol.1, pp.6–280, IEEE (online), DOI: 10.1109/AINA.2006.10 (2006).

[13] Takahashi, A. and Kinoshita, T.: Configuration and control design model for an agent based Flexible Distributed System, *Web Intelligence and Agent Systems*, Vol.9, No.2, pp.161–178 (online), DOI: 10.3233/WIA-2011-0213 (2011).

[14] Kinoshita, T. and Sugawara, K.: ADIPS Framework for Flexible Distributed Systems, *Multiagent Platforms*, Ishida, T. (Ed.), Chapter 2, pp.18–32, Springer Berlin/Heidelberg (online), DOI: 10.1007/3-540-48826-X_2 (1999).

Wenpeng Wei received his B.E. degree in software engineering from Jilin University, China, in 2008, and M.S. and Ph.D. degrees in information sciences from Tohoku University, Japan, in 2011 and 2014, respectively. His research interests include knowledge engineering, agent-orientated software engineering, multi-agent software framework and its applications.

Akiko Takahashi received both a B.E degree in information engineering and a M.S degree in information science from Tohoku University in 2002 and 2004, respectively, and a Ph.D. degree from Tohoku University in 2007. She is currently an Associate Professor at Sendai National College of Technology. Her current research interests are intelligent agents, multiagent systems, and network middleware. Shi is a member of IPSJ.

Tetsuo Kinoshita is a Professor of Research Institute of Electrical Communication of Tohoku University. He received his B.E. degree in electronic engineering from Ibaraki University, Japan, in 1977, and M.E. and Dr.Eng. degrees in information engineering from Tohoku University, Japan, in 1979 and 1993, respectively. His research interests include agent engineering, knowledge engineering, knowledge-based systems and agent-based systems. He received the IPSJ Research Award, the IPSJ Best Paper Award and the IEICE Achievement Award in 1989, 1997 and 2001, respectively. Dr. Kinoshita is a member of IEEE, ACM, AAAI, IEICE, and JSAI.