International Journal of Informatics Society

08/13 Vol. 5 No. 2 ISSN 1883-4566

Informatics Society

Aims and Scope

The purpose of this journal is to provide an open forum to publish high quality research papers in the areas of informatics and related fields to promote the exchange of research ideas, experiences and results.

Informatics is the systematic study of Information and the application of research methods to study Information systems and services. It deals primarily with human aspects of information, such as its quality and value as a resource. Informatics also referred to as Information science, studies the structure, algorithms, behavior, and interactions of natural and artificial systems that store, process, access and communicate information. It also develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields. The advent of computers, its ubiquity and ease to use has led to the study of informatics that has computational, cognitive and social aspects, including study of the social impact of information technologies.

The characteristic of informatics' context is amalgamation of technologies. For creating an informatics product, it is necessary to integrate many technologies, such as mathematics, linguistics, engineering and other emerging new fields.

Guest Editor's Message

Nobutsugu Fujino

Guest Editor of Fourteenth Issue of International Journal of Informatics Society

We are delighted to have the fourteenth and special of the International Journal of Informatics Society (IJIS) published. This issue includes selected papers from the Sixth International Workshop on Informatics (IWIN2012), which was held at Chamonix, France, Sep 4-7, 2012. The workshop was the sixth event for the Informatics Society, and was intended to bring together researchers and practitioners to share and exchange their experiences, discuss challenges and present original ideas in all aspects of informatics and computer networks. In the workshop 28 papers were presented at eight technical sessions. The workshop was complete in success. It highlighted the lasts research results in the area of networking, business systems, education systems, design methodology, groupware and social systems.

Each paper submitted IWIN2012 was reviewed in terms of technical content and scientific rigor, novelty, originality and quality of presentation by at least two reviewers. From those reviews 15 papers are selected for publication candidates of IJIS Journal. This thirteenth includes five papers of them. The selected papers have been reviewed form their original paper presented in IWIN and accepted as publication of IJIS. The papers were improved based on reviewers' comments.

We hope that the issue would be interest to many researchers as well as engineers and practitioners in this area.

We publish the journal in print as well as in an electronic form over Internet. This way, the paper will be available on a global basis.

Nobutsugu Fujino is an industry-university cooperation consultant (self-employed) and a technical advisor to Alljos Entertainment Ltd., Japan. He received B.E. and M.E. from Osaka Prefecture University in 1984 and 1986, Ph.D. from Shizuoka University in 2008, respectively. He worked as a researcher and a research manager at FUJITSU Laboratories Ltd. from 1986 to 2013. His research interests include mobile and ubiquitous computing systems. He received IPSJ Industrial Achievement Award in 2003. He is a member of IPSJ and IEICE.

Stepwise Clustering Algorithm for Wireless Sensor Networks [1]

Shin-nosuke Toyoda[†], Fumiaki Sato[‡]

[†]Graduate School of Science, Toho University, Japan
[‡]Faculty of Science, Toho University, Japan
[†]6511013t@nc.toho-u.ac.jp
[‡]fsato@is.sci.toho-u.ac.jp

Abstract - Sensor networks consisting of nodes with limited battery power and wireless communications are deployed to collect useful information from the field. Gathering sensed information in an energy efficient manner is critical to operate the sensor network for a long period of time. LEACH is very energy-efficient routing protocol based on clustering of the sensor nodes. However, energy consumption of nodes tends to become uneven in LEACH. HEED improves the LEACH clustering algorithm by using information of residual electric power of nodes. Although HEED provides better performance than LEACH, it does not consider the number of adjacent nodes. Therefore, the cluster head does not efficiently cover the nodes in HEED. HIT and MR-LEACH are based on a small transmission range and multi-hop communication. Though these methods have improved the performance dramatically, unbalance of the electric power consumption is remained. In this paper, we propose energy-efficient clustering algorithm considering adjacent nodes and residual electric power. Characteristics of our approach are stepwise clustering from an initial cluster head and dynamic change of cluster size.

Keywords: sensor networks, stepwise clustering, energy-efficient routing.

1 INTRODUCTION

In recent years, there has been a growing interest in wireless sensor networks. Wireless sensor networks are composed of a large number of sensor nodes with limited energy resources. Energy efficiency is a key design issue that needs to be enhanced in order to improve the life span of the entire network. Usually, energy consumption can be divided into three domains: sensing, communication and data processing. Of the three domains, a sensor node expends maximum energy in data communication. One of the primary concerns with respect to sensor networks applications is the design and development of energy-efficient routing protocols that consume power more evenly, thus result into a prolonged network lifetime.

Available routing protocols for sensor networks are classified as data centric, location-based, QoS aware, and hierarchical. Data centric protocols use flooding or gossiping to transmit data [1-3]. Though the cost of routing is small, the number of data will be transmitted. Location based routing require the location information to determine an optimal path so that flooding of routing-related control packets is not necessary [4-6]. On the other hand, QoS aware protocols address various requirements such as energy efficiency, reliability, and real-time requirements [7]. Finally, the hierarchical protocols such as LEACH[8], HEED[9], HIT[10] and MR-LEACH[11] form clusters with cluster heads in order to minimize the energy consumption both for processing and transmission of data.

Clustering in Wireless Sensor Networks (WSNs) provides scalability and robustness for the network; it allows spatial reuse of the bandwidth, simpler routing decisions, and results in decreased energy dissipation of the whole system by minimizing the number of nodes that take part in long distance communication. LEACH is very energy-efficient routing protocol based on the clustering of the sensor nodes. In LEACH, non-cluster-head nodes first send their data to the cluster heads (CHs), and then CHs send the data to the base station (BS). Each link of non-cluster-head to CH and CH to BS is one hop. The cluster formation in LEACH is changed and CH is also changed periodically. Therefore, the load of CH is distributed all sensor nodes. However, energy consumption of nodes tends to become uneven in LEACH. On the other hand, HEED improves the LEACH clustering algorithm by using information of remaining electric power of nodes. Although HEED provides better performance than LEACH, it does not consider the number of adjacent nodes. Therefore, the CH does not efficiently cover the nodes in HEED. HIT and MR-LEACH are based on a small transmission range and multi-hop communication. Though HIT has improved the performance dramatically, unbalance of the electric power consumption is remained. Since MR-LEACH did not take account about the node coverage by CH, the effect of the cost reduction of CH was not so high.

To improve the life time of wireless sensor networks, we have proposed an energy-efficient clustering algorithm [12]. The algorithm selects CHs by using information of adjacent nodes and residual electric power. Sensor nodes are covered with few CHs and sensed data is transmitted to sink node by multi-hop communication. Therefore, the life time of the sensor networks is improved. However, because the cluster size is fixed in our previous work, some sensor nodes not covered by CH become single CH which is a problem of our algorithm. In this paper, we propose energy-efficient clustering algorithm considering adjacent nodes and residual electric power. The size of the cluster gradually grows from

[1] This work was supported by MEXT/JSPS Grant-in-Aid for Scientific Research(C) Grant Number (22500071).

a small size, and the algorithm can efficiently cover the sensor nodes.

The remainder of the paper is organized as follows. Section 2 summarizes related work. In Section 3 we present our clustering algorithm in detail. In Section 4 we show effectiveness of our algorithm via simulations and compare it to other clustering techniques. Finally, we conclude our paper and draw directions for future work in Section 5.

2 RELATED WORKS

2.1 LEACH

In this section, we described LEACH (Low-Energy Adaptive Clustering Hierarchy)[8], a clustering-based routing protocol that minimizes global energy usage by distributing the load to all the nodes at different points in time. LEACH is completely distributed, requiring no control information from the base station, and the nodes do not require knowledge of the global network in order for LEACH to operate. The key features of LEACH are:

1) Localized coordination and control for cluster setup and operation.

2) Randomized rotation of the “base stations” or “cluster-heads” and the corresponding clusters.

As a result, the load is distributed, and longevity on the entire network can be extended. Here, the “cycle” is the period which all nodes send the data once to the base station. The “round” is the period between the changes of CH.

All nodes can communicate to the base station directly in LEACH. All nodes know the probability p which each node try to become CH in the first round. When the round changes, node n decide whether try to become CH in the new round based on the equation (2.1). If a random number created by the node n is greater than the result of the equation (2.1), the node tries to become CH.

$$T(n) = \begin{cases} \dfrac{p}{1 - p*(r \bmod \frac{1}{p})} & if\ n \in G \\ 0 & otherwise \end{cases} \quad (2.1)$$

Here, the r is a number of rounds, G is the set of nodes which did not become CH in the 1/p past round (0<p<1). In other words, each node must become CH once in 1/p rounds.

The node which tries to become the CH sends the CH advertisement to neighboring nodes. The node which does not try to become the CH waits the CH advertisement during the fixed time. The node which receives the CH advertisement adds the node to the list of CHs with the RSSI (Received Signal Strength Indicator) of the node. When the waiting time is finished, non-cluster head node chooses the CH with the strongest RSSI among the list, and transmits the participation request. Data is transmitted directly to the sink without belonging to the cluster when there is no node which received the CH advertisement. On the other hand, the node transmitting the CH advertisement waits the participation request. When all the participation requests are received, the TDMA transmission schedule of the cluster member is made, and the CH transmits to the member. If the transmission schedule is received, the member memorizes the order of the transmission until the CH alternates. If the round changes, the process is executed for each round.

The cluster member transmits the sensor data to the CH in order on schedule after the schedule reception, and the CH compresses the data and transmits the data to the sink after data is received from all members. This is a flow of one cycle of one round in LEACH.

In LEACH, however, there is a problem that the power consumption of the node becomes unbalance easily. The reason is that the decision to become CH is based on only the frequency. Therefore, the node far from the sink node consumes energy early. There is the CH that no members exist in the cluster. There is the round that any nodes do not become the CH.

2.2 HEED

HEED (Hybrid, Energy-Efficient Distributed clustering)[9] is a clustering algorithm that improves the problems in the LEACH. The probability to become CH is based on the ratio of the initial electric power E_{max} and the current residual electric power $E_{residual}$ in HEED. Therefore, the node that has the more electric power is easier to become CH.

There are two states in the CH, the tentative CH and final CH. If the node broadcasts the final CH advertisement, the node serves the CH in the round. On the other hand, if the node broadcasts the tentative CH advertisement, the node may cancel the advertisement and join to other cluster that the total communication cost becomes small.

In HEED, the probability of the node that try to become CH (CH_{prob}) is given as follows.

$$CH_{prob} = \max\left(C_{prob} * \frac{E_{residual}}{E_{\max}}, p_{\min} \right) \quad (2.2)$$

Here, C_{prob} is the rate of the CH given beforehand. p_{min} is the minimum value of the CH_{prob}, that is decided in inverse proportion to E_{max}.

After calculating CH_{prob} by equation (2.2), each node repeats the following process. Flowchart is depicted in Fig.2.1.

(1)When one or more CH advertisements are received including own one:

The node that the communication cost is smallest is selected as CH.

If the node is myself:

a)If CH_{prob}=1, the node broadcasts final_CH message.

b) If CH_{prob}<1, the node broadcasts tentative_CH message.

(2) When no CH advertisement is received:

a)If CH_{prob}=1, the node broadcasts final_CH message.

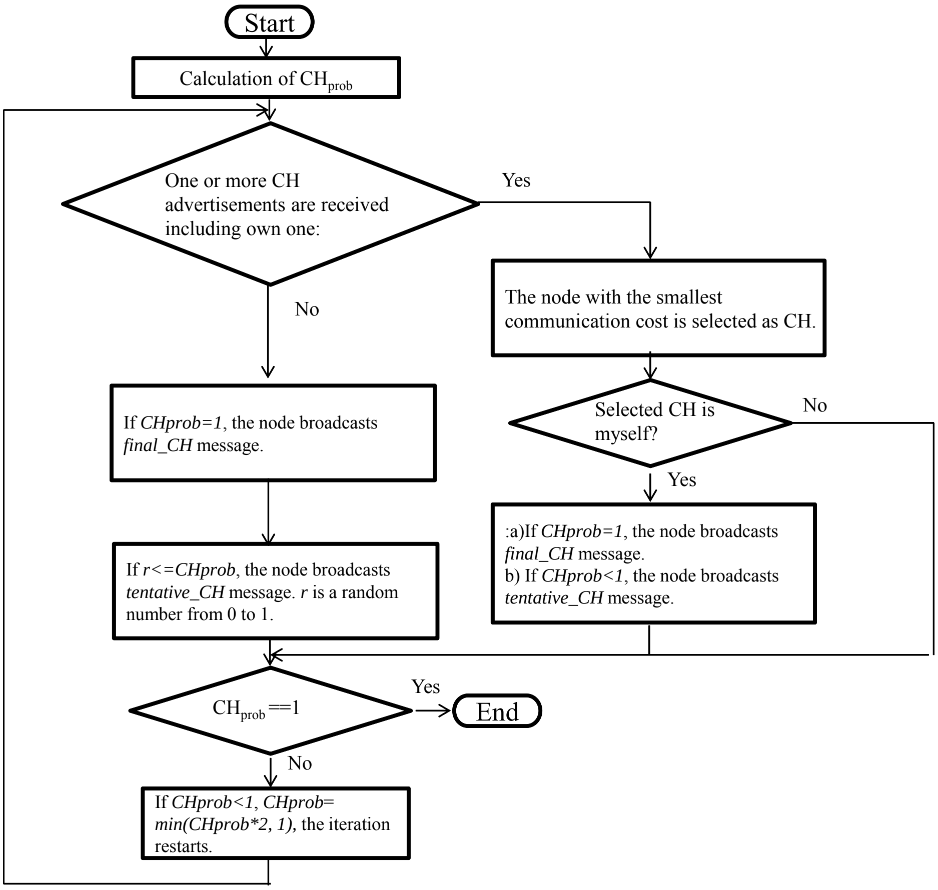

Start
Calculation of CH_{prob}
One or more CH advertisements are received including own one:
Yes
No
The node with the smallest communication cost is selected as CH.
Selected CH is myself?
No
Yes
If *CHprob=1*, the node broadcasts *final_CH* message.
:a)If *CHprob=1*, the node broadcasts *final_CH* message.
b) If *CHprob<1*, the node broadcasts *tentative_CH* message.
If *r<=CHprob*, the node broadcasts *tentative_CH* message. *r* is a random number from 0 to 1.
CH_{prob}==1
Yes
End
No
If *CHprob<1*, *CHprob= min(CHprob*2, 1)*, the iteration restarts.

Fig. 2.1 Flowchart of HEED.

b) If r<=CH_{prob}, the node broadcasts tentative_CH message. r is a random number from 0 to 1.

(3) If CH_{prob}=1, the iteration is end.

(4) If CH_{prob}<1, CH_{prob}= min(CH_{prob}*2, 1), the iteration restarts.

If the node does not broadcast the final_CH message in the iteration, the node selects the own CH from other nodes from which the node receives the final_CH message. If there is no node that the node receives the final_CH message, the node becomes CH and broadcasts the final_CH message.

2.3 HIT

HIT (Hybrid Indirect Transmissions) [10] uses multi-hop communication to control electric wave interference and to reduce the electric power consumption. It is effective to support parallel communication. HIT consists of the following seven phases.

(1)Phase 1: CH selection

In this phase, one or more CHs are selected. Each cluster has one CH. In case of single cluster, CH can be rotated based on the node ID.

(2)Phase 2: CH advertisement

In this phase, the selected CHs broadcast the node information as the Advertise message. The node j which is not CH and receives the message calculates the distance from the CH and joins to the nearest cluster. The node j has the distance to node H (CH) as the d(H, j).

(3)Phase 3: Cluster set up

In this phase, one or more clusters are created and relation of upstream/downstream are set up. At first, the node j which is not CH broadcasts Member message that includes the CH and distance to CH. By this exchange of information, all nodes calculate the distance to other node and keep the information to the distance to CH of other nodes. From this information, the upstream node u of node i is calculated by that information base on the following condition.

1)d(u, H)<d(i, H)

2)d(i, u)<d(i, H)

The condition 1) means that the transmission cost to the upstream node is smaller than to the CH. The condition 2) means that the upstream node is nearer to the CH than node i.

(4)Phase 4: Route set up

All nodes broadcast the Upstream message that includes the distance to the upstream node after the decision of the upstream node in Phase 3. All nodes are notified that the all upstream nodes of all nodes by this message. All nodes can set up the downstream node set.

(5)Phase 5: Blocking set calculation

In this phase, the node j which is blocked by node i is calculated, when node i transmit to the upstream node. The condition is as follows.

d(i, ui)>d(i, uj)

Now, ui is the upstream node of i and uj is the upstream node of j.

The nodes that satisfy the condition are called as the Block node list. Each node broadcasts the list. The node that the message receives makes the Block table that is the node set blocks to transmit to the upstream node.

(6)Phase 6: TDMA scheduling

In this phase, each node calculates the TDMA schedules that maximize the parallel communication that avoid the collision.

(7)Phase 7: Data transmission

In this phase, each node senses the environment and transmits the data based on the TDMA schedule made by the previous phase.

2.4 MR-LEACH

MR-LEACH [11] uses multi-hop communication to reduce the electric power consumption. In this method, the residual battery power is considered to select the CH node. There are two phases to construct clusters in this method. First phase is a cluster construction phase and second phase is a route construction phase.

(1)Cluster construction phase

At first in the round, all nodes in the network exchange a Hello message with the nodes in the area of distance *r*. In the message, the sender node ID and residual battery power are included. Each node receives the Hello messages from neighboring nodes and manages this information. After the exchange of the Hello message, each node compares the residual battery power. If a node has the maximum battery power in the neighboring nodes, the node becomes CH and sends CH advertisement message. In the message, the node ID of the CH is included. If non-CH node receives the CH advertisement message from neighboring node, the node ID and the strength of received signal are stored in the memory. After receiving some advertisement messages, the non-CH node select the CH which has the strongest received signal.

(2)Route construction phase

At first in this phase, the sink node broadcasts the confirmation message to all nodes. The all CH nodes which receive the confirmation message broadcast the response message within the range of *r'* (*r'*>*r*).

The sink node receives the response messages and stores the ID of CH node to the node list of the layer 1. In a word, the list of layer 1 is the list of CH in the range of *r'* from the sink. Next, the sink node broadcasts the confirmation message including the node list of layer 1.

The CH node which receives the confirmation message confirms whether this node is included in the list. If the node included in the list, the node understands that the node is included in the layer 1. Otherwise, the node broadcasts the response message within the range of *r'*.

If CH in layer 1 receives the response from other CH, it sends the node ID of the response message to the sink. If the sink receives the message, it stores the node ID to the node list of the layer 2.

After receiving all layer 2 node IDs, the sink node broadcasts the confirmation message including the node list of layer 1 and 2. Then, the CH which is not included in the layer 1 and 2 responses the message.

This process is iterated until all CH doesn't respond. As a result, a hierarchical cluster that centers on the sink is composed like Fig.2.2.

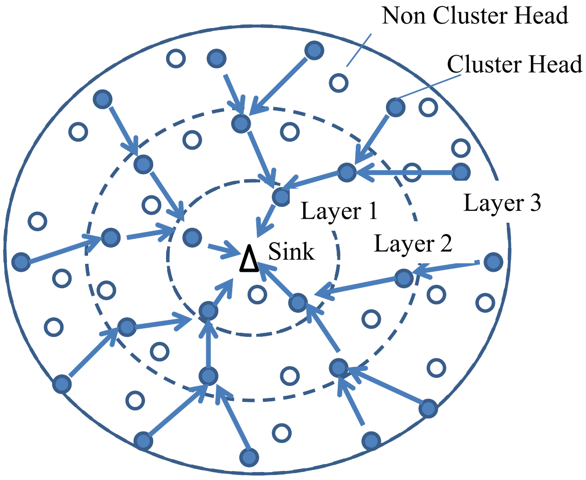


Fig.2.2 Route construction in MR-LEACH.

2.5 Other related works

There is PEGASIS[13] as one of the other methods. PEGASIS uses the chain structure instead of the cluster. TPC[14] uses the chain structure in the intra-cluster communication. These methods are based on location information. Because our method is not based on location information, these methods are not compared in this paper.

RPL[15] is a routing protocol for low power and lossy networks. Security improvement of RPL is proposed in [16]. Performance of data gathering is improved in [17]. These protocols are offering routing based on the reliability of the link. Because usual reliability is assumed in the link in our method, these methods are not compared in this paper.

3 CLUSTERING ALGORITHM

3.1 Basic concept

In this section, we propose the clustering method to consider the adjacent node set and the residual electric power. In the proposal method, all nodes other than the sink exchange the Hello message of each round, which contains information on own residual electric power and the adjacent node set. As a result, each node can maintain information on the adjacent node set and the residual electric power for the nodes.

At first, the sink selects the first CH. Other CHs are selected radially by the first CH to cover the surrounding nodes. To prevent flooding of the CH, the CH is selected to cover a lot of nodes. CH has been selected like evenly consuming the electric power by considering the amount of the electric power remainder. The transmission power is saved as the small range for collection of the sensor data. The range is controlled by the sink node. The collection of the sensor data from the node which cannot communicate with the sink node directly becomes possible by using multi-hop communication of CHs.

The transmission power used in the clustering phase is small at first. The transmission power means the size of the cluster. In our algorithm, some sizes are prepared to the cluster. If all nodes in the network are not covered by any clusters, the size of the cluster is enlarged and clustering is executed again.

3.2 Cluster head selection

In this section, the algorithm for CHs selection in a round is explained. It is designed by modifying the algorithm for the landmark node selection in ad hoc networks[17]. The following is the process of the algorithm.

(1)Hello message exchange phase.
(2)Representative node selection phase.
(3)CH selection phase.
(4)End report phase.

Each phase is explained as follows.

(1) Hello message exchange phase.

When a new round begins, the sink node broadcasts the message that request to exchange the Hello messages with each other. This request message includes the maximum transmission range R in this round.

Each node broadcasts the Hello message includes the node ID and residual electric power after receives the request message. Each node receives the Hello message from other nodes and constructs the adjacent node list (ANL) which includes the adjacent node IDs and residual electric power (REP) of them. The adjacent node means the node which exists within the range R. Each node broadcasts the second Hello message which includes the ANL after the first Hello message. Each node receives the second Hello message and constructs the two-hop adjacent node list (TANL). TANL means the list of the nodes which can be reached in just 2 hops from the node.

(2) Representative node selection phase.

After the Hello message exchange phase, each node selects the representative node. The node which has the largest REP is selected as the representative node. Because all nodes receive the Hello message, all nodes can learn who selected as a representative node. Now, the representative node is the one of the CHs.

(3) CH selection of representative node.

After the fixed time, representative node L starts the selection of the other CHs. L calculates evaluation value v_n for all nodes included in the ANL. Here, evaluation value v_n of adjacent node n is calculated by the following equation (3.1) by using c_n: the number of overlapping nodes between adjacent nodes of L and adjacent nodes of n, e_n: the residual electric power of node n, and e_{ave}: the mean value of the residual electric power of all adjacent nodes of L.

$$v_n = \frac{1}{c_n} * \left(\frac{e_n}{e_{ave}} \right)^w \qquad (3.1)$$

Here, the w is a constant which shows the weight of the residual electric power. In a word, the evaluation value rises in the node that the number of overlapping node between adjacent nodes of n and adjacent node of L is small, and the residual electric power is large. The node n1 with the largest evaluation value is selected as the one of the CHs.

The next CH would be selected if the 2-hop-coverage is smaller than the threshold. The ratio 2-hop-coverage means the ratio between the number of TANL of L and the number of node covered by the n1. L calculates evaluation values v_n of all adjacent nodes n except n1 again. v_n is calculated for the set of nodes which excluded common part with adjacent node of n1 from TANL of L, that is called as non-covered node list thereafter. v_n is calculated by the following equation (3.2) using d_n: the number of overlapping nodes between non-covered node and the adjacent nodes of n, e_n: the residual electric power of n, e_{ave}: the average residual electric power of the adjacent all node of L except n1.

$$v_n = d_n * \left(\frac{e_n}{e_{ave}} \right)^w \qquad (3.2)$$

Here, the evaluation value rises in the node that the number of overlapping node between adjacent nodes of n and non-covered node is large, and the residual electric power is large. The node n2 that has the biggest evaluation value is selected to be the next CH as well as n1, and the adjacent node of n2 is deleted from the list of non-covered nodes. And, if 2-hop-coverage does not exceed the threshold, the next CH is repeatedly chosen until the threshold of 2-hop-coverage is exceeded.

When representative node L finishes the selection of the CH, it broadcasts the CH advertisement in the range R. The CH advertisement includes information on the selected CHs and non-covered node list. The non-covered node list in the advertisement is called as the 3-hop-check-list. Adjacent node n of L which receives the CH advertisement of L adds L to the adjacent CH list of oneself. If n is selected as the CH, L is called as the n's parent CH and n starts to select the next CH with the same process.

When the selection of the CH is finished, node n creates 3-hop-check response for 3-hop-check-list from the parents CH. As for 3- hop-check response, the node which can reach by two hops from n via the adjacent CH of n is stored, which is included in the 3-hop-check-list. The adjacent CHs include the CH that n newly selects. In a word, it becomes nodes which can reach by three hops from the parents CH. After the calculation, n broadcasts the CH advertisement in the range R. The CH advertisement includes information on the selected CHs, 3-hop-check-list of n, and 3- hop-check response to parents CH.

After transmitting 3-hop-check-list, CH n waits 3-hop-check response during the fixed time. n deletes the node included in the non-covered node list, that found in the 3-hop-check response. If the fixed time ends and 3-hop-check-list does not empty, the evaluation value of the node included in the adjacent node list of n is calculated again based on expression (3.2). Node n2 whose evaluation value is the highest is newly chosen to be a CH, and 3-hop-check-list is transmitted to n2 on CH-request message.

CH n2 selects the CH according to the procedure of (3) when this is received, and broadcasts CH advertisement including 3-hop-check response to n. When 3-hop-check-list does not empty even if n receives this, CH n repeats these processes until 3-hop-check-list empties.

(4)End report phase.

When the non-covered node list empties, and the CH is not selected newly, CH n transmits the end report of the CH selection to the parents CH. In addition, when the end report is received from the all child CHs, and the non-covered node list empties, the parents CH transmits the end report to the upper parents CH. Thus, all nodes on the network can belong to either of CH when the end report is forwarded, and the sink receives the end report. Therefore, the selection of the CH is ended now.

When the selection of the CH ends, the sink node broadcast the request to participate to a cluster. The node which received this makes the CH with the strongest RSSI of the CH advertisement a parents CH among lists of the adjacent CH, and transmits the participation request. On the other hand, in case of the child CH of the CH receives the request to participate to a cluster, the child CH rebroadcast the message. As a result, all nodes will participate to either of CH.

The CH makes the cluster member's data transmission schedule and broadcasts it after the fixed time later of broadcast of the request to participate to a cluster. When the schedule is received, the child node maintains the order of the transmission until the round changes.

However, there is the case that some nodes do not belong to any clusters because the transmission range R is too small. In this algorithm, the representative node enlarges the transmission range R and restarts the clustering.

Message sequence chart of proposed algorithm is shown in Fig.3.1. Representative node in Fig.3.1 is indicated as CH1. CH2 and CH3 are selected CH by CH1 and CH2 respectively. CHx is extra selected CH by CH1 after receiving CH advertisement from CH2. CHy is selected CH by CHx.

3.3 Collection of sensing data

The timing of data collection is notified by the data request message from the sink node. If the child CH node of the sink receives the data request, the CH rebroadcasts the message to the cluster members. If there is CH node in the cluster, it rebroadcasts the message. The message is spread in all nodes by repeating this process.

If a CH node receives all sensing data from all member nodes and child CHs, it compresses these data and own data, and send to the upper CH. The parent CH also sends the data to the upper CH similarly.

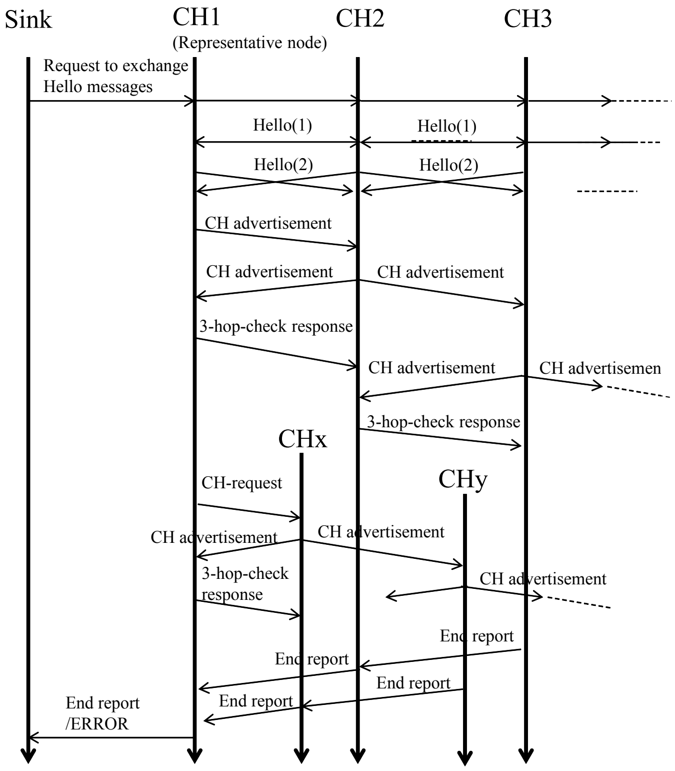


Fig.3.1 Message sequence chart of proposed algorithm.

Thus, the sensor data that all nodes collected from the end of the network is collected in the sink. When the sink finishes collecting all data, one cycle is completed. After some cycles are repeated, it moves to the next round.

4 SIMULATION

Simulation program written in Java is used to evaluate the LEACH, HEED, HIT, MR-LEACH and proposed method. Neither a physical layer nor the MAC layer are included in the simulation program. The situation that the sensors are scattered to the observation area that the person cannot enter is assumed. The sink node is out of the observation area. The assumed observation area is shown in the figure as follows.

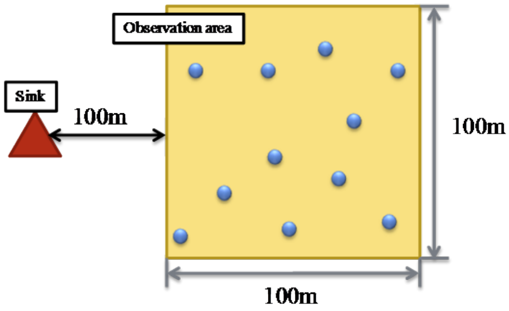


Fig.4.1 Simulation environment.

4.1 Simulation environment

In the simulation environment, the observation area is 100m x 100m and the sink node is located at the point that 50m to south and 100m to west from the northwest end point of the observation area. Maximum transmission range of the node is 150m. In the proposed method, 25m is used CH selection process in the observation area.

Power consumption model of the transmission and reception is used in [1]. Consumed power E_T for k bits transmission to the node that d m away from sender is expressed in the equation (4.1). The Consumed power E_R for k bits reception is equation (4.2).

$$E_T = E_{elec}k + \varepsilon_{amp}kd^2 \tag{4.1}$$

$$E_R = E_{elec}k \tag{4.2}$$

Here, E_{elec} is the consumed power to send/receive 1 bit, ε_{amp} is the consumed power to send the data.

The simulation is executed until all nodes exhaust the electric power. After the simulation, the number of cycles, the maximum and average residual electric power is compared. Common parameters used in the simulation are shown in Table 4.1.

Table 4.1 Common parameters.

Parameter	Value
E_{elec}	50nJ/bit
ε_{amp}	100pJ/bit/m^2
Control message size	500bits
Data size	2000bits
Number of nodes	100
Max transmission range	150m
Step of transmission range R	20m, 30m, 40m, 150m (4 steps)

The number of cycles in a round and the probability of the node to try to become CH should be decided in LEACH. From the preliminary simulation to decide the parameters, the number of cycles and probability is decided as 10 and 0.05 respectively.

The number of cycles in a round and the probability of the node to try to become CH (CH_{prob}) should be decided in HEED. From the preliminary simulation like as LEACH, the number of cycles and probability CH_{prob} is decided as 10 and 0.1 respectively.

In HIT, the number of cycles in a round is decided as 250 from the preliminary simulation.

The number of cycles in a round is 50, the threshold of 2-hop-coverage is 0.7, and the weight *w* of the residual electric power is 2.0. These parameters are obtained from the preliminary simulation. Moreover, there are some parameters in our and previous algorithms are selected the best values obtained from the preliminary experiments.

At first, the transmission range R is set as 20m, the smallest size. If there are any nodes which are not included in any clusters, the range R is changed to the next size and the clustering process restarts. The maximum size of R can cover the all nodes in the field.

4.2 Simulator model

The simulator used for our evaluation is based on the discrete event simulation. It is able to set up the arrangement of nodes and reproduce the reachability of message frame based on the distance between nodes. The power consumption model used in LEACH is also implemented.

In this method and the simulation environment, transmitting power is required to be controllable for the appropriate strength to the specified distance between nodes. The according to the received signal strength, the distance between sender node and receiver node can be estimated. These assumptions are used in the previous researches, and are not peculiar to our method.

Moreover, our simulator doesn't include the model of the packet loss caused by the interference or noise. In a word, the frame will be correctly sent and received without packet loss if the nodes in the transmission range of RF. The reason is that we would like to compare the pure cost to construct the cluster and battery power efficiency to collect the sensed data. Because the properties of packet loss and collision depend on the physical layer and MAC layer, we thought that the ideal environment is a best way to compare the algorithms. Moreover, I might think that I can reduce the packet loss and collision by using the wide inter-frame interval or the TDM based schedule. The message cost to cluster in an ideal environment that doesn't contain the frame loss or collision properties can be evaluated by using this simulator. The power consumption for the message transmission and reception also can be evaluated.

4.3 Simulation results and discussion

The first simulation result is the number of node alive until the all nodes exhaust the electric power (Fig.4.2). From the simulation results, the number of node alive of proposed method becomes better performance than other methods.

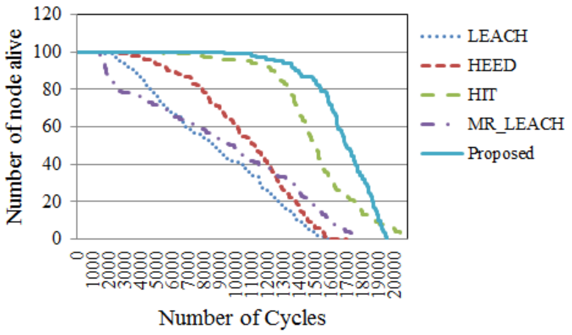


Fig.4.2 Number of node alive.

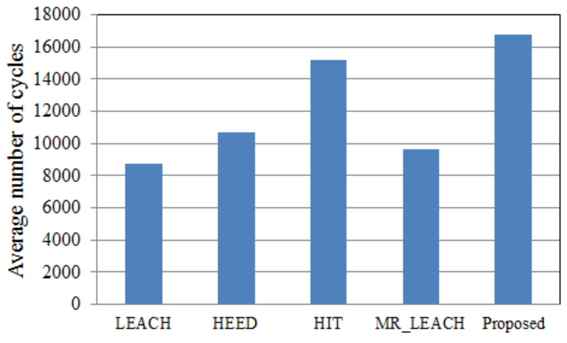


Fig4.3 Average Number of cycles.

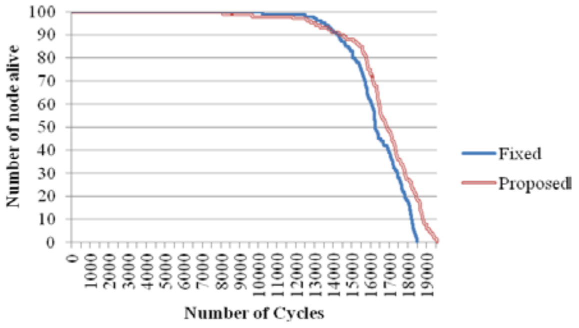


Fig.4.4 Number of node alive.

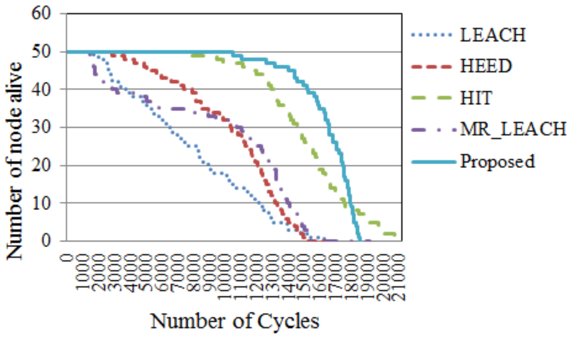


Fig.4.5 Number of node alive.(node = 50)

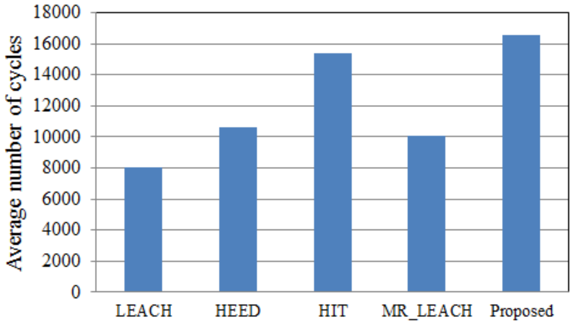


Fig4.6 Average Number of cycles.(node = 50)

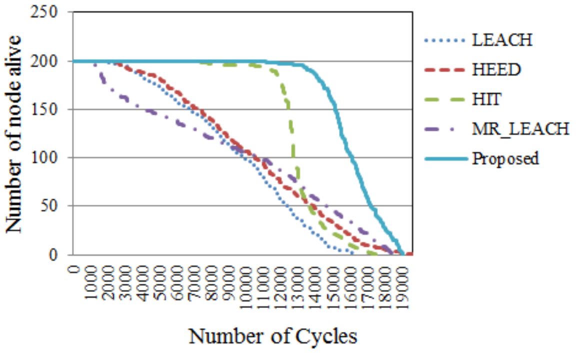


Fig.4.7 Number of node alive.(node = 200)

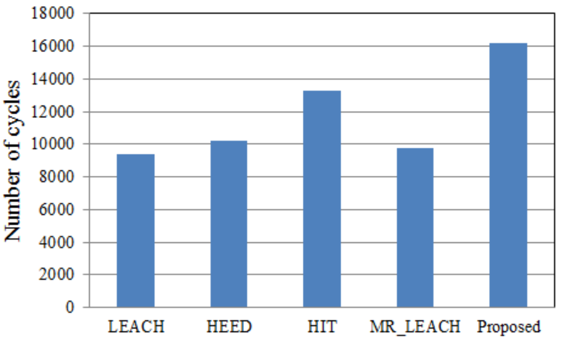


Fig4.8 Average Number of cycles.(node = 200)

Around 1700 cycle, the number of node alive becomes smaller than HIT. However, the number of cycles that 80% node alive becomes 25% longer than HIT. It means that the life time of the sensor network is improved.

Figure 4.3 shows the average number of cycles of node until the all nodes exhaust the electric power. The result of the proposed method is about 10% better than HIT.

Figure 4.4 shows the comparison between the fixed transmission power and the changed transmission power. The changed transmission power proposed in this paper is better than fixed one.

Figure 4.5 to Fig.4.8 show the results of the number of node alive and average number of cycles until the all nodes exhaust the electric power. The number of node is 50 and 200 respectively. Our algorithm shows a good result in each number of nodes. Especially, there is a difference with other methods remarkably when there are a lot of nodes in the network.

5 CONCLUSIONS

In this paper, we proposed energy-efficient clustering algorithm considering adjacent nodes and residual electric power. In addition, we inspected effectiveness of our method by comparing our method with the traditional method by the simulation.

As a result, proposed method showed higher performance than LEACH, HEED, HIT and MR-LEACH. The number of cycles that 80% node alive becomes 125 % of HIT algorithm.

However, the evaluation was executed in an ideal environment. The communication interference should be discussed in the real environment. Especially, the packet loss by the interference is the problem in our method. In our algorithm, the exchange of the Hello packet is the most critical phase we think. In our future work, the phase will be changed from broadcast to flooding of Hello messages.

On the other hand, we think the difference of shape and the size of the range will not make a big influence. Because the clustering process uses the hand-shake process, the uni-directional link based on the difference of the range can be avoided.

Furthermore, parameters used in our algorithm like as *w* which is obtained from the preliminary experiment should be selected from experiment in the real environment.

Therefore, future work of our research is detailed evaluation based on the well-known simulator which includes a physical layer model and MAC layer model. Improvement of our algorithm to consider the coverage of the sensor area is also our future work.

REFERENCES

[1] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '99), pp. 174-185 (1999).

[2] C. Intanagonwiwat, R. Govindan, D. Estrin, and J. Heidemann, “Directed diffusion for wireless sensor networking, ” IEEE/ACM Transactions on Networking (TON), Vol. 11, Issue 1, pp. 2-16 (2003).

[3] F. Ye, G. Zhong, S. Lu, and L. Zhang, “GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks,” ACM Wireless Networks, Vol. 11, pp. 285-298 (2005).

[4] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” Proceedings of the 9th international conference on Architectural support for programming languages and operating systems, pp. 93-104 (2000).

[5] J. Zhang, Z. Yang, B. Cheng, and P. McKinley, “Adding Safeness to Dynamic Adaptation Techniques,” Proceedings of the ICSE 2004 Workshop Architecting Dependable Systems, pp. 17-21 (2004).

[6] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, “TTDD: A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks,” Proceedings of the 8th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '02), pp. 148-159 (2003).

[7] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, “SPEED: a stateless protocol for real-time communication in sensor networks,” Proceedings of the 23rd International Conference on Distributed Computing Systems, pp. 46-55 (2003).

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proceedings of the 33rd Hawaii International Conference on System Sciences, pp. 1-10 (2000).

[9] O. Younis, and S. Fahmy, “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks,” IEEE Transaction on Mobile Computing, Vol. 3, No. 4, pp. 366-379 (2004).

[10] B. J. Culpepper, L. Dung, and M. Moh, “Design and Analysis of Hybrid Indirect Transmissions (HIT) for Data Gathering in Wireless Micro Sensor Networks,” ACM Mobile Computing and Communications Review, Vol. 3, pp. 61-83 (2004).

[11] M. O. Farooq, A. B. Dogar, and G. A. Shah, “MR-LEACH: Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy,” Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM 2010), pp. 262-268 (2010).

[12] S. Toyoda, and F. Sato, “Energy-Effective Clustering Algorithm Based on Adjacent Nodes and Residual Electric Power in Wireless Sensor Networks,” Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 601-606 (2012).

[13] S. Lindsey, C. Raghavenda, and K. M. Sivalingam, “Data Gathering Algorithms in Sensor Networks Using Energy Metrics,” IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 9, pp. 924-935 (2002).

[14] W. Choi, P. Shah, and S. K. Das, “A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks,” Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 203-212 (2004).

[15] RPL, “IPv6 Routing Protocol for Low power and Lossy Networks,” ROLL IETF Internet-Draft (2011).

[16] A. Dvir, T. Holczer, and L. Buttyan, “VeRA - Version Number and Rank Authentication in RPL,” Proceedings of the IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS 2011), pp. 709-714 (2011).

[17] B. Pavkovi, F. Theoleyre, and A. Duda, “Multipath opportunistic RPL routing over IEEE 802.15.4,” Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '11), pp. 179-186 (2011).

[18] [18] J. Ushijima, M. Okino, T. Kato, and S. Ito, “Proposal and Evaluation of a Selecting Method of Landmark Nodes for Message Relaying over High Density Ad hoc Network,” IEICE technical report 103(202), pp. 93-96 (2003).

(Received September 27, 2012)
(Revised March 9, 2013)

Shin-nosuke Toyoda received BSc and MSc from Faculty of Science, Toho University in 2011 and 2013. He joined Japan Process Development Co., LTD in 2013. His research interest includes the wireless sensor networks and mobile computing. He is a member of Information Processing Society of Japan (IPSJ).

Fumiaki Sato received BE from Iwate University in 1984. He received ME and DE from Tohoku University in 1986 and 1992 respectively. He joined Mitsubishi Electric Corporation in 1986. He served as Associate Professor in Shizuoka University from 1996. He is Professor of Toho University from 2005. His research areas of interest include communication protocols, ad-hoc networks, sensor networks, P2P systems, network security, distributed processing systems, and communication software design. He is a member of IPSJ, IEEE, and ACM.

Design and Development of a Security Evaluation Platform Based on International Standards

Yuji Takahashi and Yoshimi Teshigawara

Graduate School of Engineering, Soka University, Japan
{e08d5203, teshiga}@soka.ac.jp

Abstract - To obtain security attestation, organizations evaluate security products by using systems based on international standards. However, they currently must use individual systems corresponding to different versions of these standards. Therefore, we have been studying a platform that enables evaluation for different standard contents and evaluation targets by focusing on changes of the standards used as evaluation criteria. We developed and implemented the platform taking into consideration the hierarchical structure and reference relations of the standards. The platform provides functions such as a reference-related arrangement of the whole standard, the display of a reference tree, and score calculation. In addition, in order to produce the pertinent information for data conversion, we calculated the similarity between two standards. Experimental evaluation shows that covering all items and avoidance of human error can be achieved by supplementing technical knowledge and by utilizing visual effects. The validity of the platform is also confirmed.

Keywords: Security management, Information security, International standard, ISO/IEC 27000

1 BACKGROUND AND PURPOSE OF RESEARCH

In recent years, the scope of security management is expanding from self-defense for protecting the assets of an organization to preventing becoming the target of attackers who cause damage to the organization. As a result, it has become important to have the status of the implementation of safety and security measures assessed by an external agency [1]. There is a specific standard for such assessment, called ISO/IEC 27001 and the number of organizations that are being accredited by this standard is continuously increasing. By June 2012 more than 7,000 companies had been accredited worldwide, more than 4,000 of them in Japan [2].

For most of the security certifications, standards such as ISO/IEC 27001, ISO/IEC 27002, and JIS Q 15001 are taken as references and organizations are accredited by satisfying all the items that are described in those standards. In addition, security assessment systems are used to validate the achievement of criteria in the certification process [3]. However, the items of the standard are frequently changed as time passes. Compared with other standards, security related standards are changed more frequently because they are not tested precisely; user comments are taken into consideration and changes are made accordingly. In addition, because the certification process differs depending on the size of the organization and other factors, the criteria for assessment also differ. If the organization and the objective of the assessment change, changes such as revision of the standard will create a situation where a new system has to be created for redoing each certification using individual tools or personnel. Hence considerable time, personnel and money are required and this leads to problems that have huge personnel and monetary impacts on company activities. The need for a general assessment tool, allowing changes in the organization and the purpose of assessment, in place of individual security assessment tools, has been increasing.

We have been studying a security assessment platform that enables the realization of particular security assessments by replacing only raw data (hereinafter referred to as fundamental data) that address the fundamental standard without depending on the target standard [4]. In this platform, by focusing on the hierarchical structure of sentences, which is a characteristic structure of standards documents, the items of the standard as well as statements indicating the detailed conditions and references to other items (hereinafter reference relations) were organized in a hierarchical structure. Security assessment needs to be done without depending on the type of reference standard, and a method of estimating the assessment level without depending on the type of standard is required. In the platform the assessment level is estimated using the hierarchical structure and reference relations, so we have been studying a platform that aims to be suitable for achieving security requirements. In particular, we have developed an appropriate platform system and registered the data for the ISO/IEC 27000 series [5]. In this study, security assessment is conducted by changing the impact of assessment with respect to each component of the reference tree as described below. On the basis of experimental results for security assessment methods that consider the distance in the reference trees as well as the relation of each item with assessment items, we found that changing impact is effective. For considering the relation of each item with assessment items, we proposed various methods of estimating impact and experimented using these methods [5] [6] [7]. Thus, for those users who do not have a deep knowledge of attestation, we experimented with changing the sample providing function and the data migration function, using relevant information based on past cases, in order to support countermeasure selection and implementation, and we confirmed the validity of the proposed platform [7] [8]. We found that the effectiveness of the data migration function can be increased by interlocking with the sample function

ISSN1883-4566 © 2013 - Informatics Society and the authors. All rights reserved.

[8]. In this paper, we evaluate the results of the experiments done for each function separately and in combination.

2 ANALYSIS AND UTILIZATION OF STANDARDS

2.1 Relevant standards

In this paper, the experiments and verifications are performed mainly using the security standard data that have been summarized as the ISO/IEC 27000 series. This has adapted the concept of the Plan-Do-Check-Act (PDCA) cycle, which is widely used in the standards of security management and is represented in information security management systems (ISMSs).

This security assessment platform is intended to be used not in a single phase of the PDCA cycle but in every phase where the platform is applicable. If it is applied in the Plan stage, the loopholes in the countermeasures can be checked by entering the results of the present data analysis. In the Do stage when it is recognized that enforcing countermeasures does not cover the planning item, the loophole can be verified in its entirety by means of checking those items. In the Check stage, the functionality of each countermeasure can be checked according to the plan made in the countermeasure enforcement stage. The loopholes can be checked by summing those changes in the corresponding conditions that match the conditions in practice. In the Act stage, as in the Plan stage, the loopholes of the corresponding countermeasures that were re-defined can be checked.

2.1.1. ISO/IEC 27000-series

The ISO/IEC 27000-series is an information security standard family, established by the collaboration between the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). This series is broad in scope, covering privacy, confidentiality and information technology security issues. Therefore, it is applicable to organizations of all sizes and types.

To obtain security attestation in this series, organizations first assess their information security risks and then implement appropriate information security controls according to their needs. Given the dynamic nature of information security, the ISMS concept incorporates continuous feedback and improvement activities based on the PDCA cycle. As of June 2011, 10 standards of ISO/IEC 27000 had been developed and many other standards are now under development [9]. ISO/IEC 27000 is a standard reference in many areas and it shows the importance of the PDCA cycle to ISMSs.

2.1.2. ISO/IEC 27001

The objective of ISO/IEC 27001 is to provide a model for the establishment, implementation, operation, monitoring, review, maintenance and improvement of an ISMS [10]. In addition, the contents shown in each item of this standard in the operational manual created during the process of ISMS attestation, corresponds to the security requirements. It should cover all the items including those that specify what is outside the scope of the attestation. During the inspection for ISMS attestation, the security countermeasures corresponding to each item of this manual will be subject to inspection.

2.2 Standard configuration

Generally the body in the relevant standard has often been described in a hierarchical structure of three phases: 'Chapter', 'Section' and 'Item'. In a 'Chapter', the assessment targets are roughly classified. In a 'Section' the assessment targets are described in detail and in an 'Item' the contents are further described in more detail.

However, there are many individual items which are not only described as separate items but also as conditions or supplementary matters that refer to other items. For instance, Section 7.1 of ISO/IEC 27001 contains a reference to Item 4.3.3 and this relationship is expressed in the reference tree used in this study, as shown in Fig.1.

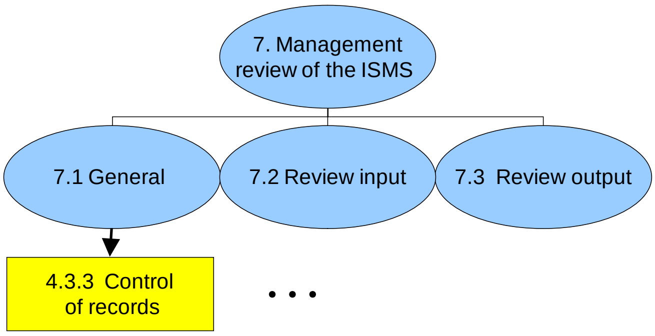


Figure 1: Reference-related example of ISO/IEC 27001

2.3 The problems of covering items related to countermeasures and their solution

In security attestation, the criteria should be comprehensively covered. Depending on the framework of each chapter of the configuration, the required policy decisions, such as implementation of countermeasures and acceptance of the risk, will be made. At that time, since there is a need for the comprehensive cover of the standard for each chapter, it is necessary to capture precisely the hierarchical structure of each chapter and reference relations for each item.

However, in all standards, not just ISO/IEC 27001, there are many references and there is a wide variety of content (items) which should be covered. Hence, understanding all of them precisely and choosing comprehensive measures becomes difficult. Therefore, it is desirable to manage collectively all the items covered in each chapter. To solve these issues, we propose a platform that can collectively manage all the items to be covered by using the hierarchical structure and the reference relations. Since the hierarchical structure and reference relations that are described in the platform describe information from the standard with similar characteristics, the platform can cope if there is a change in the standard or even if the standard is a different one.

3 OVERVIEW OF THE PLATFORM

3.1 Structure of the platform

This platform consists of three parts namely, the data input unit, the data management unit and the score calculation unit. The configuration of the platform is shown in Fig.2. In the data input unit, the fundamental data of the standard, structural information, reference information, countermeasure information and other relevant information are entered. Initially the input of countermeasure information can be based on sample information created by the data management unit. Based on these fundamental data and the structural information, the data management unit organizes the data, develops the reference relations by using the reference information and configures the reference tree. In the score calculation unit, the calculated assessment values (score data) are managed. Also, based on the countermeasure information or other relevant information that has been input, the sample data are generated. In the score calculation unit, based on the reference information stored in the reference tree and the information about the registered countermeasures, the assessment value is calculated and the calculated data are passed to the data management unit.

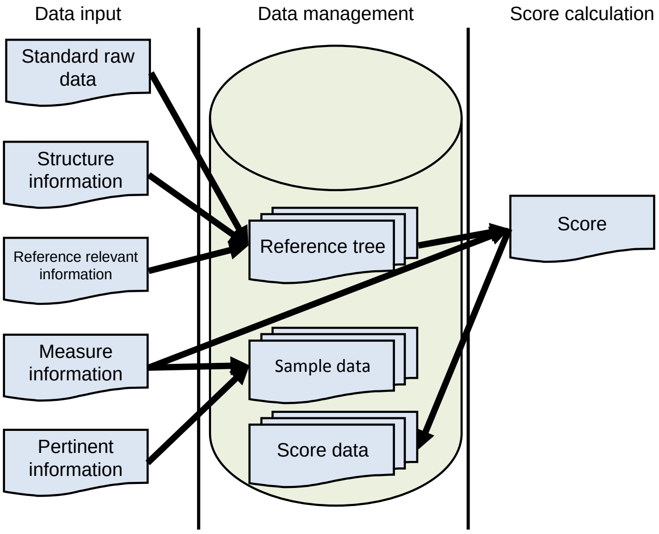


Figure 2: Structure of proposed platform

3.2 Behavior of the platform

First, the standard’s fundamental data from the data input unit are stored. Then, the structural information based on the hierarchical structure described in section 2.3 is stored along with previously registered data. Subsequently, the hierarchy-based information and the direct reference information (hereinafter referred to as direct references) that are described in the standard document are registered. The registered multiple criteria (standards) are related to each other and, if relevant information is provided showing the requirement of measures, in terms of which items of each criterion are related to other items, that information is also registered. After data registration is completed, the registered data are delivered to the data management unit and then migrated to the next operation.

In this platform, the hierarchy is defined using levels. Chapters are defined as level '1' and the following stages as level '2' and so on. Level 'm' is assumed to refer directly to the items of level 'm+1'. In this study, this type of hierarchical structure is also defined as a part of the reference relations.

The basic tree is configured with an item that has a direct reference as the root (hereinafter referred to as the parent reference) and the described items that should be referenced (hereinafter referred to as a reference) are the leaves of the tree. If the leaf of a basic tree becomes the root of another basic tree, a new tree combining the part of the leaf of the former tree with the root of the latter tree is configured. During configuration, a leaf may have the same item as a reference as the root of the tree. If this repeated reference relation has multiple references at multiple locations with the same field as a reference, it will cause a reference loop to occur when the tree is configured. When these references occur, the part that overlaps is designated as the leaf and the configuration of the tree is continued. Thus, binding of the tree is continued until it becomes impossible to bind further and the largest tree becomes the reference tree.

In a reference tree, the relation between the items is expressed as a distance. The distance of those that are referenced directly is 1 and for each iteration of the following references the distance between the items increases gradually.

Subsequently, a standard for security attestation using that reference tree is created in the score calculation unit. The criterion is intended to provide an assessment value for the entire reference tree. In fact, in the data input unit, information about the countermeasure implementation, based on the information of the reference tree, countermeasures in past projects included in the sample data and the compliance status of each item in the standard, is suggested.

Based on the countermeasure information and the reference tree information that has been input, an assessment value is calculated. In addition, if the sample data are set in the data management unit during that time, countermeasures and the supporting data within the compliance status information of each item will be stored as sample data. After that, when the compliance status of the corresponding countermeasure is input by another user, the sample data can be input referring to the sample data that have already been provided.

If relevant information on other criteria is referred to when the assessment is done under new criteria, by means of the data migration function in the data management unit, sample data are generated based on the compliance status data of the underlying criteria and the data can be input while browsing.

3.3 Features of the platform

In this platform, when there is a change in the standard, the information in the data input unit is updated. After updating the information, the reference tree will be automatically reconfigured in the data management unit. In the score calculation unit, reassessment and the recalculation of the score can be done in accordance with the changed contents of the standard.

In addition, the relationships between the items can be visualized by configuring the reference tree. Choosing the countermeasures while checking the reference tree can help to set the effective countermeasures. In the sample data display function, managers who may not have sufficient expertise can share information. In the data migration function, during the reassessment process, the sample data that can be used as reference can be generated without any extra effort.

3.4 System configuration of the platform

This platform has been developed in Visual Basic, and various experiments have been performed so far. First, the entire platform is configured as a single program. The program is composed of independent subprograms: a subprogram that composes information for configuring the reference tree, after registering the criteria, hierarchical structure information and reference relation information; a subprogram that displays the reference tree; a subprogram that organizes the status of the countermeasures; and a subprogram that performs assessment value calculation. These subprograms ensure smooth running of the system by running in the background. For instance, when the data are first registered or when any change is made to the data, changes are made to the reference relations of all the criteria in the background and so, even during the process of making changes, the history of the data can be viewed. In addition, the body of the platform can always be run by operating the time consuming subprograms, such as displaying the reference tree and changing the status of the countermeasures, as independent programs during the process.

In addition, as the assessment value calculation is a separate subprogram, it can be easily changed to a new method. This is useful when introducing or testing multiple methods of assessment value calculation. Similarly, in the display function of the reference tree, instead of replacing the entire program to meet the user's demands, a display program that matches the user's preferences can be easily introduced.

4 METHOD OF CALCULATING IMPACT OF EACH COMPONENT OF THE REFERENCE TREE

In this study, which focuses on the number of items in the reference tree and the distances between them, the value of the assessment can be compared using the security assessment method that changes the impact of each component. In addition when items from other chapters are referred to in the reference tree, it is possible to determine the impact on the calculation results of those items due to the change in the calculation method.

There are four methods tested so far. Method 1 focuses only on the component number. The numbers of existing measures, measures in progress and measures yet to be implemented, in the reference tree which is the root of the estimated item is called '*n*'. The *i*th component is given the value x_i, where x_i is equal to 1 if the estimation item is applicable and is equal to 0 otherwise. The evaluation value $Score_1$ is given by

$$Score_1 = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

Method 2 is an estimation method depending on the maximum distance. For the *i*th component of the reference tree which is the root of the evaluation item, the distance is d_i, and the maximum distance is d_{max}. As for Method 1 the component number is '*n*' and the *i*th component has the value x_i, which is equal to 1 if the estimation item is applicable and equal to 0 if it is not applicable. The degree of impact of the *i*th item is taken as d_{max}–d_i,+1. The evaluation value $Score_2$ is given by

$$Score_2 = \frac{\sum_{i=1}^{n} \{x_i(d_{\max} - d_i + 1)\}}{\sum_{i=1}^{n} (d_{\max} - d_i + 1)} \tag{2}$$

In this method, though there is change in the impact based on the maximum distance in the reference tree, depending on the distance, the impact of the degree of assessment is determined in monotonically decreasing form. Characteristically, the impact of each item on the assessed item falls slowly.

Method 3 uses the reciprocal of the distance. The assessment value $Score_3$ is given as follows:

$$Score_3 = \frac{\sum_{i=1}^{n} \frac{x_i}{d_i}}{\sum_{i=1}^{n} \frac{1}{d_i}} \tag{3}$$

In this method, the impact of each component is not affected by the maximum distance of the reference tree; impact is determined purely by distance. In this method the distance between the items has a great effect for small distances and, as the distance gets larger, the impact slowly falls.

In Method 4, when the evaluation item represented in the hierarchical structure and the chapter of the component are the same, the degree of impact is reduced; when the represented chapter in the reference structure is different, sudden reduction in the degree of impact occurs in accordance with the distance. In addition, when a similar concept is referred to in the reference structure, the calculated degree of impact is relatively high.

5 CALCULATION OF SIMILARITY

5.1 Similarity calculation

In studies of the classification of documents many methods of calculating the similarity have been proposed. In this paper, we adopt the most commonly used technique as our similarity calculation method. The general procedure for calculating the similarity is shown in Fig.3.

First of all, when calculating similarity, the text information in each document is to be determined ((1) in Fig.3). Then, by morphological analysis, this text information is resolved into morphemes and extracted ((2) in Fig.3). These are the index terms (items representing the contents of the document) [11]. One morphological analysis program is “ChaSen” [12] developed by the Nara Institute of Science and Technology. Then, the words that become dissonant are removed as unnecessary words. ((3) in Fig.3). In addition, the extracted words are weighted ((4) in Fig.3). For the weighting method, index word frequency Term Frequency (TF) and Inverse Document Frequency (IDF), or a combination of these, TFIDF, are often used [11]. Finally, the similarity between texts, which are converted to vectors or matrices by weighting, is calculated ((5) in Fig.3).

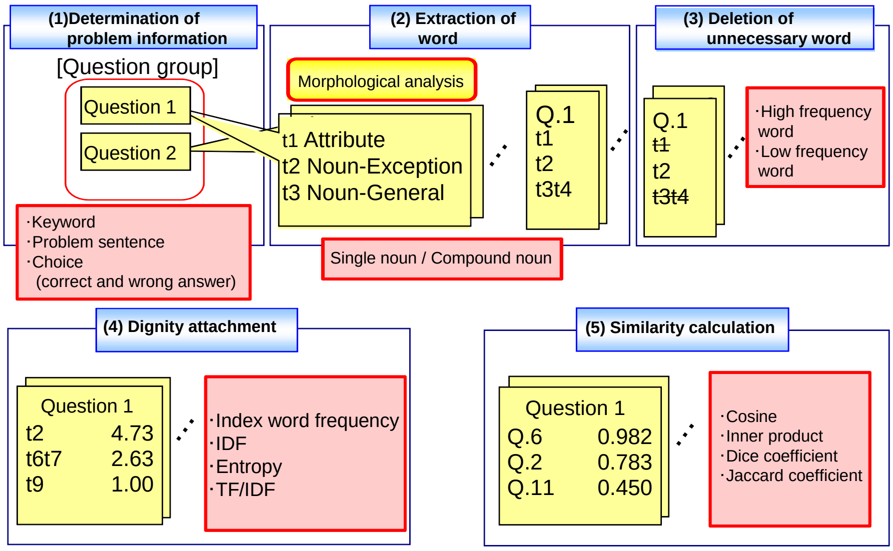


Figure 3: General procedure for calculating similarity

Table 1: Evaluation values for all methods

Category	standard value	evaluation value1	evaluation value 2	evaluation value 3	evaluation value 4
4. Information security management system	20%	11.32%	11.92%	10.45%	13.98%
5. Management responsibility	50%	13.24%	10.79%	13.36%	18.06%
6. Internal ISMS audits	0%	13.24%	10.34%	10.14%	4.91%
7. Management review of the ISMS	0%	0.00%	0.00%	0.00%	0.00%
8. ISMS improvement	0%	13.24%	8.78%	8.59%	1.84%

Table 2: Differences for all proposed types

Category	standard value	difference 1	difference 2	difference 3	difference 4
4. Information security management system	20%	-8.68%	-8.08%	-9.55%	-6.02%
5. Management responsibility	50%	-36.76%	-39.21%	-36.64%	-31.94%
6. Internal ISMS audits	0%	13.24%	10.34%	10.14%	4.91%
7. Management Review of ISMS	0%	0.00%	0.00%	0.00%	0.00%
8. ISMS improvement	0%	13.24%	8.78%	8.59%	1.84%

5.2 Application example

When experiments are carried out using a different standard, the data on the countermeasure's status in the already assessed standard are assumed.

The following application example can be considered. If the standard is updated, it is possible to locate the items of the revised chapter or the items moved to a newly created chapter. Suppose global criteria of an international standard are taken as the base. While creating the local criteria of the internal standard, the platform verifies the extent to which the underlying contents of the standard can be reflected, as well as whether any loophole has occurred. If an internal standard is provided and the aim is to obtain security attestation, the platform can be used to check how close the current internal criteria are close to the target criteria for attestation.

6 EXPERIMENTS BASED ON EACH FUNCTION

6.1 Experiment 1: Evaluation value calculation

We compared the evaluation value using the security evaluation method, which adds a weight factor to each item paying attention to the number of items and distance of a reference tree using Methods 1–4. In addition, we found that there was an impact on the results for items, when items in other chapters were being referred to within a reference tree.

6.1.1. Outline of experiment

First, we asked an evaluator who has expert security knowledge to evaluate the security of an organization, and we summarized the results in a table for every category. Next, we used Methods 1, 2, and 3, evaluated for the same security countermeasures, and compared these evaluation values with the evaluator's assessment. We also investigated whether an improvement in a value could be obtained by using Method 4, based on the knowledge acquired from the experiment.

6.1.2. Experimental results

1) Calculation of evaluation values using Methods 1, 2, 3, and 4

We input the security countermeasures into the platform, and calculated the evaluation value by each method. These values are called evaluation values 1, 2, 3, and 4. The results are shown in

Table 1.

2) Comparison of evaluation values

We compared standard values with evaluation values 1, 2, 3, and 4, and we investigated which method gives a value closest to the standard value in each management field. The differences from the standard value for each evaluation value and each category are shown in Table 2 as differences 1, 2, 3, and 4. Since a lower absolute value of difference indicates a result that is closer to a standard value Method 2, out of methods 1–3, is the most effective in Category 4. This means that countermeasures for the items of the category are in place. On the other hand, Method 3 is the most effective in Categories 5, 6 and 8, which means that reference items instead of the item of the category are being addressed. It was never the case that Method 1, 2 or 3 was the most effective in all the categories. Therefore, we used Method 4 as an impact calculation method in the form where the features of each method were harnessed. This produced an improvement in all categories.

6.2 Experiment 2: Sample presentation

6.2.1. Outline of experiment

We investigated the correspondence of the countermeasures to items of standards by showing that sample data could be generated by administrators who do not have in-depth knowledge of security attestation. This experiment was executed in the form of role play. The sample data were generated by an author who had experience in general security operations and knowledge of security standards. Countermeasure data were generated by a graduate student in our laboratory who has general knowledge about security but does not have in-depth knowledge of security standards.

6.2.2. Experimental results

1) Analysis of the countermeasures by the administrator

First, we asked an administrator to manually distinguish items of standards corresponding to the countermeasures. Since the administrator's knowledge of security standards was not sufficient, he chose items focusing on his notion of a countermeasure. Therefore, the selected results have many effective items for every countermeasure.

Then, the same task was undertaken while viewing reference-related information in a reference tree. Items with low relevance compared with the main items in each management measure were rejected. The same task was undertaken once again while viewing the sample data. The sample data were displayed in two forms, in which the data generated when extracting a countermeasure and the data generated by the administrator were distinguishable. A further reduction in the number of items judged corresponding to the countermeasures was obtained.

2) Interview of the administrator

We interviewed the administrator concerning his changing selection criteria and the results. He was able to determine the relationship among items by using the platform and selected items with confidence after presentation of the sample data. In addition, he said that he left the data that were not in samples with confidence in his judgment in practical jobs.

Table 3: Reproduced rates and assurance of items with a relation

	Number of pertinent items	Number of extraction items	OK	FN	FP	NG	Reproduced rates	Assurance
Top category	10	8	8	2	0	0	80.00%	100.00%
Middle category	31	28	25	5	2	1	80.65%	89.29%
Bottom category	116	97	95	19	0	2	81.90%	97.94%

6.3 Experiment 3: Data conversion

6.3.1. Outline of experiment

First, we compared countermeasures from two viewpoints: "the ISO/IEC 27001 Annex A" and "an ISMS attestation standard Ver.2.0 attachment". Next, we checked the results by carrying out data conversion from each dataset. We asked a graduate student who is an administrator of our laboratory to participate in an experiment using the countermeasures adopted in our laboratory.

6.3.2. Experimental results

We used about 20 countermeasures. The number of different items with a correspondence was a little more than 120. We could obtain all patterns, including opposite selection and one side selection. By analyzing the contents of items that showed a difference, we could classify the differences into the following six patterns.

i. The contents of the item were specified in detail.
ii. The contents of the item became ambiguous.
iii. If the contents of an item at a higher level to an item differ; those items to which it points also differ.
iv. The contents are expressed differently; the meaning does not change.
v. The same contents are viewed from another aspect.
vi. An item does not belong to the same category in both standards.

6.4 Experiment 4: Pertinent information extraction by similarity

6.4.1. Outline of experiment

We calculated the similarity between two standards, the ISO/IEC 27001 Annex A (hereinafter Standard A) of the international standard and an ISMS attestation standard Ver.2.0 attachment "detailed management measure" (hereinafter Standard B) which is part of a Japanese standard. The pertinent information in the two standards is already specified. We defined items that have the maximum calculated similarity between the two standards as "items with a relation" and we checked how many specified relations were reproduced. We classified items that were not reproduced into three categories: False Negative (FN), which means they were not extracted although there is a relation; False Positive (FP), which means they were extracted although there was no relation; and NG, which indicates that the wrong item was extracted.

6.4.2. Experimental results

A comparison of the pertinent information in Standards A and B the items extracted as items with a relation is shown in Table 3. The reproduced rates exceed 80% in the top, middle and bottom categories. Each assurance has a value exceeding 89%.

We investigated the 31 errors (26 FN, 2 FP and 3 NG) to determine the cause. We found that most of the combinations that cause errors have low similarity. In the top category, which contained few technical terms, if a more suitable judgment could be made, we could transfer one of FNs to the correct combination. Also, if similar words could be correctly distinguished between items, we could also transfer the other FN to the correct combination. Each of three combinations detected as FP and NG in the middle category had similarities less than 0.5. Moreover, the NG item was extracted using only an item name, so the similarity was 1, and full match was carried out. However similarity was decreased by combining the name with a portion of the detailed description. For the FNs there were also cases which showed coincidences or high similarity of item names. Other causes of FNs were a low maximum similarity viewing from both standards A and B, or a maximum similarity viewing from one side whereas the similarity is the second or third value from the other side and could not be detected because of its small margin. In the bottom category, FPs did not appear. The two combinations in the NG category showed the maximum similarity seen from one side, and had second or third similarity values seen from the other side. We could classify most of the 19 FNs into the same two cases as for the middle category

The following knowledge was acquired from these analytical results.

i. An item which has a maximum similarity less than 0.5 does not have a related item in many cases.
ii. When a description is divided into an item name and detailed description, the similarity of the item name becomes more important.
iii. Related items can be detected in many cases if they include an item with higher similarity, even when the

maximum similarity from both sides indicates that there is no related item.

6.5 Discussion

6.5.1. Experiment 1

Through these experiments based on the thinking of an evaluator, we found an influence on achievement level in the management category from items at a large distance in the reference tree of the platform. These items make reference to items outside of the management category. In addition, we found that using reference trees is an effective way to avoid human errors, such as overlooking the influence of items referring to other categories.

Moreover, the evaluation value has been improved in Category 5. "Management responsibility" by changing a method to reflect comments from an interview. However, the difference between the participant's evaluation value and the standard result is still large. This may be because possibilities are added as evaluation criteria, or because contents other than actual evaluation criteria may be reflected in a result.

6.5.2. Experiment 2

There was a tendency for a participant with insufficient professional knowledge to select more items for countermeasures. Through an interview we found out that relationships between standards were difficult to discern for the administrator who had insufficient knowledge. We also found out that it is effective to express relationships visually using reference trees and that the presentation of sample data was useful.

6.5.3. Experiment 3

From items i, ii, and iii in Section 6.3.2, since changes may come out in countermeasures by expressional range, we recognize that it is not appropriate to simply change data. From items iv, v, and vi, we see that errors can be avoided by showing the sample data.

6.5.4. Experiment 4

We confirmed that high reproducibility can be obtained by extracting items that have relations based on text similarity. We found in particular that the assurance of the items extracted was very high. Some of the causes of errors were due to improper range division of words at the time of the analysis of wording. In addition, different words with the same meaning cannot be automatically judged because technical terms are used and the similarity is low. In spite of using a simple similarity calculation, a high reproduction rate and high assurance were obtained. So it appears that it is effective to use the technique of extracting related items to determine the similarity between standards by similarity calculation methods currently used in the field of natural language processing. Moreover, it is expected that still higher reproduction rates and assurance can be obtained by creating pertinent information using a more sophisticated technique.

Once the data for sample presentation are generated, it seems to be important to reduce FP and NG items even if FN increases. This is because we assume users who do not have much specialized knowledge. For example, the following techniques may improve the results. When the standard document is divided into item names and detailed descriptions, importance should be placed on the item name instead of employing the weighting used in our experiments. Since a standard has a hierarchical structure, the similarity and detection of relations of items in higher categories should also be taken into consideration.

6.5.5. All experiments

From experiments 1, 2, and 3 we found out that platform is effective in preventing human error. The errors that can be prevented are different in each experiment, but what is considered as the primary cause of errors depends on the complicated composition of the standards used as the base document which is one of the targets of this research. In this approach, visual correspondence was provided by using reference trees, and contributed to problem solving.

In particular, visual support was provided by reference trees in experiments 1 and 2, and this contributed to the prevention of errors. In experiments 2 and 3, visual support was provided by the presentation of the sample data, which also contributed to the prevention of human error.

Moreover, in experiment 4, by using the technique of text similarity calculation, pertinent information was extracted from the standards, even where relations between the standards are not indicated. We confirmed that pertinent information can be generated from various standards, such as a global standard and a local standard.

These experiments are highly flexible and their application is not limited to security-specific standards. However, since experiment 2 is designed for choosing the relation between security countermeasures and a standard, the security viewpoint is strongly reflected here.

7 FUTURE WORK

The sample presentation function has basic issues, such as determining a sample collecting rule and reliability. Currently, we are considering solutions based on practical use rather than technical considerations. Regarding the sample collection rule we have proposed a rule in which the data generated from the sample data are provided as a new sample. Regarding reliability, we have proposed the following method, which uses a central server, in order to improve the reliability of the sample data. If entries corresponding to the same item about the same measures are stored more than a fixed number of times, the server will automatically judge that the sample data are reliable, and adopt them as the sample data. Otherwise, the data are checked by a human and adopted if their validity can be confirmed.

We have experimented with using the phases of gap analysis and present data analysis. However, there are many phases in which security evaluation can be carried out. Some of examples are the phase in which the detailed risk analysis is conducted, and the phase in which attestation acquisition has already finished and the PDCA cycle corresponding to the phase that carries out security evaluation has already been employed. Therefore, we will also conduct a security evaluation experiment of an organization with other phases, and examine the validity of the platform.

We conducted an experiment using a calculation of similarity, and we used a standard that has pertinent information. Nevertheless, errors occurred. We will try to avoid these errors by using semantic similarity and raising text analysis accuracy. For example, we could use the structural information (e.g., about hierarchical structure) and reference information of a standard. We are planning experiments in which we will calculate the similarity by assuming that the item name is more important, if the standard item consists of a name and detailed description.

8 CONCLUSION

In this paper, we verified, based on the experimental results, not the validity of an individual function but the validity of the whole platform. We found that the platform is effective for such problems as oversight and insufficient knowledge by using visual support that presents reference trees or samples.

In this platform, we confirmed that potential influence is expressed using reference-related information in cases where influence may be overlooked even if the evaluator has expert knowledge. In addition, we recognized that the provision of visual information by reference trees and sample presentation was effective for oversight and avoidance of misjudgment, when knowledge was insufficient.

Furthermore, we found that each function can be utilized more effectively by interlocking two or more functions, such as sample presentation and data conversion. In this study, we expressed the status of countermeasures by the two choices "done" and "not yet" for simplicity. In addition, we expect that evaluation of potentiality can be improved ascertaining the optimal rate of "not yet" if potentiality and the state of being under way are expressed by using a third choice of "doing".

We conducted experiments using two standards where pertinent information was clearly specified. We recreated the pertinent information with a high reproduction rate and high assurance. By determining such pertinent information through a similarity calculation technique, we were able to lessen the rollback of the reappraisal carried out when the standard changes. It is expected that better results can be obtained by using a more sophisticated technique.

We will continue to examine the adaptability of our platform to various phases and we will try to improve its validity.

REFERENCES

[1] JIPDEC, “The international trend of ISMS, and the actual condition of a measure <2004 edition>,” (2005).

[2] Information Management Systems Promotion Center (IMSPC), “The number transition of attestation acquisition organizations,” The attestation acquisition organization of a certificate authority exception and a prefecture level, http://www.isms.jipdec.jp/lst/ind/suii.html.

[3] IPA, “Security design evaluation supportive tool V03,” http://www.ipa.go.jp/security/fy13/evalu/cc_system/CCtool_V03/secevtoolv03.htm.

[4] Y. Takahashi, and Y. Teshigawara, “A Study on a Security Evaluation Platform Based on International Standards,” IPSJ Computer Security Symposium 2008 The 2nd separate volume of collected papers, pp. 815–819 (2008).

[5] Y. Takahashi, and Y. Teshigawara, “A Study on an Effectiveness of Security Evaluation Platform Based on International Standards,” IPSJ SIG Technical Report, Vol. 2009-CSEC-46, No.13, pp.1-8 (2009).

[6] Y. Takahashi, and Y. Teshigawara, “A Study of Security Evaluation Method Based on Reference Relationships among International Standards,” IPSJ SIG Technical Report, Vol. 2010-DPS-142, No. 53, pp.1-8 (2010).

[7] Y. Takahashi, and Y. Teshigawara, “A Study on Measures Presentation Function for Non-Professional Persons of Security Evaluation Method Based on Reference Relationships among International Standards,” Multimedia, Distributed, Cooperative, and Mobile Symposium.(DICOMO2011), pp. 127–134 (2011).

[8] Y. Takahashi, and Y. Teshigawara, “A Study on Data Conversion Function of Security Evaluation Method Based on Reference Relationships among International Standards,” IPSJ Computer Security Symposium 2011(CSS2011), pp. 666–671 (2011).

[9] Information Management Systems Promotion Center (IMSPC), “International trend "ISO/IEC 27000 family,” http://www.isms.jipdec.or.jp/27000family_20111220.pdf .

[10] ISO/IEC 27001, “Information technology - Security techniques - Information security management system – Requirements,” (2005).

[11] T. Tokunaga, “Information retrieval and language processing,” University of Tokyo Press (1999).

[12] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, and Masayuki Asahara, “Japanese Morphological Analysis System ChaSen 2.0 Users Manual,” NAIST Technical Report, NAIST-IS-TR99012, Nara Institute of Science and Technology (1999).

(Received October 20, 2012)
(Revised January 7, 2013)

Yuji Takahashi received the B.E and M.E from Faculty of Engineering, Soka University in 2001 and 2003. He is currently doing his Ph.D project at Soka University. His research interests are security management and international standard of security. He is a member of Information Processing Society of Japan (IPSJ).

Dr. Yoshimi Teshigawara is a Professor of Department of Information Systems Science, Faculty of Engineering at Soka University since 1995, He began his professional career in 1970 at NEC Corporation, engaged in the design and development of computer networks. From 1974 to 1976, Dr. Teshigawara was a Visiting Research Affiliate with ALOHA System at the University of Hawaii where he did research on packet radio and satellite networks. He served Dean of Faculty of Engineering and Dean of Graduate School of Engineering at Soka University. His current interests are network security, e-learning, and ubiquitous sensor networks. Dr. Teshigawara received his PhD from Tokyo Institute of Technology, Japan, in 1970. He is a fellow of Information Processing Society of Japan as well as Japan Operation Research Society. He is a member of IEEE and ACM.

Evaluation of Site-Independent Creativity Consistent Support System for Actual Work Environment

Tomohiro Kokogawa[†], Toshihiro Ajiki[†], Junko Itou[‡], and Jun Munemori[‡]

[†]Graduate School of Systems Engineering, Wakayama University, Japan
[‡]Faculty of Systems Engineering, Wakayama University, Japan
koko@fw.ipsj.or.jp, munemori@sys.wakayama-u.ac.jp

Abstract - In recent times, it has become increasingly important not only to maximize the knowledge and expertise gained at work, but also to create new knowledge from those. Creativity methods such as the KJ method[1] are suitable for addressing these challenges and effectively developing creative ideas. For effective application of knowledge at a work site, without constraints on time and place, we propose a site-independent creativity consistent support system based on Quiccamera and GUNGEN-SPIRAL II. Further, we present the results of experiments to evaluate the system and demonstrate its effectiveness.

Keywords: KJ method, photograph, creativity support system, tablet device

1 INTRODUCTION

In recent times, the growth of information and communication technologies has highlighted the realities of globalization in economics and business competition. Companies must foster continuous innovation through short-cycle product development, business efficiency improvement, cost reduction, and rapid decision making in order to accommodate the diversity of markets and technologies. Therefore, it becomes increasingly important not only to maximize the knowledge, experience, and expertise accumulated in the organization but also to create new knowledge from those.

Further, in the event of natural disasters such as floods, earthquakes, and tsunamis, people may provide additional information through social media. Various studies have analyzed this behavior and have attempted to utilize it for disaster management measures [1], [2]. Such information includes knowledge, experience, and expertise from victims or experts, and it will be useful in designing crisis management measures for similar disasters in the future. Major disasters also highlight the importance of risk assessment for events that are rare but have great impact [3]. Such information is difficult to extract without a range of divergent opinions.

Various creativity methods [4]-[8] and their support systems [9]-[11] have been proposed. However, the application of idea generation methods requires a certain level of practice, and a specific amount of time is necessary in order to achieve results of good quality. Many organizations encounter the dilemma of satisfying the requirements for gathered knowledge and dedicating enough time to gather the required knowledge. Creativity methods for developing creative ideas effectively are suitable for addressing these challenges.

Most creativity methods and their support systems are designed to be applied in meetings that require the participants to be present within a single room. In order to gather the ideas of those concerned, particularly in the actual work environment, the creativity methods should be applicable irrespective of time and place.

In this study, we propose a consistent support system for creativity in the actual work environment. This can support the entire process of a creativity method on-site, without imposing any restriction on the actual work site. Further, we describe the implementation of this system using tablet devices with cameras. Our experiments demonstrate the effectiveness of the proposed system.

2 RELATED WORK

The KJ method developed by Jiro Kawakita [5] is a creativity method based on the theory of problem solving and teamwork, and is also referred to as the affinity diagram and is included in the Seven Management and Planning Tools [8] used in total quality control. The process typically used in the KJ method, which is based on the human thinking process for creative problem solving [9], is as follows:

(P-0) Data gathering
Data (ideas, opinions, issues, etc.) with a specific theme are gathered.

(P-1) Label creation (divergent thinking)
While selecting data and brainstorming, each idea is recorded as a label

(P-2) Category creation (convergent thinking)
The labels are organized into groups based on the natural relationships between labels, and each group is given a title.

(P-3) Chart creation (idea crystallization)
Each group is spatially allocated to a chart (affinity diagram) according to the natural relationships between groups. Steps (P-2) and (P-3) are processes typical of the KJ method, and we refer to these as the narrowly defined KJ method in this paper.

(P-4) Conclusion (idea verification)
Concluding sentences are added to express the meaning of the diagram.

[1] The KJ method is a registered trademark of Kawakita Research Institute.

PAN/KJ [10] is a KJ method support system that can utilize multimedia data such as images or audio data. This system uses multimedia data as hyperlinks for card labels, but does not use multimedia data directly in the form of labels.

GUNGEN-SPIRAL II [11] enables the consistent process of the KJ method to be implemented as a Web application, thus facilitating idea generation using multiple devices such as PCs or smart phones with modern Web browsers.

Geographical Location Information-Based Bulletin Board System (GLI-BBS) [12] is a groupware system that can share geographical location information among communities. This system enables the data uploading of photographs from a GPS-equipped cellular phone to a BBS, which shows the photographs with the related geographical information.

Evernote[2] is a memorandum sharing system based on cloud computing. This system enables submissions of text memos, freestyle drawings, and photographs easily by using PCs or smartphones, and it facilitates sharing among devices. Some case studies have demonstrated the effectiveness of the functionality of Evernote in the collaboration process [13].

Digital Card Cabinet [14] was featured in a special exhibition of the work of Tadao Umesao at the National Museum of Ethnology in Osaka, Japan. Tadao Umesao was known for his special B6-size paper cards (known in Japan as Kyoto University cards) that improve intellectual productivity [15]. The Digital Card Cabinet system allowed visitors of the exhibition to create personal Umesao-style cards, store them in digital form in a digital cabinet, and share them with other visitors by using a tabletop touch screen panel. Each digital card was created with text, photographs, and freestyle drawings by using an iPhone[3] native application, or scanning a paper card.

3 REQUIREMENTS

There are two kinds of work sites in the actual work environment; the head office site and the actual work site. The personnel at the head office site gather data from the actual work site, analyze those, and make decisions to solve problems. At the head office site, there are also executives, staff, and external experts who eventually judge the decisions. The personnel at the actual work site gather data there and send this to the head office site for a decision. At the actual work site, there are actual workers who have various amounts of knowledge and experience of the work environment. In a disaster situation, there are also victims and volunteer staff to rescue the victims.

Most previous creativity support systems merely supplied the process at the head office site with sufficient facilities such as PCs, networks, and hot-wired meeting rooms. In order to obtain an effectual decision with a creativity method, it is important for the people at the head office site to collaborate with those at the actual work site by combining their different set of implicit knowledge for each thought process. However, it is difficult to gather both groups of people simultaneously because of the restrictions of time and place, especially the restrictions on infrastructure at the actual work site.

In this study, we propose a consistent support system for creativity on-site, which enables users to perform an entire creativity process under restrictions of time and place but independent of the actual work site. A conceptual model of our proposal is shown in Fig.1. The solid arrows in the figure show the flow of the human thought process for a creativity method such as the KJ method. The dashed arrows in the figure show the flow of data for collaboration between the head office site and the actual work site. The purpose of our proposed system is consistent support for the creativity process at the actual work site, which is under many restrictions and in collaboration with the head office site consistently.

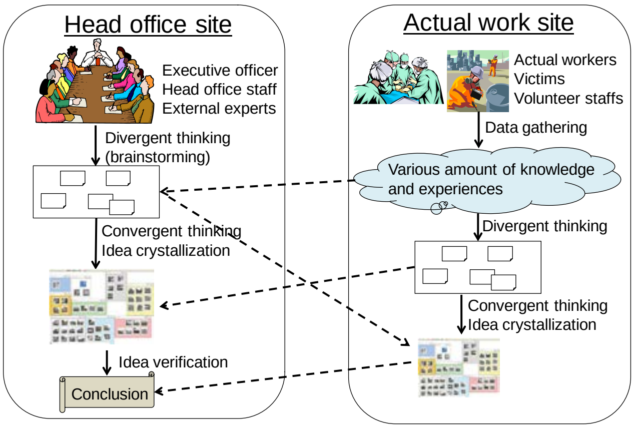


Figure 1: Conceptual model of site-independent creativity consistent support system.

The requirements of the system are as follows:

(1) System users

In order to collect a diversity of useful opinions, a variety of people should join the system and implement creativity methods as a team. In addition to head office staff, the group must consist of external experts or staff from the actual work site (e.g. industrial plants, construction fields, or disaster site).

(2) Time and place

The system should be accessible from any place and at any time, thus enabling the participants to implement the creativity method whenever necessary.

(3) Target process

As described above, a variant of the KJ method has a sequence of processes (from divergent thinking to convergent thinking or conclusion) to obtain results using creativity methods. Each process may be executed at a different place or time.

(4) User abilities

People with a variety of skills may use the system and creativity methods; hence, a simple user interface is required. In addition, most creativity methods need a large workplace in order to gather a large number of ideas and obtain an overview.

[2] http://www.evernote.com/

[3] http://www.apple.com/iphone/

4 APPROACH

4.1 Basic Design

We developed Quiccamera as a support system that enables rapid data gathering and label making in the actual work environment [16]. By means of Quiccamera, comments using text, pictographs (emoji), or handwriting can be added to photographs captured at the location. Then, the photographs can be sent directly to GUNGEN-SPIRAL II [11] as idea labels for implementing the KJ method.

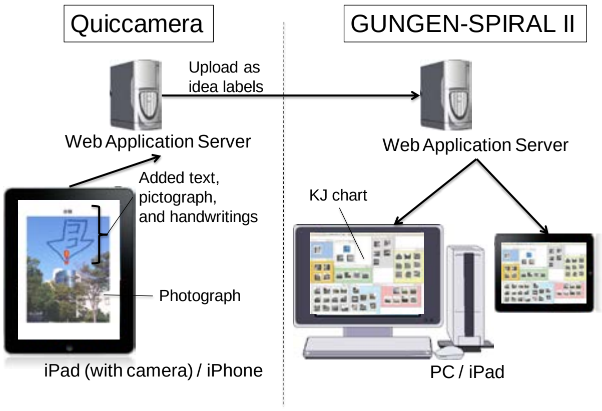


Figure 2: Overview of creativity consistent support system based on Quiccamera and GUNGEN-SPIRAL II.

An overview of our complete implementation of a creativity consistent support system based on Quiccamera and GUNGEN-SPIRAL II is shown in Fig.2.

4.2 Function Design

The main functions of Quiccamera are as follows:

(1) Use of client terminal with touch panel and camera

The system should support a short handwritten note with a photograph. Hence, this client function is enabled for a smartphone or tablet device with a touch panel and camera, such as the iPhone or iPad[4] (except the first model).

(2) Single button for image submission

With a smartphone or tablet device, an image can be uploaded to a server by several methods such as e-mail or ftp. However, these methods are not simple because multiple applications or operations are required. Hence, the entire operation (capturing, editing, and uploading the photograph) is implemented through an application that requires minimum operating effort.

(3) Freestyle comments on the photograph

Handwriting is one of the simplest methods for adding comments to a photograph. However, sometimes handwritten letters are difficult to read or manage. Therefore, a more useful method is the text input function that is provided. In addition, a variety of pictographs can be used to convey feelings related to the target object in the photograph [17].

(4) Direct use of photographs with comments as idea labels in GUNGEN-SPIRAL II

Photographs uploaded to the Quiccamera server are converted and uploaded directly to the GUNGEN-SPIRAL II server as XML-style idea labels. Thus, we can implement the entire KJ method as one consistent process with our proposed system.

4.3 System Implementation

We developed Quiccamera as a native iPhone or iPad application written in Objective-C and a server application written in PHP for receiving photographs.

GUNGEN-SPIRAL II is a Web-based server application written in PHP and JavaScript. It enables the KJ method on all terminals having modern Web browsers, such as PCs, smart phones, and tablet devices.

We chose an iPad2 (second generation iPad) as the client terminal for Quiccamera and GUNGEN-SPIRAL II, because the iPad2 has a camera and a wide display (similar to those of a notebook PC).

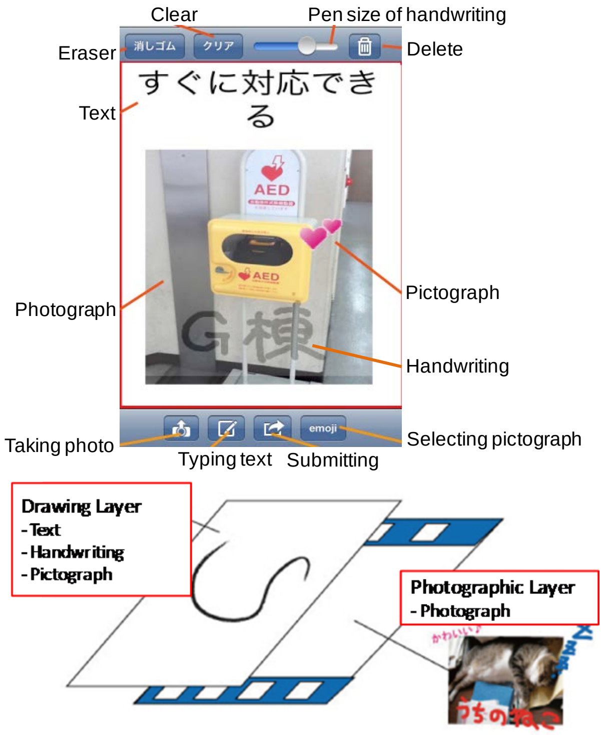


Figure 3: Overview of main screen of Quiccamera application.

First, the user launches the Quiccamera application and captures a photograph using the built-in camera of the iPad 2. The captured photograph is displayed on the main screen of the Quiccamera application, and the user can add several freestyle comments on the photograph by using text, various pictographs, or handwriting with multiple pen sizes. Four

[4] http://www.apple.com/ipad/

pictographs were implemented for comments of laughing, crying, surprise, and love. Figure 3 shows the main screen of the Quiccamera application. The comments are stored in the drawing layer, which is separate from the photograph layer, and the user can edit these freely. In our experiments we prepared the four pictographs shown in Fig.4.

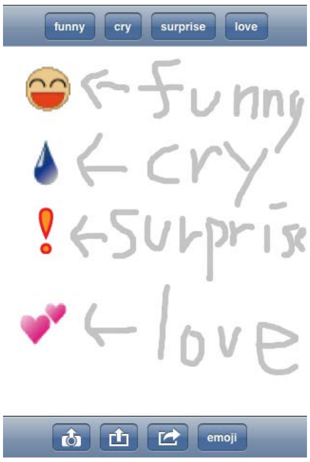


Figure 4: Variety of pictographs.

Once the editing is complete, the user presses the upload button to save the data in the PNG format by combining the layers and submit the result to the Quiccamera server. The data is automatically uploaded to the GUNGEN-SPIRAL II server in the form of idea labels. Then, the user launches the GUNGEN-SPIRAL II application to initiate the categorizing and charting process of the KJ method by using the uploaded idea labels (photographs with several freestyle comments).

5 EXPERIMENTS AND DISCUSSION

5.1 Experimental Environment

The experiments were conducted with six groups of participants: each group consisted of three students from Wakayama University. The theme of the KJ method was "ultimate methods for adopting measures to deal with the occurrence of a disaster or tsunami." As the actual work site, we selected Arita County (the towns of Yuasa and Hirokawa) in Wakayama Prefecture, which is known for an old tsunami story [18] [5]. Wakayama University was simulated as the head office site. The participants executed all the steps of the KJ method (from data gathering (P-0) to conclusion (P-4)), and four other individuals evaluated the quality of the result sentences by Yagishita's method [19], which expands the application scope of the traditional Analytic Hierarchy Process (AHP) [20]. We adopted six evaluation factors (originality, usability, appeal, concreteness, possibility of realization, and possibility of application) to calculate the satisfaction scores of result sentences.

Our experiments were conducted and evaluated on the basis of three aspects:

(1) Evaluating the effect of photographic idea labels (Groups A and B)

The participants each took an iPad 2 or iPhone to the actual work site (Arita County), gathered data by taking photographs (data gathering), and created idea labels by drawing several memorandums with Quiccamera. After several days, they performed the remaining processes of the KJ method (category creation, chart creation, and conclusion) with GUNGEN-SPIRAL II at the head office (Wakayama University) using a PC. For the purposes of comparison, they also created textual idea labels (without using Quiccamera) just before creating the categories. We evaluated the number of idea labels generated at (P-1) and the number of categories generated at (P-2). We also evaluated the processing time of the narrowly defined KJ method from (P-2) to (P-3) and the satisfaction score of the concluding sentence at (P-4). The time gap between label creation and category creation as well as the difference between the devices emulates an inconsistent creativity process.

(2) Evaluating the support for a consistent process (three Groups C, D, and E)

The participants each took an iPad2 to a location near the head office site (Wakayama University) as an actual work site with fewer restrictions on time, place, and equipment. They gathered data and created idea labels in a manner similar to that described in (1). Then, they returned to the head office and performed the remaining processes of the KJ method with the iPad2 consistently. We evaluated the processing time of the narrowly defined KJ method and the satisfaction scores of the concluding sentences. The number of idea labels was fixed at 21 (each participant created 7 labels) in each experiment to evaluate all idea labels under the same experimental condition, when considering the legibility on the screen size of the iPad2.

(3) Pre-evaluating the support for a consistent process in the real actual work site (one group: F)

The participants each took an iPad2 to the actual work site (Arita County), gathered data, and created idea labels in a manner similar to that described in (1). Then, they assembled at a suitable location within the site and performed the remaining processes of the KJ method with the iPad2 consistently. The evaluation criteria and the experimental condition (the number of idea labels) were the same as in (2).

Table 1 shows the environments for all the experiments. W and O represent the locations at which the participants executed the processes of the KJ method. W indicates the actual work site (Arita County) and O indicates the head office site (Wakayama University). In each experiment, the category creation, chart creation, and conclusion processes were simultaneously performed at the same place.

[5] The story was translated as "Inamura-no-Hi" by Tsunezo Nakai and used as a government-designated teaching material in Japanese elementary schools before World War II.

Table 1: Experimental environments.

Group		Data Gathering	Label Creation	Category Creation
A	A1	W	W (iPad)	O (PC)
	A2	W	O (Text)	O (PC)
B	B1	W	W (iPad)	O (PC)
	B2	W	O (Text)	O (PC)
C		O	O (iPad)	O (iPad)
D		O	O (iPad)	O (iPad)
E		O	O (iPad)	O (iPad)
F		W	W (iPad)	W (iPad)

Figure 5 shows an example of providing idea labels as input by using Quiccamera at the actual work site (Yuasa Town). Figure 6 shows an example of using Quiccamera to submit an idea label. The text in Fig.6 shows a short question about the addition of the dike in the photograph. The pictograph (exclamation mark to indicate "surprise") emphasizes the need to focus attention. The freestyle drawing (arrow) shows the size of the added dike.

Figure 5: Experiments at Yuasa.

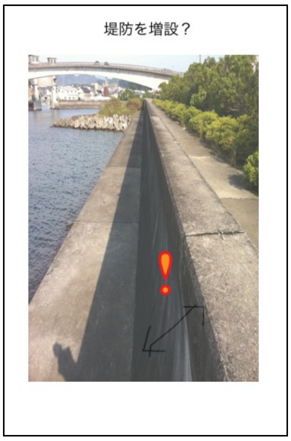


Figure 6: Example of photographic idea label with text, pictograph, and handwritings.

Figure 7 shows an example of the KJ method being performed using a PC at the head office (Wakayama University). KJ charts in GUNGEN-SPIRAL II were displayed on the wall by using a projector in order to share the entire workspace with all the participants. The submitted photographs were shown as idea labels in GUNGEN-SPIRAL II. The participants applied the KJ method by displaying the entire screen of GUNGEN-SPIRAL II using a projector.

Figure 8 shows an example of the KJ method being applied with an iPad 2 at the actual work site (a tearoom in Yuasa). The iPad 2 was connected to the GUNGEN-SPIRAL II server at the head office by using a 3G-WiFi router.

Figure 7: Performing the KJ method using a PC at the office.

Figure 8: Performing the KJ method using an iPad 2 at a tearoom in Yuasa.

5.2 Results

Figures 9 and 10 show the resultant KJ charts, considering the experiments of Group A as examples. Figure 9 shows the resultant KJ chart after the chart creation process using idea labels from Quiccamera. Each photographic idea label contains a photograph with text, pictographs, or handwriting. Figure 10 shows the textual (traditional) KJ chart.

Table 2 lists the results of Experiments A1-B2. It shows the number of generated idea labels at (P-1), the number of generated categories at (P-2), the process time of narrowly defined KJ method from (P-2) to (P-3), and the average of satisfaction score of the result sentence at (P-4) calculated with Yagishita's method [19]. The time is given in minutes. Table 3 shows the results for label making by using text, handwriting, and pictographs at Experiments A1 and B1.

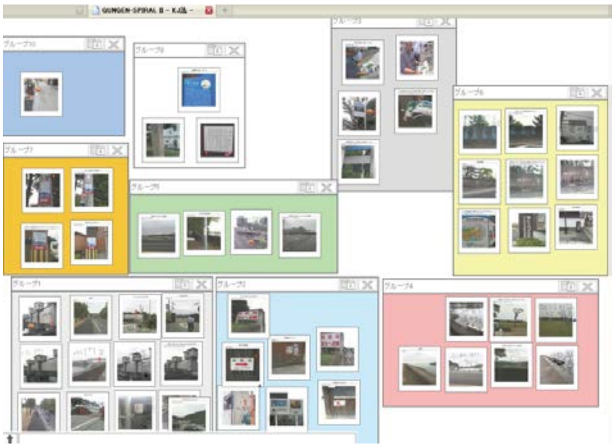


Figure 9: Example of KJ chart using photographic idea labels from Quiccamera (A1).

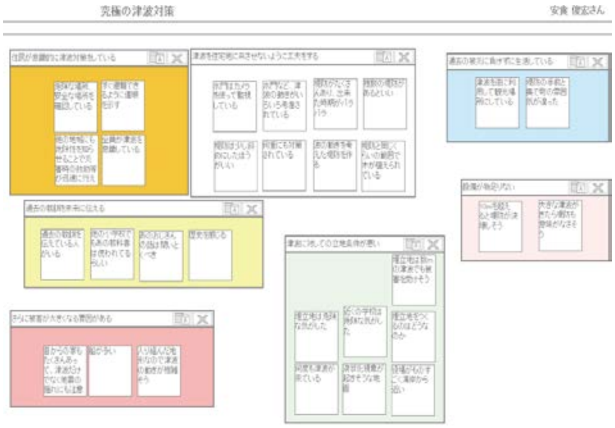


Figure 10: Example of KJ chart using textual idea labels (A2).

Table 2: Experimental results (A1-B2).

Group	Num. Labels	Num. Categories	Time (min.)	Avg. Score [19]
A1	53	9	40	4.0
A2	30	7	20	2.7
B1	31	6	52	4.3
B2	35	10	45	2.0

Table 3: Breakdown of making idea labels with Quiccamera.

Contents		A1	B1
Labels with text		24	28
Labels with handwriting		28	10
Labels with pictographs		11	11
Variety of pictograph	Funny	4	0
	Cry	1	0
	Surprise	9	11
	Love	1	0

Table 4 lists the results of Experiments C-F, which also shows the process time of narrowly defined KJ method, and the average of satisfaction score of the result sentence.

Table 5 lists the questionnaire regarding the usability of Quiccamera, and Table 6 lists the questionnaire regarding the usability of GUNGEN-SPIRAL II. Each score shows the average provided by each participant from 1 to 5 score.

Table 4: Experimental results (C-F).

Group	Time (min.)	Avg. Score [19]
C	50	3.9
D	25	2.8
E	56	4.9
F	41	2.8

Table 5: Questionnaire results for Quiccamera.

	Questionnaire	O (C-E)	W (F)
1	Did you feel it a burden to take and to submit a photograph with Quiccamera? *(1:Not a burden-5: A great burden)*	1.6	1.3
2	Was the addition of a memorandum by electronic handwriting simple? *(1:Did not feel so-5: Strongly felt so)*	3.9	4.0
3	Was the addition of memo written by text simple? *(1:Did not feel so-5: Strongly felt so)*	3.7	4.3
4	Was it effective to add the contents of electronic handwriting to your own memorandum? *(1:Did not feel so-5: Strongly felt so)*	4.0	4.7
5	Was it easy to upload photographs? *(1:Not easy -5:Easy)*	4.6	4.7
6	Did the pictographs help in adding information? *(1:Did not feel so-5: Strongly felt so)*	2.9	2.7
7	Was the variety of pictographs suitable? *(1:Too few-5: Too many)*	2.6	3.0
8	Do you think that electronically handwritten memorandum is effective even without photographs? *(1:Did not feel so-5: Strongly felt so)*	2.0	2.7

Table 6: Questionnaire results for GUNGEN-SPIRAL-II.

	Questionnaire	W PC (A1,B1)	O iPad (C-E)	W iPad (F)
1	Were the idea labels completely readable? *(1:Did not feel so-5:Strongly felt so)*	3.3	3.0	4.0
2	Was it more convenient to perform the KJ method with the proposed system than with the paper-based method? *(1:Did not feel so-5:Strongly felt so)*	4.0	3.6	4.7

5.3 Discussion

In the experiments above, we observed the following results:

(1) Evaluating the effect of photographic idea labels

In Table 2, the average score for concluding sentences is higher with photographic idea labels than with textual labels, although there are no significant differences between the numbers of labels or categories generated.

Table 7 shows portions of the concluding sentences of Group A as examples. The underlined portions indicate the significant differences between the photographic idea labels and the textual idea labels. With the photographic idea labels, the concluding sentences include more specific expressions such as examples of actual scenarios. This would help in the generation of more practical output for actual work environments.

Table 7: Examples of concluding sentences.

Concluding sentences with photographic idea labels (A1)
The ultimate action to prepare measures for dealing with a tsunami is to exploit the review of past experiences. For example, a conscious measure was taken by locating a sign for people living in the area to escape to safety upland away from a tsunami, or by locating a sign about how dangerous the place is during a tsunami.
Concluding sentences with textual idea labels (A2)
The ultimate action to prepare measures for dealing with a tsunami is to hand on the past lessons to the future generations. That makes it possible to take conscious measures by indicating the way for inhabitants to escape immediately, or by checking the safety of places.

Table 3 indicates that textual comments were predominantly used. The number of comments using pictographs was less than the number using text or handwriting, respectively. In the pictograph comments, the pictograph for surprise was typically used.

(2) Evaluating the support for a consistent process

In Tables 2 and 4, we compare the average scores of concluding sentences between inconsistent processes (A1 and B1) and consistent processes (C, D, E, and F). The results show that there are no significant differences between these.

Table 5 shows that the score for uploading a photograph is high (4.7) and that the effort to submit photographic idea labels is low (1.6). This result indicates that the main function of Quiccamera is achieved.

Although Table 6 shows that the readability and operability are slightly higher when a PC is used than when an iPad 2 is used, there are no significant differences between these.

(3) Pre-evaluating the support for a consistent process at the actual work site

The experimental results of Group F show that there are no significant differences in the scores of concluding sentences between Groups C–E and Group F, although the average score of concluding sentences is lower for Group F. This indicates that there were no dependences on where experiments were performed. Hence, the system was able to support the whole KJ method consistently at both the head office and the actual work site.

6 CONCLUSION

In this study, we proposed a site-independent creativity consistent support system, which can apply creativity methods without constraints on time and place. We also demonstrated the effectiveness of our proposal by conducting experiments.

The proposed system consists of Quiccamera and GUNGEN-SPIRAL II. Quiccamera supports divergent thought processes and GUNGEN-SPIRAL II supports convergent thinking for conclusions. This system can consistently support the entire process of a creativity method at any time and at any place by using a tablet device (iPad 2) as a client terminal.

Experimental results showed that the use of photographic idea labels is easier with Quiccamera. Further, the quality of the resultant conclusion is better when photographic idea labels are used than when textual idea labels are used, owing to the more concrete idea generation. In addition, there were no significant differences in the quality of concluding sentences between using a PC and using an iPad 2 or between working at the head office and working at the actual site.

In future studies, we will increase the variety of freestyle comments by including multicolored handwriting and additional pictographs to support the creativity method more effectively.

REFERENCES

[1] D. Yates, and S. Paquette, “Emergency knowledge management and social media technologies: A case study of the 2010 Haitian earthquake,” International Journal of Information Management, Vol. 31, Issue 1, pp. 6-13 (2011).

[2] S. Doan, B. H. Vo, and N. Collier, “An Analysis of Twitter Messages in the 2011 Tohoku Earthquake,” Electronic Healthcare, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 91, pp. 58-66 (2012).

[3] K. J. Hole, “Toward Risk Assessment of Large-Impact and Rare Events,” IEEE Security & Privacy, Vol. 8, Issue 3, pp. 21-27 (2010).

[4] A. F. Osborn, “Applied Imagination: Principles and Procedures of Create Problem Solving (Third Edition),” Charles Scribner’s Son, New York, (1963).

[5] J. Kawakita, “The Original KJ Method (Revised Edition),” Kawakita Research Institute (1991).

[6] T. Buzan with B. Buzan, “The Mind Map Book,” BBC WorldWide Limited (1993).

[7] M. Brassard, and D. Ritter, "The Creativity Tools Memory Jogger," Goal/QPC (1998).

[8] M. Brassard, "The Memory Jogger Plus+ Featuring the Seven Management and Planning Tools," Goal/QPC (1996).

[9] S. Kunifuji, and N. Kato, "Consensus-making support systems dedicated to creative problem solving," International Journal of Information Technology Decision Making, Vol. 6, No. 3, pp. 459–474 (2007).

[10] Y. Ohmi, K. Kawai, and H. Ohiwa, "A Card-handling Tool for Multimedia," Proc. 1999 IEEE International Conference on Multimedia Computing & Systems, IEEE CS, pp. II-250–254 (1999).

[11] J. Munemori, H. Fukuda, and J. Itou, "Application of a Web Based Idea Generation Consistent Support System," Proceedings of the 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2012), pp. 1827–1836 (2012).

[12] A. Abe, T. Sasaki, and N. Odajima, "Development and Operational Evaluation of the Groupware Based on Geographical Location Information for Local Community Activities," IPSJ Journal, Vol. 45, No. 1, pp. 155–163 (2004) (in Japanese).

[13] F. Geyer, and H. Reiterer, "Experiences from Employing Evernote as a Tool for Documenting Collaborative Design Processes," Proceedings of the ACM Conference on Designing Interactive Systems (DIS 2012), Workshop: Supporting Reflection in and on Design Processes (2012).

[14] T. Takahashi, H. Vermeulen, H. Ueda, and Y. Konagaya, "Digital Card Cabinet - Gathering and Sharing Impressions about a Exhibition with both Analog and Digital Ways," Proceedings of the VRSJ the 16th Annual Conference, pp. 550–553 (2011) (in Japanese).

[15] T. Umesao, "The Art of Intellectual Productivity," Iwanami-shoten (1969) (in Japanese).

[16] T. Ajiki, H. Fukuda, T. Kokogawa, J. Itou, and J. Munemori, "Application to the Disaster Data of an Idea Generation Consistent Support System," Proceedings of the 7th International Symposium on Frontiers of Information Systems and Network Applications (FINA 2011) in conjunction with IEEE AINA 2011, pp. 153–158 (2011).

[17] J. Munemori, T. Nishide, T. Fujita, and J. Itou, "Development of a Distributed Pictograph Chat Communicator IV," Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2011), Vol. 6883, pp. 77–85 (2011).

[18] L. Hearn, A Living God, "Gleanings in Buddha-Fields," Houghton Mifflin Company (1897).

[19] K. Yagishita, J. Munemori, and M. Sudo, "A Proposal and an Application of an Evaluation Method for Sentences of B Type KJ Method Based on Contents and Structures," IPSJ Journal, Vol. 39, No. 7, pp. 2029–2042 (1998) (in Japanese).

[20] T. L. Saaty, "The Analytic Hierarchy Process," McGraw-Hill (1980).

(Received October 19, 2012)
(Revised December 4, 2012)

Tomohiro Kokogawa received the B.E. and M. E degrees in Control Engineering from Osaka University. He joined Nippon Telegraph and Telephone Corp. in 1993, and currently worked as a senior research engineer of NTT Network Technology Laboratories. He is also currently a graduate school student of Wakayama University. He specializes in information sharing, public service systems, and creativity support systems. He is a member of IPSJ and IEEE.

Toshihiro Ajiki received the B.E. and M.E. degrees in Design and Information Sciences from Wakayama University, 2010 and 2012, respectively. In 2012, he joined NTT DOCOMO, INC. He is a member of IPSJ.

Junko Itou received the M.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 2001. In 2005, she joined the Department of Design and Information Sciences at Wakayama University, as an assistant researcher, and from 2007, she has been an assistant professor of Wakayama University. Her interests are human computer interface, groupware, image processing. She is a member of IPSJ.

Jun Munemori received the B.E. and M.E. degrees in electrical engineering from Nagoya Institute of Technology, Nagoya, Japan, the D.E. degree in electrical and electrical communication engineering from Tohoku University, Sendai, Japan, in 1979, 1981, and 1984, respectively. He worked in Mitsubishi Electric Corp., Kagoshima University, and Osaka University. He is currently a professor of Department of Design and Information Sciences at Wakayama University. His interests are groupware, human interface, and neurophysiology. He received IPSJ SIG Research Award, IPSJ Best Paper Award, IEEE CE Japan Chapter Young Paper Award, and KES2005 Best paper award, in 1997, 1998 2002, and 2005, respectively. He is a member of ACM, IEEE, IPSJ and IEICE.

Implementation of a Prototype Bi-directional Translation Tool between OCL and JML

Kentaro Hanada[†], Hiroaki Shinba[†], Kozo Okano[†]and Shinji Kusumoto[†],

[†]Graduate School of Information Science and Technology, Osaka University, Japan
{k-hanada, h-shimba, okano, kusumoto}@ist.osaka-u.ac.jp

Abstract - Object Constraint Language (OCL), which is an annotation language for the Unified Modeling Language (UML), can describe specifications more precisely than can natural languages. In recent years, model-driven architecture (MDA) based techniques have emerged, and thus translation techniques such as translation from OCL to the Java Modeling Language (JML) have gained much attention. Our research group has been studying not only a translation method from OCL to JML but also from JML to OCL. Bi-directional translation between OCL and JML supports (1) development by round-trip engineering (RTE) at the design level, and (2) multi-translations between various formal specification languages. This paper presents our implementations based on model translation techniques.

Keywords: model-driven architecture, OCL, JML, design by contract

1 Introduction

In recent years, model-driven architecture (MDA) [14] based techniques have emerged. MDA targets numerous languages. Thus, translation techniques such as translation from the Unified Modeling Language (UML) to some program languages have gained much attention. Several research efforts have proposed methods that automatically generate Java skeleton files from UML class diagrams [6], [11]. Some of these are publicized as plug-ins for Eclipse. Translation techniques such as the Object Constraint Language (OCL) [20] to the Java Modeling Language (JML) [15] have also been studied. These two languages are described as follows.

- OCL describes detailed properties of UML and is standardized by the Object Management Group (OMG).

- JML specifies properties of a Java program. It is also used in some static program analyzers such as the Extended Static Checker for Java (ESC/Java2) [8].

However, JML describes more detailed properties than does OCL. Both OCL and JML are based on design by contract (DbC) [18] and are able to provide property descriptions of classes or methods.

We previously proposed a method that translates a UML class diagram with OCL into a Java skeleton with JML [19]. Our translation tool is implemented by mapping each of the statements in OCL and JML by a Java program. However, model translation, which uses abstract models to represent common aspects of the target languages, is the primary function of MDA. One of our original goals was providing uniform techniques to translate from OCL to many specification languages. Our previous prototype of a translation tool and other tools provided by other researchers [19], [23] have low reusability, because the goals were fulfillment of translation, not usability. Thus, we need a tool that supports both translation and usability.

This paper presents a prototype translation tool from OCL to JML. First, we define the syntax of UML with OCL by using Xtext, which is a plug-in for Eclipse [5]. Next, we describe the translation rules from UML with OCL to a Java skeleton with JML. The syntax and rules are used for translation in the framework provided by Xtext. The syntax description is independent of the translation rules in Xtext; therefore, the syntax part has high reusability. However, because Xtext can generate a dedicated editor of the defined syntax, this editor has high usability functions, such as code completion and detection of syntax errors.

We also implement a tool that translates from JML to OCL by using the same approach as translation from OCL to JML. Round-trip Engineering (RTE) [17],[25] is a method that gradually refines a model and source code by the repeated use of forward engineering and reverse engineering. The aim of implementation of translation from JML to OCL is to support RTE at the specification description level.

The organization of the remainder of the paper is as follows. Section 2 describes the background of this research and related work. Sections 3, 4, and 5 describe the implementation of our tool, the experimental results, and discussions, respectively. Finally, Section 6 concludes the paper.

2 Background

In this section, we present the background of our research, including techniques and related work.

2.1 Design by Contract

DbC is one of the concepts of object-oriented software designing. The concept regards specifications between a supplier (method) and a client (calling the method) as a contract, with the goal of enhancing software quality, reliability, and reusability. The contract means that if a caller of a class ensures the pre-condition, then the class of the caller must also ensure the post-condition. A pre-condition is a condition that should be satisfied when a method is called. For example, conditions for the arguments of a method are pre-conditions. In contrast, a post-condition is a condition that should be satisfied when a process of a method ends. If the pre-condition is not satisfied, then the caller of its class has errors, and if the post-condition is not satisfied, then the class has errors. These

ISSN1883-4566 © 2013 - Informatics Society and the authors. All rights reserved.

separate responsibilities have a clear distinction for developers, and so they are useful to identify the causes of software defects.

2.2 OCL and JML

OCL, which is standardized by OMG, details the properties of UML models. Because a UML diagram alone cannot express the rich semantics of the relevant information of an application, OCL allows one to describe precisely the additional constraints on the objects and entities present in the UML model.

JML details the constraints of Java methods or objects [15]. These constraints are based on DbC. It is easy for novices to describe the properties in JML because the syntax of JML is similar to that of Java. Various kinds of tools verify source codes with JML annotations. For example, JML Runtime Assertion Checker (JMLrac) [24] checks whether contradictions exist between JML constraints and runtime values of the program. JMLUnit automatically generates a test case skeleton and a test method for JUnit [1]. Since the original use of JML was for runtime assertion checking [4], several other program verification tools have been developed, such as ESC/Java(2) [7], [13], JACK [3], KeY [2], and Krakatoa [16].

2.3 Model Translation

The Query Verification Tool (QVT) [9] and the ATL Transformation Language (ATL) [12] are typical model translation techniques. Model translation has two types. One is Model2Model (M2M) that translates from model to model, and the other is Model2Text (M2T) that translates from model to code. For example, UML2Java [6] provides M2T translation capability.

2.4 Round-trip Engineering

RTE is a method that gradually refines the model and the source code by the repeated use of forward engineering and reverse engineering. RTE development needs to keep the conformity of the models with the source code. By using RTE, QVT feature and requirement changes are easier to make [17], [25]. In general, when the code or models are changed, then the corresponding code or models are changed automatically by using a tool supporting RTE.

2.5 Xtext

Xtext [5] is a support framework for defining both the syntax of a model and the translation rules from the model to the text. Xtext can generate a dedicated editor of the defined syntax. This editor has high usability functions, such as code completion and detection of syntax errors. Moreover, if textual models are written on the editor, the models are automatically translated to text according to the defined translation rules.

2.6 Related Work

Some existing methods [10][23] do not adequately support the iterator feature, which is the most basic operation among collection loop operations. Our research group proposed a technique to resolve this problem by inserting a Java method that is semantically equal to each OCL loop feature [19].

```
private T1 mPrivateUseForJML01(){
  μ(init);
  for (T2 e: μ(c1))
    res = μ(body)
  return res;
}
```

Figure 1: General Java template for the iterate feature method

An iterate feature is an operation that applies an expression given as the argument to each element of a collection, which is given as another argument.

$$\text{Set}\{1,2,3\}{-}{>}\ \text{iterate}(i\text{: Integer;}\quad sum : \text{Interger} = 0 \mid sum + i) \tag{1}$$

Expression (1) defines an operation that returns a value representing the sum of all elements in the Set. In expression (1), the first argument ($i : Integer$) defines an iterator variable. The second argument ($sum : Integer = 0$) defines a variable used to store the return value and its initialization. The third argument ($sum + i$) defines the expression executed iteratively in the loop.

In JML or Java, expressions such as "$sum + i$" cannot be evaluated dynamically. For example, if expression (1) is resolved in the same way as expression (2), the result of the translation would be expression (3).

$$\text{JMLTools.flatten}(setOfSets) \tag{2}$$

$$\text{JMLTools.iterate}(\text{int } i, \text{int } sum = 0, sum + i, set) \tag{3}$$

In expression (3), "$sum + i$" is evaluated only once when the method is called. In other words, the expression is not evaluated iteratively and dynamically in every collection element.

To resolve this problem, our research group proposed a technique that inserts a Java method that is semantically equal to each OCL loop feature [22]. It is worthwhile to have such an algorithm to deal with the iterate feature, because the iterate feature is widely used.

Expression (4) shows the general format of an iterate feature. The variables e, $init$, $body$, and c indicate an iterator variable, a declaration of the return value and its initialization, an expression executed in the loop, and a Collection type variable, respectively.

$$c{-}{>} \quad \text{iterate}(e; init \mid body) \tag{4}$$

Figure 1 shows the general format of our newly created method. The keywords $\mu()$, T_1, and T_2 and the variable res are a function translating an OCL expression into a Java expression, a variable declared in $init$, a variable e, and the name of the variable declared in $init$, respectively.

3 Implementation

In this section, we present the implementation of our translation tool.

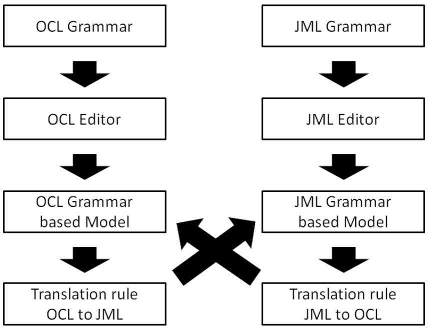


Figure 2: Overview of implementation using Xtext

3.1 Policy of Implementation

We implement the translation tools by using Xtext. First, we define the syntax of the models. Next, we define the translation rules from the syntax of the models to the source code. Both translations, from OCL to JML and from JML to OCL, are implemented by the above method. Figure 2 shows the overview of the implementation.

Our implementation method has the following advantages.

- Syntax and translation rules are defined independently; thus, the syntax description can be reused.
- Xtext can generate a dedicated editor of the defined syntax. The high usability functions are explained in the previous section.

3.2 Translation from OCL to JML

In this section, we present the implementation of a translation from OCL to JML.

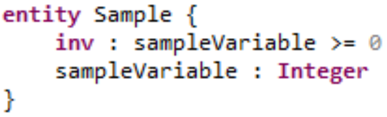
3.2.1 Syntax definition of UML with OCL annotation

We define the syntax of the UML class diagram with OCL. For the UML part, we use conventional syntax rules and extend the syntax. The extended syntax can append the OCL constraints. For the OCL part, we consider some cases of return types and other syntax. Translation rules depend on the syntax of the model; therefore, careful case analysis helps the semantic analysis and enhances the reusability of the syntax of a model. The function of the generated editor depends on the defined syntax. Therefore, the more we take into account the case analysis, the more usability the generated editor has. In summary, careful consideration of the case analysis helps both usability and reusability.

3.2.2 Definition of translation rule from OCL to JML

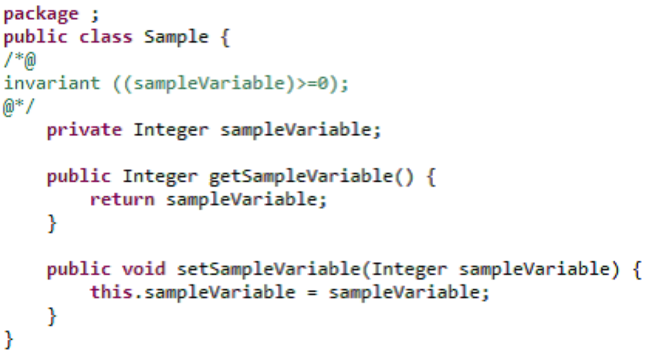
Table 1 shows parts of the translation rules of OCL to JML. A translation function of an OCL statement to a JML statement is expressed by μ. Here, Integer, Real, and any type of Boolean are expressed by a_i. Any type of Collection is expressed by c_i.

We define the translation rules OCL-JML in accordance with many of the same rules used in existing research [19]. In Table 2, many collection loops can be replaced by iterate features. Therefore, our current research replaces the collection loop with the iterate feature. However, this translation method has some challenges. For example, low readability of the generated code is one challenge. To resolve this problem, if the OCL loop feature directly translates the JML loop feature, we do not replace the collection loop with the iterate feature.



```
entity Sample {
    inv : sampleVariable >= 0
    sampleVariable : Integer
}
```

Figure 3: Input model



```
package ;
public class Sample {
/*@
invariant ((sampleVariable)>=0);
@*/
    private Integer sampleVariable;

    public Integer getSampleVariable() {
        return sampleVariable;
    }

    public void setSampleVariable(Integer sampleVariable) {
        this.sampleVariable = sampleVariable;
    }
}
```

Figure 4: Result of translation from OCL to JML

Figure 3 is an example of a textual model based on the defined syntax. Figure 4 is an example of the result of a translation from the model to the text.

3.2.3 Oclvoid Type

The OclVoid type is a class having only the constant named Undefined. The constant is returned when an object is cast into an unsupported type or when a method gets a value from the empty collection. The counterpart of this constant in JML is null. It must be noted that in OCL, a logical expression such as “True or Undefined” is evaluated as an undefined expression, not True. To deal with OclVoid correctly, the translation

Table 1: μ translation rules from OCL to JML

$\mu(a_1 = a_2)$	$=$	$\mu(a_1) == \mu(a_2)$
$\mu(a_1 > a_2)$	$=$	$\mu(a_1) > \mu(a_2)$
$\mu(a_1 < a_2)$	$=$	$\mu(a_1) < \mu(a_2)$
$\mu(a_1 >= a_2)$	$=$	$\mu(a_1) >= \mu(a_2)$
$\mu(a_1 <= a_2)$	$=$	$\mu(a_1) <= \mu(a_2)$
$\mu(a_1 <> a_2)$	$=$	$\mu(a_1)! = \mu(a_2)$
$\mu(c_1 = c_2)$	$=$	$\mu(c_1)$.equals($\mu(c_2)$)
$\mu(c_1 > c_2)$	$=$	$\mu(c_1)$.containsAll($\mu(c_2)$)&&!$\mu(c_1)$.equals($\mu(c_2)$)
$\mu(c_1 < c_2)$	$=$	$\mu(c_2)$.containsAll($\mu(c_1)$)&&!$\mu(c_1)$.equals($\mu(c_2)$)
$\mu(c_1 >= c_2)$	$=$	$\mu(c_1)$.containsAll($\mu(c_2)$)
$\mu(c_1 <= c_2)$	$=$	$\mu(c_2)$.containsAll($\mu(c_1)$)
$\mu(c_1 <> c_2)$	$=$	!$\mu(c_1)$.equals($\mu(c_2)$)
$\mu(c_1 ->$size())	$=$	$\mu(c_1)$.size()
$\mu(c_1 ->$isEmpty())	$=$	$\mu(c_1)$.isEmpty()
$\mu(c_1 ->$notEmpty())	$=$	!$\mu(c_1)$.isEmpty()
$\mu(c_1 ->$excludes(a_1))	$=$	$\mu(c_1 ->$count(a_1) $= 0)$
$\mu(c_1 ->$count(a_1))	$=$	$\mu(c_1 ->$iterate(e; acc : Integer $= 0 \vert$ if $e = a_1$ then $acc + 1$ else acc endif))

tool needs to treat OclVoid as follows.

$$(a_1\ ==\ null\,?\,false\ :\ throw\,new\,JMLException())$$

3.3 Translation from JML to OCL

In this section, we present the implementation of the translation from JML to OCL.

3.3.1 Syntax definition of Java skeleton code with JML annotation

We define the syntax of Java skeleton with JML. For Java, we define the syntax of class declaration, class modifier, field variable, and method declaration as the targets of translation. The variable type and others are needed to translate correctly, so we define the syntax of the Java skeleton. For JML, our translation tool can translate a part of the formula defined in the JML Reference Manual. JML is a more detailed language than OCL, and JML has complex expressions that cannot be expressed by OCL. For example, JML has an assignment operation and a shift operation, but OCL does not have either of these operations. At the time of syntax definition, we omit the operations and syntax that cannot be translated from JML to OCL. By omitting syntax that does not support translation from JML to OCL, a user can input only the JML expressions supported by the generated editor. For this reason, it becomes much easier to understand the corresponding syntax.

3.3.2 Definition of translation rule from JML to OCL

Table 3 shows some of the translation rules from JML to OCL. Here, the translation function of an JML statement to a OCL statement is expressed by μ'. Any type of all are expressed by a_i. A type of boolean is expressed by b_i.

In terms of elementary operation, the translation of JML to OCL only has to replace the operator of JML with the operator of OCL. However, to translate correctly, a part of the operator needs to interchange an operand. The syntax of JML is similar to that of Java. For example, the "+ operator" is used in various cases, such as "Integer + Integer" and "String + Integer". OCL does not support operation on different types. In contrast, JML supports "+ operator" involving non-numerical types. In terms of loop operation, exists and forall and other terms are defined as operations of the Collection type in OCL. However, sometimes exists and forall and other terms are used as a for loop of Java in JML. Therefore, the loop operation of JML cannot be translated by the loop operation of OCL. If the loop operation is used as a Collection in JML, our tool translates JML to OCL. If the loop operation is not used as a Collection in JML, our tool outputs error messages.

Table 2: A part of the correspondence table of Collection-Iterate

c_1−>exists($a_1 \mid a_2$)	=	c_1−>iterate(a_1; res : Boolean = false $\mid$ res or a_2)
c_1−>forAll($a_1 \mid a_2$)	=	c_1−>iterate(a_1; res : Boolean = true $\mid$ res and a_2)

c_1−>count(a_1)	=	c_1−>iterate(
		e; acc : Integer = 0 $\mid$
		if $e = a_1$ then $acc + 1$
		else acc endif)

st_1−>select($a_1 \mid a_2$))	=	st_1−>iterate(a_1; res :
		Set(T) = Set {} $\mid$
		if a_2 then res −>includeing (a_1)
		else res endif)

st_1−>reject($a_1 \mid a_2$))	=	st_1−>select(a_1 $\mid$ not a_2)
c_1−>any($a_1 \mid a_2$)	=	c_1−>select($a_1 \mid a_2$)−> asSequence()−>first()
c_1−>one($a_1 \mid a_2$)	=	c_1−>select($a_1 \mid a_2$)−>size()= 1

3.4 Type Inference

In OCL, "==" evaluates whether two objects are equivalent. However in JML, "==" evaluates whether two objects are equivalent, and "$equals()$" method evaluates whether two reference types are equivalent. To translate correctly, the variable type, and so on, must be correctly distinguished. When translating from JML to OCL, our tool can distinguish the type information correctly. However, when a user writes a textual model, our tool cannot distinguish the type information.

4 Experiments

This section explains our experiments in detail.

4.1 Overview of Experiments

We conducted two experiments. The goal of the first experiment (Experiment 1) was to evaluate the quality of translation from JML, described as the experimental object, to OCL. The goal of the second experiment (Experiment 2) was to evaluate the quality of translation from OCL, generated by our translation tool, to JML. These experiments were conducted to ensure that our tool has possible applications for RTE.

4.2 Measurements

To evaluate the results of translation, we measured the following two items.

Ratio of Transformation

$$Ratio = OCL_{translated}/JML_{all}$$

Table 3: μ' translation rules from JML to OCL

$\mu'(b_1?b_2{:}b_3\ )$	=	if $\mu'(b_1)$ then $\mu'(b_2)$
		else $\mu'(b_3)$ endif

$\mu'(b_1{<}{=}{=}{>}b_2\ )$	=	$\mu'(b_1)$= $\mu'(b_2)$
$\mu'(b_1{<}{=}!\ {=}{>}b_2\ )$	=	$\mu'(b_1) <> \mu'(b_2)$
$\mu'(b_1{=}{=}{>}b_2\ )$	=	$\mu'(b_1)$ implies $\mu'(b_2)$
$\mu'(b_1{<}{=}{=}b_2\ )$	=	$\mu'(b_2)$ implies $\mu'(b_1)$
$\mu'(b_1\&\&b_2\ )$	=	$\mu'(b_1)$ and $\mu'(b_2)$
$\mu'(b_1\vert\vert b_2\ )$	=	$\mu'(b_1)$ or $\mu'(b_2)$
$\mu'(b_1\vert b_2\ )$	=	$\mu'(b_1)$ or $\mu'(b_2)$
$\mu'(b_1\ \hat{}\ b_2\ )$	=	$\mu'(b_1$ xor μ'(b)
$\mu'(b_1\&\ b_2\ )$	=	$\mu'(b_1)$ and $\mu'(b_2)$
μ'(\result)	=	result
μ'(\old(a_1))	=	$\mu'(a_1)$@pre
μ'(\not_modified(a_1))	=	$\mu'(a_1)$ = $\mu'(a_1)$@pre
μ'(\fresh(a_1))	=	$\mu'(a_1)$.oclIsNew()

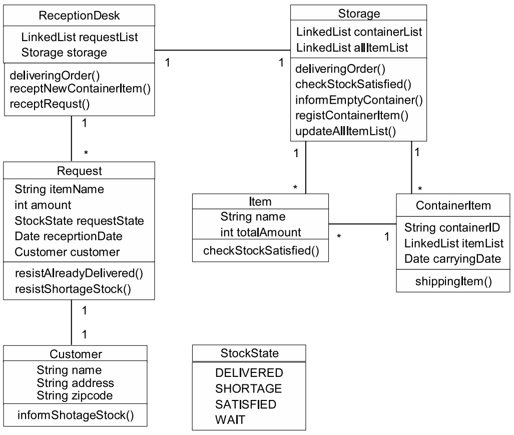


Figure 5: UML class diagram of warehouse management program

Ratio of Reverse Transformation

$$Ratio = JML_{reverse}/OCL_{translated}$$

JML_{all} is the number of pre-conditions and post-conditions. $OCL_{translated}$ is the number of OCL statements translated from JML statements by our translation tool. $JML_{reverse}$ is the number of JML statements translated from generated OCL statements by our translation tool.

4.3 Results of Experiments

4.3.1 Experiment 1

Experiment 1 uses a warehouse management program. Figure 5 shows the class diagram of the warehouse management program, which consists of seven classes. Table 4 shows the components of the warehouse management program in detail.

The warehouse management program [21] has correct JML statements, as shown by the results of past research [21]. The number of described pre-conditions, post-conditions, and class-invariants is 130. We use these statements to evaluate the quality of the translation. The result shows that the number of correctly translated statements is 102, and the Ratio of Transformation is 78.4%. Figures 6 and 7 show cases of failure translations.

Many cases of failure translations can be found. For example, if multi-variables are declared in the forall feature, then the translation from JML to OCL fails. Additionally, we can classify the following expressions as failures: expressions with type operations, typeof operations, applying ”+” between a String type and numeric type expressions, and so on.

Table 4: Components of warehouse management program

Class Name	# of methods	# of lines
ContainerItem	12	224
Customer	10	156
Item	7	110
ReceptionDesk	8	162
Request	16	245
StockState	0	9
Storage	10	258
TOTAL	63	1164

```
/*@
ensures \result.matches("containerID." + containerID
          + "CarryingDate | " + carryingDate + "\n{1}")
@*/
String toString(){
}
/*@
ensures (\forall Request r; requestList.contains(r);
         r.getAmount() > 0);
ensures (\forall Request r; requestList.contains(r)
         && r.getAmount() != \old(r.getAmount());
         r.getRequestState() == StockState.SHORTAGE);
@*/
List deliveringOrder(){
}
```

Figure 6: Example of failure translation from JML to OCL (input)

```
context ContainerItem::toString()::String
post : result.matches('ContainerID.'
       [type error][type error][type error][type error])

context ReceptionDesk::deliveringOrder()::List
post : requestList->forAll(r:Request|r.getAmount() > 0)
post : requestList and r=(r)@pre and ->forAll(
       r:getRequestState() = StockState.SHORTAGE)
```

Figure 7: Example of failure translation from JML to OCL (output)

4.3.2 Experiment 2

In Experiment 1, 102 statements are translated correctly. We recheck whether these generated statements are recognized as translation objects of the prototype translation tool from OCL to JML. In terms of correctly translated OCL, the Ratio of Transformation of translation from OCL to JML is 100%. For this reason, translation from JML to OCL by our tool has no problems. However, some bugs are found in the translation from OCL to JML, because our translation rule is still in the trial phase. As a result, 98 statements out of 102 statements as input statements are translated correctly, and the Ratio of Transformation is 96.1%. The result shows that four statements have some bug. Figures 8 and 9 show examples of failure cases.

The OclAsType method is described in the lexical specification. However, the OclAsType method is not described in the translation rules, so our tool could not translate the OclAsType method. After reviewing these results, we modified the method to successfully translate the four statements. Therefore, we will apply our modified translation rule in future work.

5 Discussions

As stated earlier, the result of the Ratio of Transformation of the translation from JML to OCL is 78.4% in Experiment 1. We implemented our tool as a prototype, so our tool has unsupported statements. However, the Ratio of Transformation of the experimental result shows that majority of JML consisted of elementary operations, and thus shows the validity

```
pre : o.oclIsTypeOf(Request)
post : result = (receptionDate.getTime()-
       (o.oclAsType(Request)).getReceptionDate())
       .oclAsType(Integer) or result = 0
op compareTo(o : Object)
```

Figure 8: Example of failure translation from OCL to JML (input)

```
/*@
requires o.getClass().equals(Request);
ensures (\result == (receiptionDate.getTime()-
         ((o.oclAsType(Request)).getReceiptionDate()))
         .oclAsType(Integer)) || (\result == 0);
@*/
public void CompareTo(Object o){
}
```

Figure 9: Example of failure translation from OCL to JML (output)

of our translation tool. We now describe a part of the failure translation.

Our tool could not translate the \type keyword, which is a primitive operator returning a type name. The reason for the above situation is that OCL has no counterpart of the \type operator to identify a type name from a designated expression. To solve this problem, the following approach is considered. First, our tool keeps information on the parameter type before translation from JML to OCL. Next, our tool outputs the parameter type directly in OCL statements.

Result of Ratio of Reverse Transformation is 96.1% in Experiment 2. In Experiment 2, some unsuccessful translated statements also occur in the translation result, because our translation tool from OCL to JML is a prototype. The input OCL is recognized as correct input; therefore, the result shows that the quality of translated OCL is not a problem, but the translation rules have some imperfections.

For this reason, the generated OCL has high quality. Some of the failure translations are due to omissions in the implementation. In terms of this failure translation, our tool will be able to translate correctly with the modified implementation.

Next, we will examine correctness of the rules. Table 1 and 3 show a part of the rules. In general, we have to check that successive application of μ and μ' and vice versa, are preserved. I.e., $\mu'(\mu(o)) = o$ and $\mu(\mu'(j)) = j$ must hold, where (o and j are an OCL expression and a JML expression, respectively). We have manually checked that it holds for every combination of elementary operations. For example, $\mu(\mu'(\backslash result)) = \backslash result$ hold. However, for the iterate operator, some expressions cannot be preserved.

$\mu'(\mu(c_1{-}{>}iterate(a_1; res : \text{Boolean} = \text{false} \mid res \text{ or } a_2)))$
$= \mu'(mPrivateUseForJML01())$
$= mPrivateUseForJML01()$ is a one of such concrete examples. To deal with such expressions is one of our future works.

6 Conclusion

This paper presents a method of implementing the translation from OCL to JML and from JML to OCL. The aim of the implementation of translation from JML to OCL is to support RTE at the specification description level. We applied our tool to a warehouse management program as an experimental object and showed the results of the experiments. One future work is to complete our translation tool, because our tool is at the experimental stage. For example, our tool cannot treat Undefined correctly and needs to be modified.

There are some expressions which cannot be translated correctly by our method including the expressions with iterate operation, and JML loop expressions. To deal with such expressions is one of our future works.

After we improve the implementation of our tool, we will conduct additional experiments. We will again evaluate the quality of translation from OCL to JML and from JML to OCL. We have not yet evaluated the translation from OCL to JML, except for the number of successful translations.

In the future, we will also compare the result of applying generated JML with the review tool for JML and the result of applying described JML manually with the review tool for JML. Two examples of review tools for JML are esc/java2 and jml4c. In terms of the translation tool from JML to OCL, we will compare the generated OCL and the OCL described manually to evaluate the readability. Also, we will apply the generated OCL to the review tool for OCL. One example of a review tool for OCL is Octopus. In addition, we will evaluate whether our tool can do mutual transformations repeatedly by using our translation tool from OCL to JML and from JML to OCL.

7 Acknowledgments

This work is conducted as a part of a Grant-in-Aid for Scientific Research C (21500036).

REFERENCES

[1] JUnit. http://www.junit.org/.

[2] W. Ahrendt, T. Baar, B. Beckert, M. G. R. Bubel and, R. Hahnle, W. Menzel, W. Mostowski, A. Roth, S. Schlager, and P. Schmitt. The KeY tool. *Software and System Modeling*, 4(1):32–54, 2005.

[3] L. Burdy, A. Requet, and J.Lanet. Java applet correctness: A developer-oriented approach. *K. Araki, S. Gnesi, and D. Mandrioli, editors, FME 2003*, 2805:422–439, 2003.

[4] Y. Cheon and T. Leavens. A runtime assertion checker for the Java Modeling Language (JML). *In Hamid R. Arabnia and Youngsong Mun, editors, the International Conference on Software Engineering Research and Practice (SERP'02)*, pages 322–328, 2002.

[5] Eclipse Foundation. Xtext - Language Development Framework. http://www.eclipse.org/Xtext/.

[6] G. Engels, R.H.ücking, S. Sauer, and A. Wagner. UML collaboration diagrams and their transformation to Java. In *UML1999 -Beyond the Standard, Second International Conference*, pages 473–488, 1999.

[7] C. Flanagan, K. Rustan, M. Leino, M. Lillibridge, G. elson, J. Saxe, and R. Stata. A runtime assertion checker for the Java Modeling Language (JML). *Extended static checking for Java. In ACM SIGPLAN 2002 Conference*

on Programming Language Design and Implementation (PLDI'2002), pages 234–245, 2002.

[8] C. Flanagan, K. Rustan, M. Leino, M. Lillibridge, G. Nelson, J. Saxe, and R. Stata. Extended static checking for Java. In *Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation*, pages 234–245, 2002.

[9] O. M. Group. Documents associated with meta object facility (mof) 2.0 query/view/transformation, v1.1, 2011. http://www.omg.org/spec/QVT/1.1/PDF/.

[10] A. Hamie. Translating the Object Constraint Language into the Modeling Language. In *In Proc. of the 2004 ACM symposium on Applied computing*, pages 1531–1535, 2004.

[11] W. Harrison, C. Barton, and M. Raghavachari. Mapping UML designs to Java. In *Proc. of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 178–187, 2000.

[12] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev. ATL: A model transformation tool. *Science of Computer Programming*, 72(1-2):31–39, 2008.

[13] J. Kiniry and D. Cok. ESC/Java2: Uniting ESC/Java and JML. *Construction and Analysis of Safe, Secure and Interoperable Smart devices (CASSIS'2004)*, 3362:108–128, 2005.

[14] A. Kleppe, J. Warmer, and W. Bast. *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2003.

[15] G. Leavens, A. Baker, and C. Ruby. JML: A Notation for Detailed Design. *Behavioral Specifications of Businesses and Systems*, pages 175–188, 1999.

[16] C. Marche, C. Paulin-Mohring, and X. Urbain. The KRAKATOA tool for certification of Java/JavaCard programs annotated in JML. *J. Log. Algebr. Program*, 58(1-2):89–106, 2004.

[17] N. Medvidovic, A. Egyed, and D. S. Rosenblum. Round-trip software engineering using uml: From architecture to design and back, 1999.

[18] B. Meyer. *Eiffel: the language*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1992.

[19] K. Miyazawa, K. Hanada, K. Okano, and S. Kusumoto. Class enhancement of our ocl to jml translation tool and its application to a curriculum management system. *In IEICE Technical Report*, 110(458):115–120, 2011.

[20] Object Management Group. OCL 2.0 Specification, 2006. http://www.omg.org/cgi-bin/apps/doc?formal/06-05-01.pdf.

[21] M. Owashi, K. Okano, and S. Kusumoto. Design of Warehouse Management Program in JML and Its Verification with Esc/Java2 (in Japanese). *The IEICE Transaction on Information and Systems*, 91(11):2719–2720, 2008-11-01.

[22] M. Owashi, K. Okano, and S. Kusumoto. A Translation Method from OCL into JML by Translating the Iterate Feature into Java Methods (in Japanese). *Computer Software*, 27(2):106–111, 2010.

[23] M. Rodion and R. Alessandra. Implementing an OCL to JML translation tool. 106(426):13–17, 2006.

[24] A. Sarcar and Y. Cheon. A new Eclipse-based JML compiler built using AST merging. *Department of Computer Science, The University of Texas at El Paso, Tech. Rep*, pages 10–08, 2010.

[25] S. Sendall and J. Küster. Taming model round-trip engineering. In *In Proceedings of Workshop Best Practices for Model-Driven Software Development*, pages 1–13, 2004.

(Received October 15, 2012)
(Revised December 3, 2012)

Kentaro Hanada received the BI degree from Osaka University in 2011. He is a master course student in Osaka University. His research interests include model translation, especially translation between OCL and JML.

Hiroaki Shinba received the BI degree from Osaka University in 2012. He is a master course student in Osaka University. His research interests include model translation, especially translation between OCL and JML.

Kozo Okano received the BE, ME, and Ph.D degrees in Information and Computer Sciences from Osaka University, in 1990, 1992, and 1995, respectively. Since 2002, he has been an associate professor in the Graduate School of Information Science and Technology, Osaka University. In 2002, he was a visiting researcher of the Department of Computer Science, University of Kent at Canterbury. In 2003, he was a visiting lecturer at the School of Computer Science, University of Birmingham. His current research interests include formal methods for software and information system design. He is a member of IEEE_CS, IEICE of Japan and IPS of Japan.

Shinji Kusumoto received the BE, ME, and DE degrees in information and computer sciences from Osaka University in 1988, 1990, and 1993, respectively. He is currently a professor in the Graduate School of Information Science and Technology at Osaka University. His research interests include software metrics and software quality assurance technique. He is a member of the IEEE, the IEEE Computer Society, IPSJ, IEICE, and JFPUG.

Automated Prevention of Failure in Complex and Large Systems: Fighting Fire with Fire

Behzad Bordbar[†]and Philip Weber[†]

[†]School of Computer Science, University of Birmingham, UK
{b.bordbar, p.weber}@cs.bham.ac.uk

Abstract - People, businesses and economies are increasingly dependent on Cloud and internet services. At the same time, the systems on which these services are built are becoming more complex and interdependent. The cost of failure is high, but systems are too complex for human detection of problems. We review methods for online fault diagnosis, process mining and Virtual Machine Introspection. We suggest bringing these techniques together for automated identification, diagnosis and prediction of risk of failure in large systems. We present examples from telecoms and Cloud industries in support of these ideas.

Keywords: Models, process mining, diagnosis, failure prevention, complex systems

1 INTRODUCTION

The Internet and the services it supports now play a key role in most people's daily lives, and in the day-to-day operation of business enterprises and nations. The Oxford Internet Surveys 2011 [1] surveyed over 2000 respondents in Britain and reported that over 80% of employees used the internet to obtain news and information. Usage for other purposes and by people in other 'life categories' was only slightly lower. Students were the largest consumers of media via the internet (over 90%), closely followed by other groups.

More importantly, the survey reported increasing levels of use of the internet for accessing critical services such as banking, grocery shopping and paying bills. Increasingly, citizens accessed government services online, with only 21% of households lacking internet access. Use of online services was highest among the young, wealthy and well-educated.

New technologies such as Cloud and virtualisation are enabling the move to internet-based systems architecture. Cloud enables computing resources to be provided on demand according to a utility model, using many instances of commodity hardware and software components shared between services. Costs of set up and ongoing provision of new services are thus reduced, since payment is only for the resources or time used. Businesses need no longer invest in costly infrastructure [2], [3]. Virtualisation abstracts services from the hardware, operating systems, storage and network. This makes the resources more flexible, increasing business value by increasing agility and resilience. Services are democratised by use of open Service-oriented Architectures (SoA) and standards such as SOAP and HTTP. The success of public clouds has seen private versions of the same concepts implemented within businesses.

However, new technologies and pace of change present new risks from system problems, and new opportunities for nefarious activities such as malware and cyber-attack. Heterogeneous systems (cloud and open standards) duplicated manyfold may all be affected by the same bug, security breach or performance problem [3]. Shared resources mean one problem may affect many services, and introduces the risk that the activities of one business may impact those of another. A business abstracting its infrastructure to a Cloud platform faces new questions of availability and performance unpredictability. Well documented outages to cloud services (see for example references from [2]) have taken many hours to resolve, each outage affecting many services.

Security can also be a concern. As far back as 2004, Byres *et al.* reported a steady rise in reported incidents of industrial problems caused by 'cyber attacks' [4]. They attributed this to increased use of heterogeneous interconnected systems, and the increasing attractiveness of targets due to the wide consequences possible. These factors are multiplied in today's online environments. Water [4] and power transmission industries [5] are given as examples of interconnected, critical, vulnerable industries which have been the subjects of attacks.

The security viewpoint provides extreme examples of the seriousness of potential damage caused by problems or attacks on highly interconnected systems, the difficulty of containing such problems and their potential to spread beyond the 'cyber' world to physical effects. Examples are the the Stuxnet attacks on Iranian nuclear facilities (e.g. [6]), and industrial 'cyber-crime' using 'botnets' (networks of many hijacked connected computers), reported to be responsible for disruption to Estonia's national networks in 2007, and in the 2008 Russia-Georgia war among others (also [6]).

We conclude that reliability of online services is of crucial importance. Problems may affect very many people simultaneously, prevent access to critical services, and have the greatest impact on the most economically and politically active groups. Service outages thus have the potential to impact economies, enterprises and national governments, both financially and through damaged reputations. 'The major problem for cloud computing is how to minimise such kinds of outage/failure to provide reliable services'[2]. A major challenge is how can we deliver such crucial services reliably while reducing cost?

In this paper we consider the internet- and cloud-based technologies underlying these services and describe three of the techniques used to tackle the above challenges. First we introduce model-based methods for automated detection of faults or undesirable scenarios using automated 'Diagnosers' (software modules or services) to diagnose occurrence of failure or undesirable scenarios in real time or near real time. Model-based techniques are powerful, but sometimes there are no models of the system available or it is very costly or even impossible to produce a model of the system. For example, systems produced from merging of legacy systems are often too complex and large to be modelled. Often there is no access to the designers of such systems and it is costly to re-engineer a model. However, most modern systems produce logs capturing run-time information for various purposes. The second group of techniques discussed in this paper applies Process Mining to extend this diagnosis framework to situations where we do not have models of the system. Finally, we address the specific challenges of diagnosis of faults related to occurrences of malicious behaviour in Cloud. There are similarities between malicious behaviour as malware writers tend to use components available on the web which are used in existing malware. We present a framework which uses symptoms caused by using such components to discover newly emerging malware. We present examples from telecoms services and security in the Cloud to describe the three sets of methods.

The paper is organised as follows. In section 2 we introduce terminology and concepts necessary for the remainder of the paper, particularly to specify what are system failures or undesirable behaviours. Section 3 describes the problem in more detail. The core of the paper is sections 4, 5 and 6 in which we discuss in depth our methods for diagnosis. Section 7 concludes the paper.

2 PRELIMINARIES

We describe terminology and concepts necessary to understanding the rest of the paper.

2.1 Service-Oriented Architectures (SoA)

A SoA is a distributed business application architecture where heterogeneous components communicate and provide services to each other using open standards. Examples of open standards are WSDL [7] and XSD [8], which are used to define the interfaces between services, and communication protocols such as SOAP and HTTP. The use of such open standards means that the services and protocols can be changed with minimal impact on the service. A simplified SoA, for broadband failure resolution in a telecoms business, is illustrated in Fig. 1 (described in full in section 4).

Most essential is the business process describing how these services interact to complete the task such as resolving a fault.

2.2 Business Processes and Process Mining

Business processes describe activities carried out to fulfil a business function, and the relations between them. Among other aspects we may describe the process ‘control-flow’, i.e. how the activities are related; interactions between people and organisations; or business rules or constraints. In this paper we are concerned with the control-flow. Business process are commonly represented formally by languages such as BPEL, BPMN and Petri nets. Figure 2 shows a BPMN model of part of the business process for broadband fault resolution.

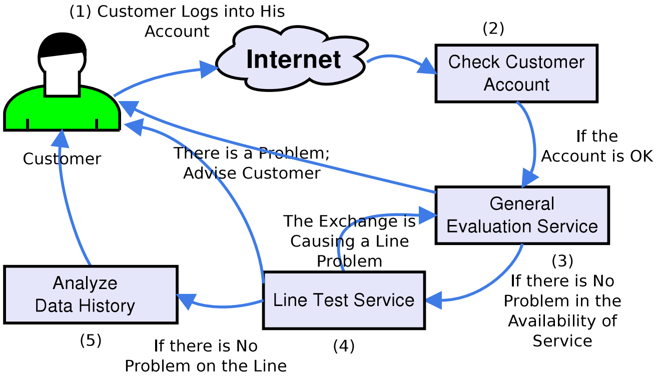

(1) Customer Logs into His Account
Internet
(2)
Check Customer Account
If the Account is OK
Customer
There is a Problem; Advise Customer
General Evaluation Service
The Exchange is Causing a Line Problem
(3)
If there is No Problem in the Availability of Service
Analyze Data History
Line Test Service
(5)
If there is No Problem on the Line
(4)

Figure 1: An interaction between the Customer and System

Let A be a set of business activities. A single pass through the business process from start to end task is a *case*, for example processing one order, The *events* of their occurrence are recorded in an *Event log* E. We assume that as a minimum, each event e is recorded with a case ID c, activity name $a \in A$ and timestamp t. An event log can then be represented by a non-empty set of triples

$$E = \{e : e = (c, a, t)\}^+, \qquad (1)$$

assuming that timestamps are unique. Process mining algorithms [9], [10] use workflow logs to learn models of the business processes. We discuss process mining in section 5.

Process mining algorithms often assume that events are atomic (taking no time), are uniquely labelled (the same label always refers to the same event and vice versa), and make no use of additional information such as timing of events, merely the order in which they are recorded. We assume that the underlying process to be discovered is unchanging.

If the activities in our example service were encoded with symbols $a, b, \ldots$ from some alphabet Σ then (abstracting from detail) the ‘trace’ of one possible enactment of the process might be recorded in the event log as a string, e.g.‘$abcdef$’. These strings are also called *traces*. A workflow log W is a multiset over traces,

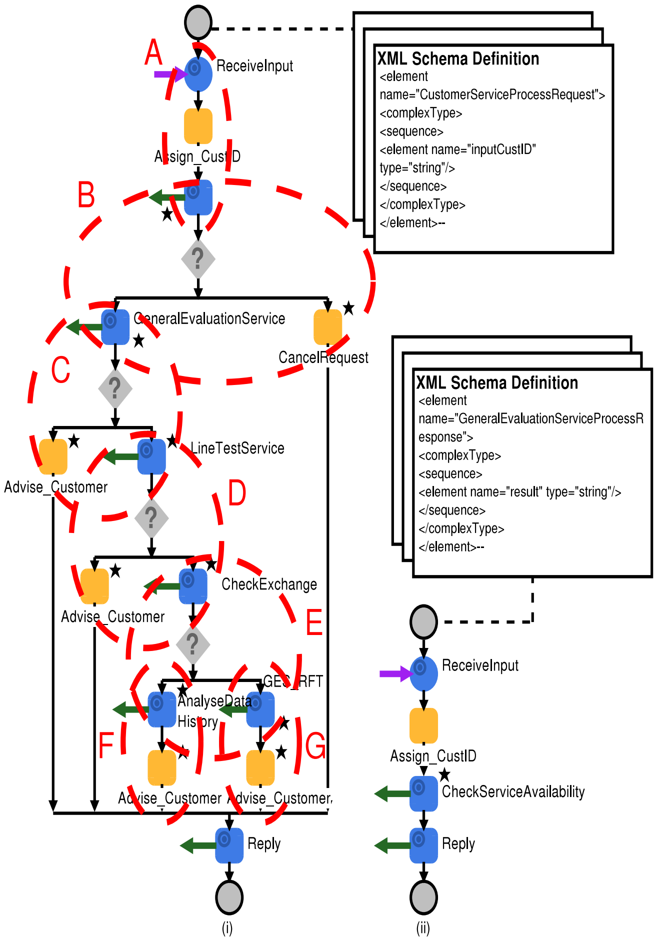
$$W = \{x : x \in \Sigma^+\}^+, \qquad (2)$$

e.g. $W = \{abcdef, abcdef, abcdeg, \ldots\}$.

2.3 Discrete Event System

A Discrete Event System (DES) is a ‘discrete-state, event-driven system whose state depends on the occurrence of asynchronous discrete events over time’ [11]. DES uses models to curb complexity. We can define a general model of a DES as a tuple $G = (X, \Sigma, \delta, x_0, A, L)$, where

- X is a set of states,
- Σ is a set of events,
- $\delta \subseteq X \times \Sigma \times X$ is a set of transitions between states,
- $x_0 \subseteq X$ is a set of initial states.



XML Schema Definition

```
<element
name="CustomerServiceProcessRequest">
<complexType>
<sequence>
<element name="inputCustID"
type="string"/>
</sequence>
</complexType>
</element>--
```

XML Schema Definition

```
<element
name="GeneralEvaluationServiceProcessR
esponse">
<complexType>
<sequence>
<element name="result" type="string"/>
</sequence>
</complexType>
</element>--
```

Figure 2: Customer Service BPEL, with some process structures highlighted (sequences A, F, G, XOR splits B, C, D, E).

- A is an alphabet of event labels, with labelling function $L : \Sigma \rightarrow A$.

Such a model can be realised using various modelling languages including automata, Petri nets, Workflow graphs models [12], [13] or ad hoc graphical representations. We do not give details here.

2.3.1 Observable and Un-Observable Events

Events $\tau \in \Sigma$ are partially observable (an event is either observable or unobservable), i.e. $\Sigma = \Sigma_O \cup \Sigma_{UO}$ where $\Sigma_O \cap \Sigma_{UO} = \emptyset$.

Some unobservable events indicate failure, i.e.

$$\Sigma_f \subseteq \Sigma_{UO}$$

We do not concern ourselves with observable failure events, which can be handled trivially, without need for diagnosers. There may be different types of failure, i.e. Σ_f is partitioned

$$\Sigma_f = \Sigma_{f_1} \cup \Sigma_{f_2}, \ldots, \Sigma_{f_n},$$

such that $\Sigma_{f_i} \cap \Sigma_{f_j} = \emptyset, 1 \leq i < j \leq n$.

Let string $\sigma = \tau_1\tau_2\tau_3'\tau_4\tau_5\tau_6', \ldots$ represent a sequence of events generated by G. Of these events only a subset are observable, e.g. τ_3', τ_6'.

Definition 1 (Projection to Observable Events)**.** We define a mapping P to project sequences to just the events which are observed,

$$P : \Sigma \rightarrow \Sigma_O \cup \{\epsilon\} \text{ such that} \tag{3}$$

$$P(\alpha) = \begin{cases} \epsilon & \text{if } \alpha \notin \Sigma_0, \text{ i.e. } \alpha \text{ is not observable,} \\ \alpha & \text{otherwise, where} \end{cases}$$

ϵ is the identity of the alphabet, i.e. $\alpha\epsilon = \epsilon\alpha = \alpha, \alpha \in \Sigma_O$.

Definition 2 (Extend P to Sequences of Events)**.** Let

$$P : \Sigma^* \rightarrow (\Sigma_0 \cup \{\epsilon\})^*, \text{ where}$$
$$P(\alpha_0\alpha_1 \ldots \alpha_n) = P(\alpha_0)P(\alpha_1) \ldots P(\alpha_n). \tag{4}$$

For example, $P(\tau_1\tau_2\tau_3'\tau_4\tau_5\tau_6') = \tau_3'\tau_6'$.

2.4 Cloud and Introspection of Virtual Machines

Benefits of moving to Cloud are well publicized; adopting could result in lower cost of IT due to the economics of scale, reduce the up-front cost for infrastructure, decrease the time to market by using off-the-shelf components, and boost the ‘Green’ credentials of the company [14]. However, in order for the Cloud environment to be profitable, there is temptation to homogenize the applications and operating systems used. But as the Cloud becomes more homogeneous, it will provide bigger and richer targets for attackers; places where the attacker may be confident of finding lucrative information or where disruption will have the greatest impact. As a result, ensuring security of the cloud is seen as a major engineering challenge [15]. In the next two subsections we shall give a brief description of two of the technologies used in Cloud.

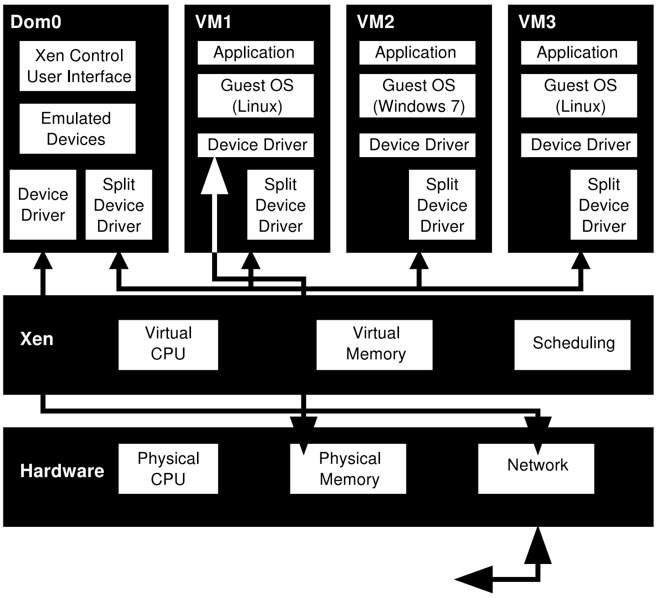


Figure 3: Virtual Machine Architecture

2.5 Virtualisation

Virtualisation is ‘A framework or methodology of dividing the resources of a computer hardware into multiple execution environments…’ [15]. Virtualisation relies on Virtual Machines (VMs), software that emulates or simulates the capabilities of the hardware. It is capable of running a complete operating system along with any applications that run on top of that OS [16]. Figure 3 depicts a high level view of Xen [15], which is an open source virtualisation software based on ‘paravirtualization’ technology. In this architecture, the Virtual Machine Monitor (VMM) is an abstraction of the underlying physical hardware and provides hardware access for the different virtual machines. Xen includes a special VM called Domain 0 (Dom0). Only Domain 0 can access the control interface of the VMM, through which other VMs can be created, destroyed, and managed. This powerful VM is used to create other Virtual Machines that can access the hardware through secure interfaces provided by Xen. In addition it is possible to create other virtual machines that can access the physical resources provided by Domain 0’a s control and management interface in Xen. Virtual Machines are heavily used within the Cloud. In addition to the advantage of running multiple operating systems simultaneously, Virtualisation reduces the cost of infrastructure implementation and the associated cost of maintenance by optimising the utilisation of resources. A user can ask for new VMs when extra resources are required and decommission some of the VMs, when they are no longer required. Virtualisation also makes it possible to secure the VMs by a powerful technique, which is commonly known as Virtual Machine Introspection.

2.6 Virtual Machine Introspection

Virtual Machine Introspection (VMI) can be defined as a virtualisation based technique that enables one guest VM to monitor, analyse and modify the state of another guest VM by observing its virtual memory pages. Such introspection can be carried out by a VMM that hosts the VM or another VM which has been granted special privileges by the VMM. VMI will allow product developers and researchers to move the security related software out of a probable target host or VM and take advantage of the host’s lack of awareness to detect any malicious events or code that is being executed in runtime. One of the early methods of introspecting a Virtual Machine from an external VM is by Garfinkel and Rosenblum [14]. They used VMI to develop an Intrusion Detection System (IDS), called Livewire, for a customized version of VMWare Workstation for Linux. VMI techniques have also been used in Digital Forensics [17] and [18]. Hyperspector [19] implemented another Intrusion Detection System for distributed computer systems using VMI to isolate the IDS from the servers that they monitor. These isolated IDSs are located inside distinct VMs which are termed as IDS VM. There are also commercial products built using VMI technology [20].

3 PROBLEM STATEMENT

As discussed in the introduction, we see a proliferation of online public services provided over the internet. These services are provided by multiple suppliers, whose information systems interact through standards-based ‘services’ (e.g. SOAP, HTTP). At the same time, the information systems providing these public services are evolving towards Cloud infrastructures comprised for example of many homogeneous commodity servers with standardised operating systems, applications and hardware. This allows computing services to be provided as a utility, with virtualised hardware and applications, and the consumer of services unaware of how or where they are hosted.

At the same time, people, corporations and governments are more dependent on these services. So the cost of service failure is high, while at the same time risks are multiplied. Heterogeneity of systems mean a system problem or successful attack can have rapid, widespread effect, while pooled resources increase the attractiveness of targets. However the complexity and interconnectedness of systems means they are impossible for humans to diagnose.

The problem we face is how to detect, diagnose and predict problems in such system architectures. We discuss this under three headings. Firstly we look at model-based online fault diagnosis. Next we discuss using Process mining techniques where models are not available, and finally we discuss some results in prediction of problems in cloud-based systems.

4 MODEL-BASED DIAGNOSIS OF FAILURE

Models representing Internet-based systems are partially observable. The growing trends of using Service oriented Architecture means that services are developed and their interfaces are made available for the users. As a result, business processes models are produced that capture external behaviour of the system by accessing the interfaces of the services while the internal behaviour remains hidden from the

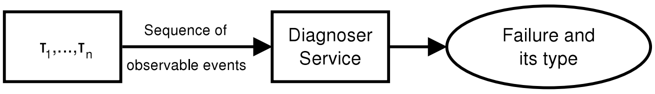


Figure 4: Diagnosis of Fault

users. As a result such models are inherently partially observable. In this context, as depicted in Fig. 4, diagnoser services (sometimes called Monitors or simply Diagnosers) are themselves services which receive sequences of observable events produced by the system and identify if a failure has happened or may have happened, in addition to the type of failure. Producing Diagnosers deals with two challenging issues:

1. is the system Diagnosable? i.e. whether it is possible to create a Diagnoser, and
2. creation of algorithms to construct Diagnosers from any given model.

The theory of Diagnosability of partial observable systems for Discrete Event Systems is well developed. Sampath *et al.* [21] in their seminal paper formulate Diagnosability and present a necessary and sufficient condition for Diagnosability. They also provide an algorithm for creating Diagnosers for Regular Languages. In their approach failure is modelled as transitions. Sampath *et al.* [21] has been extended to larger categories of models such as Petri nets [22]–[24] and even temporal logic [25], among others. The following definition from [26] extends the classic definition of diagnosability [21].

Definition 3. Consider a Petri net $\mathcal{N}$ with an initial marking M_0, which has no deadlock after firing of a transition which represents failure. We say $\mathcal{N}$ is Diagnosable if there are no two firing sequences s_1 and s_2 satisfying the following conditions:

1. $P(s_1) = P(s_2)$,
2. no failure transition appears in s_1,
3. there exists at least one failure transition in s_2
4. It is possible to make s_2 arbitrarily long after the occurrence of a fault.

The above definition states that in a diagnosable system it is not possible to come across any two execution sequences with the same observable behaviour ($P(s_1) = P(s_2)$), so that only one of them has a failure transition. The part about '… arbitrarily long after the occurrence …' is to ensure that the systems continues long enough after occurrence of failure and is also present in [21]. The classic theory of Diagnosability which was originally designed for DES has now been adopted to develop Diagnosers for Service oriented Architectures and Telecom services [27]–[33], [13]. In these approaches, a number of services are considered in a SoA, as depicted in Fig. 5. We assume that models of such systems exist and failure which is going to be diagnosed is also modelled. Then, if the system (consisting of all involved services) is Diagnosable a new service is created and *integrated* in the infrastructure to use observable events and establish occurrence and type of failure. Our recent work also uses code generation techniques to produce the Diagnosers and interfaces for integrating them into the system automatically [30]–[32].We shall explain this process with the help of an example [32].

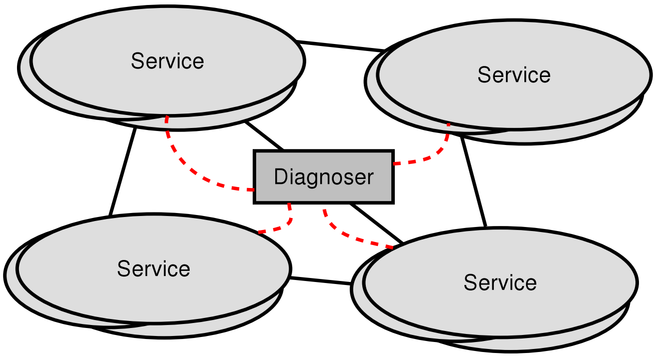


Figure 5: Diagnoser in a SoA

Example 1. Right-First-Time failure. Consider a simplified interaction between a customer and a number of services in a typical Telecommunication Company for technical support related to the Broadband connection.

As depicted in Fig. 1, the customer logs[1] onto the company website and enters details such as the account number. Choosing the 'Broadband problem' option, he submits his form online. Next, the company's Check Customer Account (CCA) service determines whether the customer account is in a satisfactory condition in order to progress the fault report. If the current status of the account is not satisfactory the customer is advised to phone the call center and the process ends. If the account status is satisfactory, the CCA invokes a request to another service called General Evaluation Services (GES). The GES examines the availability of service at the exchange side and ensures that everything is up and running, in which case the process moves to the next step. If GES identifies any problem with the availability of the services at the exchange side, the customer is informed of the status and a separate process is invoked to deal with this problem (not shown as part of this example). If everything is fine on the exchange side, the Customer Services sends a request to Line Test Service (LTS), which is an automated service to check line status up to the customer premises. However, LTS can also indicate problems on the exchange side which were not detected by the GES. There are three possible outcomes: 1) the line has no problem, move to next step, 2) the line has some problems, advise the customer or 3) There is no problem with the line, although there is likely a problem with the exchange. Option 3 is shown by the bold arrow in Fig. 1. If case 3 happens, a failure emerges which means that GES should repeat its course of action violating Right-First-Time. Finally, LTS sends a request to analyse data history in the customer router. If it is possible to carry out analysis then get a decision from the analysis algorithm (either all OK so the customer has to call technical support, or the analysis finds the problem and

[1] We assume that the Customer can log into the company's website, for example supposing the customer is not happy with the speed of his Broadband connection.

customer is advised what to do).

For the details of the method of automated production of the Diagnoser we refer the reader to [32], where four methods of integration of the Diagnoser are also described. We make use of the models of the system that represent the interaction between the involved services. Figure 2 shows the example of the model used in BPEL. We converted this model based on the formalism suggested by Vanhatalo *et al.* [34] which draws on Petri net theory so that to apply Petri net Diagnosability theory techniques. Without such a model and formulation of failure, it is not possible to design Diagnosers on the basis of this technology. In the next section, we focus on techniques which are applicable to scenarios where models of the system are not available.

5 MONITORING OF LARGE SYSTEMS VIA LOGS USING PROCESS MINING

In the previous section we discussed automatically producing Diagnosers for near identifying failure in near real time. These require a model of the system to be diagnosed. In this section, we ask what we can do if there is no model, for instance if the services are built on legacy systems or is too complex or poorly understood to model. In this case we first need to find a model of the system to which we can apply Diagnosers. For this we use Process Mining.

Figure 1 showed a simplified problem resolution process from telecoms, implemented using a SoA. In a more complex example, this might be spread across several service providers (business entities). Each part of the process will involve information systems, so events pertaining to the business processes (e.g. Fig. 2) may be recorded in multiple event logs E_i. Assuming event logs defined as in (1), the E_i can be easily merged and traces extracted into a single workflow log W. This log contains full process traces from start to end activity, e.g. from the customer loggin in to the website, to resolution of the problem.

Process mining [9], [10] is the discovery and analysis of models of business processes from workflow logs. A process discovery algorithm Φ uses a minimal log such as W to attempt to recover a model M of the 'control flow' of the underlying process, such as that in Fig. 2, i.e.

$$\Phi(W) \to M \tag{5}$$

The recovered model M represents the 'true' business process and can be compared with an 'assumed process' M' (Fig. 6), used to troubleshoot differences, check adherence to business rules, SLA and audit requirements. Mined model M can be extended (e.g. with performance information) and used for performance analysis and identifying bottlenecks. M may also be used for planning, e.g. of business change, load balancing or energy efficiency by using as a basis for modifications, simulating the changed model. Models showing the interactions between people or organisations can be used to analyse the efficiency of work practices.

Many algorithms have been proposed for the control-flow discovery aspect of process mining. These start from different theoretical bases, or focus on different priorities. We refer the interested reader to references in [9], [10], [35] for further details of algorithms. Business processes are often characterised by structuredness and concurrency: process models are (ideally) composed of substructures such as sequences and matching splits and joins, and activities or parts of the process may take place in parallel.

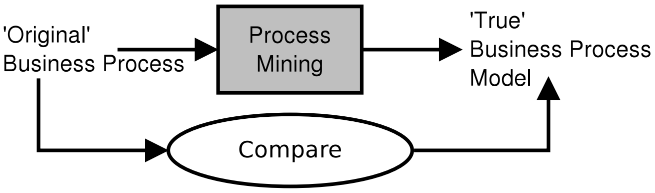


Figure 6: Process Mining

As a simplified example of a process discovery algorithm we outline the Alpha algorithm [36]. Consider two events a and b from the set of activities A belonging to a process M recorded in a workflow log W. These two events must be related in one of four relations, defined as follows.

- $a \to b$ (a may appear immediately before b in traces in W, never b before a), or conversely
- $b \to a$,
- $a \parallel b$ (sometimes a appears immediately before b, sometimes immediately after),
- $a \# b$ (a and b are always separated by at least one other activity).

The algorithm processes workflow log W to determine the relation between each pair of activities $(a, b) \in A \times A$. From this set of relations compiled for each pair of activities, a Petri net is created that satisfies all these relations. Note that this assumes that events are always recorded correctly in W.

One key question that arises is, if a is seen before b thousands of times and b before a only once, should this be interpreted as a mistake in the log or a rare scenario? In general this question can only be answered with knowledge of the business environment or service, and different algorithms make different assumptions.

So using a process mining algorithm such as Alpha we can discover process models as a basis for the diagnosis techniques described in the previous section. However,workflow logs can be large, and processing them can be computationally expensive (or data can be expensive or time-consuming to collect). Can we minimise the amount of data we need to use? How many process traces do we need to be confident that the model we have mined is the correct one? We next look at these questions.

5.1 Real Time Business Process Mining (RTBPM)

In this section we outline a probabilistic framework for considering process mining questions (for a fuller presentation see [35]). This provides a rigorous basis for answering questions such as 'how many traces do we need to be confident in the results of mining?', 'how different are two models?', and 'what is the probability that a detected fault is real and not an artefact of the data?'.

We here describe using this framework to determine the probability of identifying an undesirable scenario. Given a workflow log W, what is the probability P_f of identifying a failure or undesirable scenario, if we only use $X\%$ of the log? Conversely, given a desired P_f, can we calculate X?

To answer such questions involving uncertainty, we first need a probabilistic framework within which to consider business processes and process mining. Whereas business processes have traditionally been viewed as languages over activities, with no probabilistic structure, we consider business processes as probability distributions over strings of activities ('traces'). The primary task of a process discovery algorithm is to learn these distributions.

As introduced in section 2.2 we represent activities as symbols from a finite alphabet Σ, and traces as strings $x \in \Sigma^+$. We assume a probabilistic model for the generation of event traces, i.e. that traces are drawn into the event log *i.i.d.* (independently and identically drawn). The *true* business process M is modelled by a probability distribution P_M over traces, where the probability of a trace x is $P_M(x)$, such that $\sum_{x \in \Sigma^+} P_M(x) = 1$. As before, the workflow log W is a finite multiset over Σ^+, now understood to be drawn *i.i.d.* from P_M. The task of a process mining algorithm is to learn from W a distribution $P_{M'}$, to approximate P_M.

We are now assuming that we have a correct process model M of the system, e.g. previously mined from a 'large' log. We want to use process mining to monitor the system for failure, so the question becomes how many traces n do we need to use from the log W to be confident in mining M correctly? If we answer this question, we can be confident that if we use n traces and mine a significantly different model M', then the underlying process changed and we may have a fault scenario.

Since processes are distributions over traces, we use distances between distributions, such as the Euclidean distance, to test for significant difference between models, e.g.

$$d_2(P_M, P_{M'}) = \sqrt{\sum_x \left(P_M(x) - P_{M'}(x)\right)^2} > \epsilon, \qquad (6)$$

for small $0 < \epsilon \ll 1$

We use the Alpha algorithm [36] as an example. First we consider the basic substructures from which business processes are constructed, highlighted for example in Fig. 2. For acyclic processes, Alpha can discover sequences of activities, exclusive (XOR) splits (to alternative sequences of activities) and parallel (AND) splits (to parts of the process that may execute concurrently) and the corresponding join structures. Next we analyse the probabilistic behaviour of the algorithm to produce formulae for the probability of successful mining of these substructures, in terms of the probabilities in the model and n, the number of traces used for mining. These probabilities for discovery of structures can be combined to give the probability of successful mining by Alpha of the whole model M.

The discovery of structures in the model can be treated as conditional on the discovery of 'earlier' structures in the model, so if M is the example model in Fig. 2, then

$$P_\alpha(M) = P_\alpha(A) \times P_\alpha(B|A) \times P_\alpha(C|B) \times \ldots, \qquad (7)$$

where $P_\alpha(S)$ is the probability of Alpha correctly mining structure S, $P_\alpha(M)$ the probability of mining the full model. These probabilities are given in terms of n (the number of traces in the workflow log used for mining) and probabilities of substrings in the log.

To obtain the number of traces n needed to ensure that with confidence P_c the algorithm will produce the correct model, we invert the equation and fix a desired confidence in the mining results, $P_\alpha(M) = P_c$. Thus when a model is mined from a log of n traces, if the distance between the true and mined models $d(M, M') > \epsilon$ (equation 6), then with probability $P_f = P_c$ we have identified a fault.

The Alpha algorithm is relatively simple and makes many assumptions, e.g. no noise in the recording of the traces, and that the underlying process can be modelled by a restricted Petri net (Structured Workflow Net). However the same method can in principle be applied to any process mining algorithm.

6 MONITORING EMERGING MALICIOUS BEHAVIOUR

Having discussed using model-based Diagnosers to identify known faults, and process mining to learn unknown business process models from logs, in this section we ask whether we can diagnose a new fault which we have not seen before. This seems impossible in general. However it is possible in some cases to discover failure which is associated to emerging behaviour which has not seen before. In this section we give an example of such failure detection technology. The proposed method can be compared to the use of symptoms in human pathology, in which study of symptoms directs physicians to diagnosis of a disease or possible causes of illness. Observing unusual symptoms, even a physician cannot identify the illness, he will be alerted to conduct further experiments or to ask for expert advice. In that sense, from the observation of unusual symptoms the possibility of illness is discovered. In this section we argue that modern malware is becoming component wise. We also argue that in an environment such as Cloud in which introspection is possible, components used in malware produce symptoms. As a result, similar to pathology, observing of the symptoms can lead to discovery of possible malicious behaviour which can be new malware, or malware created from components used in old malware.

6.1 Reuse of Components and Techniques in Modern Malware

A malware writer must overcome a large number of obstacles to reach his objective. Among them, there are problems related to how to gain entry to a machine, how to install malicious code, how to evade detection, how to prevent the infected machine informing the owner, how to propagate, how to make analysis difficult, how to stop other malware writers to gain access to an infected machine and so on. Considering the sophisticated nature of modern defence, solving all these problems demands huge resources. In addition, a low quality malware might 'give the game away' resulting in alerting security experts of the vulnerabilities of the target system. As a result, malware developers reuse the existing components, algorithms and techniques to improve the quality of the code. Some of the reuse is of legitimate components, for example using existing encryption libraries, and some are illegal software available online [37]. Consequently, it is common to come across variants of the same script within various malware products [38].

6.2 Symptoms That Point to Malicious Activities

Reusing code or techniques can leave symptoms behind. For example, a wide range of malware disables the defences of the system by stopping the antivirus software. Conficker [39] for example is a well-known computer worm that targets the Microsoft Windows operating system and forms a botnet. in the *Conficker C*, 23 processes are immediately aborted whenever they are discovered running on the victim host, including *sysclean*, *tcpview*, *wireshark*, *confik* and *autorun*. See page 12 of [39] for a list. Absence of Antivirus software from the Process Table of a system can be seen as a symptom that points to the possibility of malicious behaviour. Of course, it is possible that the Antivirus has been stopped for various legitimate reasons. Other examples of symptoms are unusual values for registry keys, or existence of high entropy code associated with encryption, which is essential for the malware when communicating with the malware writer. For a list of symptoms see [40], where we have included a list of symptoms which we have come across when studying well known malware.

The key point is that appearance of the symptoms can be a reason for further investigation. In particular, observing more than one symptom can convince us of the greater possibility of an undesirable behaviour. This is similar to the patient who is suffering from a disease which has caused multiple symptoms. Shifting the attention to looking for the symptoms, as opposed to looking for the malware that creates the symptoms, can alert us of existence of malicious behaviour.

6.3 FVMs and Monitoring of Malware While Remaining Hidden

To cope with sophisticated defence mechanisms deployed in modern systems, malware writers have developed techniques to remain hidden. For example, a common practice is to stop an infected system from contacting security vendors such as antivirus providers. In some extreme cases, malware writers can completely incapacitate the system by conducting aggressive actions such as killing the operating system to cover their tracks [37]. However, it is very difficult to remain invisible to someone viewing from 'outside' when VMI is used. Relying on VMI, the external viewer can observe the changing state of a VMs memory, processes that take inordinately long times to initialize, snippets of program code that has been obfuscated, snippets of code containing known crypto algorithms, or any modifications to the system code. Figure 7 depicts the outline of the approach suggested in this paper. It shows a number of small independent VMs, called Forensic Virtual Machines (FVMs), which have been given the capability to inspect the memory pages of specific Customer Virtual Machines. Once a symptom has been detected, then the FVM reports its findings to other FVMs via secure multicast. In such cases, other FVMs will be prompted to inspect the VM for additional symptoms. In addition, when a symptom is discovered, this fact is reported, via Dom0, to a Command & Control centre. The Command and Control Centre correlates this information with information from other sources to identify an appropriate mitigation. For instance, the Command & Control, through the Dom0 and hypervisor, can 'freeze' the customer's VM by denying it any CPU cycles as a result to stop the malicious activity. The memory will remain frozen until it can be forensically examined or copied for further analysis.

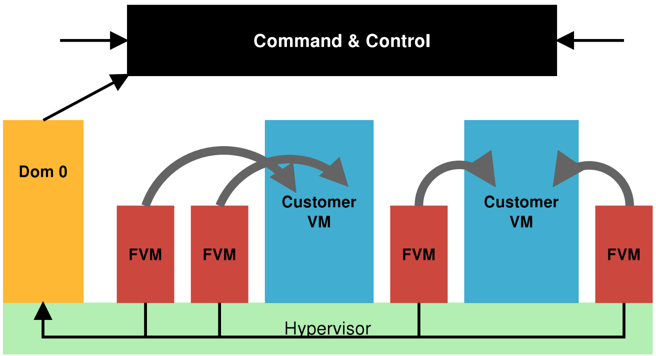


Figure 7: Forensic Virtual Machine Architecture

FVMs make use of the computational resources that could otherwise be allotted to the customers VMs. As a result, management of the efficient allocation of the resources to the FVMs is crucial. In particular, creating and deploying an FVM is computationally intensive. In addition, permanent monitoring of an FVM is costly and wasteful, as the symptoms are expected to appear sparsely. We have designed the FVMs so that they regularly change their target Customer VM. To achieve this, a distributed algorithm is created to allow the FVM to schedule moving its searching process from one Custome's VM to another. We refer to such algorithms as mobility algorithms. For an example of a mobility algorithm see [40].

6.4 Limitations

The proposed approach has a number of limitations. Firstly, the suggested approach cannot cope with malware products which do not make use of component or existing algorithms. This although it seems unlikely is not impossible. Secondly, compromising Dom0 will allow taking over the virtualisation layer. To the best of our knowledge this has not happened yet. Securing the virtualisation layer is the subject of extensive research and technical innovations and will possibly define the battleground between malware writers who focus on Cloud.

Thirdly, it is possible to detect if a system is running on a virtualised environment. This would alert malware writers who wish to remain undetected to stay away from Cloud and focus on systems which are not virtualised.

7 CONCLUSION

In this paper we argue that the problem of ensuring correct functioning of modern systems is essential, due to their ubiquity and involvement in every area of modern life. We presented three examples of how we can approach these problems. Firstly, when we have a model of the system to be diagnosed, and secondly using logs to produce such a model when one does not already exist. Finally we discussed the situation when we are interested in emerging behaviour, such as detecting new malware threats in the Cloud, from the symptoms they present.

These examples all deal with very large and complex problems, where the size, complexity and amount of computation involved means it is not possible to manually avoid or even detect any the failures in the above categories. Therefore we have no no choice but to use computational resources to deal with the problems. As a result we are 'fighting fire with fire', using modern, distributed computing techniques to deal with faults caused within modern, highly distributed computer based systems.

REFERENCES

[1] W. H. Dutton, and G. Blank, "Next Generation Users: The Internet in Britain," Oxford Internet Survey, Oxford Internet Institute, University of Oxford (2011).

[2] R. B. Prasad, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," In Jinhwa Kim, D. Delen, Jinsoo Park, F. Ko, Chen Rui, Jong Hyung Lee, Wang Jian, and Gang Kou, editors, *NCM*, IEEE Computer Society, pp. 44–51 (2009).

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, Gunho Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communications of the ACM, Vol. 53, No. 4, pp. 50–58 (2010).

[4] E. Byres and J. Lowe, "Myths and facts behind cyber security risks for industrial control systems," Engineering Technology, Vol. 7, No. 10, pp. 48–50 (2004-2005).

[5] P. Mohajerin Esfahani, M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson, "Cyber attack in a two-area power system: Impact identification using reachability," Proceedings of the 2010 American Control Conference (ACC 2010), pp. 962–967 (2010).

[6] J. P. Farwell, and R. Rohozinski, "Stuxnet and the Future of Cyber War," Survival, Vol. 53, No. 1, pp. 23–40 (2011).

[7] R. Chinnici, J. Moreau, A. Ryman, and S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0," `http://www.w3.org/TR/wsdl20/`, W3C (2006).

[8] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, "XML Schema Part 1: Structures," `http://www.w3.org/TR/xmlschema-1/`, W3C (2004).

[9] W. M. P. van der Aalst, and A. J. M. M. Weijters, "Process Mining: a Research Agenda," Computers in Industry, Vol. 53, No. 3, pp. 231–244 (2004).

[10] A. Tiwari, C. J. Turner, and B. Majeed, "A Review of Business Process Mining: State-of-the-Art and Future Trends," Business Process Management Journa, Vol. 14, No. 1, pp. 5–22 (2008).

[11] P. J. Ramadge and W. M. Wonham, "Modular Supervisory Control of Discrete Event Systems," Proceedings of the 7th International Conference on Analysis and Optimization of Systems, pp.202–214 (1986).

[12] J. Vanhatalo, H. Völzer, and F. Leymann, "Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition," In B. J. Krämer, Kwei-Jay Lin, and P. Narasimhan, editors, *ICSOC*, volume 4749 of Lecture Notes in Computer Science, pp. 43–55 (2007).

[13] M. Alodib, and B. Bordbar, "A Modelling Approach to Service Oriented Architecture for On-line Diagnosis," Service Oriented Computing and Applications, pp. 1–17 (2012).

[14] T. Garfinkel and M. Rosenblum, "A Virtual Machine Introspection Based Architecture for Intrusion Detection," Proceedings of the 2003 Network and Distributed Systems Security Symposium (NDSS), pp.191–206 (2003).

[15] D. E. Williams, and J. R. García, "Virtualization With Xen: Including XenEnterprise, XenServer, and XenExpress," Syngress Media, Syngress (2007).

[16] R. P. Goldberg, "Survey of Virtual Machine Research," Computer, Vol. 7, pp. 34–45 (1974).

[17] K. L. Nance, B. Hay, and M. Bishop, "Investigating the Implications of Virtual Machine Introspection for Digital Forensics," Proceedings of the International Conference on Availability, Reliability and Security 2009 (ARES'09), pp. 1024–1029 (2009).

[18] B. Dolan-Gavitt, B. D. Payne, and W. Lee, "Leveraging Forensic Tools for Virtual Machine Introspection," Technical Report GT-CS-11-05, Georgia Institute of Technology, `http://www.bryanpayne.org/storage/GT-CS-11-05.pdf` (2011).

[19] K. Kourai, and S. Chiba, "HyperSpector: virtual distributed monitoring environments for secure intrusion detection," Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments (VEE'05), pp. 197–207 (2005).

[20] Hypertection. "Hypervisor-Based Antivirus," in Hypertection Team. Web, `www.hypertection.com` (accessed 13/09/2011).

[21] M. Sampath, R. Sengupta, and S Lafortune, "Diagnosability of Discrete-Event Systems," IEEE Transactions on Automatic Control, Vol. 40, pp. 1555–1575 (1995).

[22] S. Genc and S. Lafortune, "Distributed Diagnosis of Discrete-Event Systems Using Petri Nets," Proceedings of the 24th International Conference on Applications and Theory of Petri Nets (ICATPN 2003), pp. 316–336 (2003).

[23] G. Jiroveanu, R. Boel and, and B. Bordbar, "On-line Monitoring of Large Petri Net Models Under Partial Observation," Discrete Event Dynamic Systems, Vol.18, Issue 3, pp.323–354 (2008).

[24] A. Giua and C. Seatzu, "Observability of Place/Transition Nets," IEEE Transactions on Automatic Control, Vol. 47, No. 9, pp. 1424–1437 (2002).

[25] S. Jiang, and R. Kumar, "Failure Diagnosis of Discrete-Event Systems With Linear-Time Temporal Logic Specifications," IEEE Transactions on Automatic Control, Vol. 49, No. 6, pp. 934–945 (2004).

[26] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of Bounded Petri Nets," Proceedings of the 48th IEEE Conference on Decision and Control 2009 (CDC2009), pp. 1254–1260 (2009).

[27] Y. Wang, T. Kelly, and S. Lafortune, "Discrete Control for Safe Execution of IT Automation Workflows," Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007 (EuroSys'07), pp. 305–314 (2007).

[28] Yuhong Yan and P. Dague, "Modeling and Diagnosing Orchestrated Web Service Processes," Proceedings of the IEEE International Conference on Web Services 2007, Vol. 9, pp. 51–59 (2007).

[29] W. Hamscher, L. Console, and J. de Kleer, editors, "Readings in Model-Based Diagnosis," Morgan Kaufmann Publishers Inc. (1992).

[30] M. Alodib, B. Bordbar, and B. Majeed, "A Model Driven Approach to the Design and Implementing of Fault Tolerant Service Oriented Architectures," Proceedings of the 3rd International Conference on Digital Information Management (ICDIM2008), pp.464–469 (2008).

[31] M. Alodib and B. Bordbar, "A Model Driven Architecture Approach to Fault Tolerance in Service Oriented Architectures, a Performance Study," Proceedings of the 3rd International Workshop on Modeling, Design, and Analysis for Service-oriented Architectures (MDA4SOA), pp293–300 (2008).

[32] M. Alodib and B. Bordbar, "A Model-Based Approach to Fault Diagnosis in Service Oriented Architectures," In R. Eshuis, P. W. P. J. Grefen, and G. A. Papadopoulos, editors, Proceedings of the 7th IEEE European Conference on Web Services (ECOWS'09), pp. 129–138 (2009).

[33] Yuhong Yan, Y. Pencole, M.-O. Cordier, and A. Grastien, "Monitoring Web Service Networks in a Model-based Approach," Proceedings of the 3rd IEEE European Conference on Web Services (ECOWS2005) (2005).

[34] J. Vanhatalo, H. Völzer, and F. Leymann, "Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition," Proceedings of the 5th International Conference on Service-Oriented Computing (ICSOC'07), Springer-Verlag, pp. 43–55 (2007).

[35] P. Weber, B. Bordbar, and P. Tiňo, "A Framework for the Analysis of ProcessMining Algorithms," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 43, Issue 2, pp. 303–317 (2013).

[36] W. M. P. van der Aalst, T. Weijters, and L. Maruster, "Workflow Mining: Discovering Process Models from Event Logs," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 9, pp. 1128–1142 (2004).

[37] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. M. Youssef, M. Debbabi, and L. Wang, "On the Analysis of the Zeus Botnet Crimeware Toolkit," Proceedings of the 8th Annual International Conference on Privacy Security and Trust, pp. 31–38 (2010).

[38] Sheng Yu, Shijie Zhou, Leyuan Liu, Rui Yang, and Jiaqing Luo, "Malware Variants Identification Based on Byte Frequency," Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC2010), Vol. 2, pp. 32–35 (2010).

[39] P. Porras, H. Saidi, and V. Yegneswaran, "Conficker C analysis," SRI International (2009).

[40] K. Harrison, B. Bordbar, S. T. T. Ali, C. Dalton, , and A. Norman, "A Framework for Detecting Malware in Cloud by Identifying Symptoms," Proceedings of the 16th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp.164-172 (2012).

(Received October 13, 2012)

Behzad Bordbar Behzad Bordbar has his BSc, MSc and Ph.D in Mathematics (PhD from Sheffield, UK). Following his PhD, he worked as a researcher on a number of projects at University of Ghent, Belgium and University of Kent, UK. He is currently affiliated to the School of Computer Science, University of Birmingham, UK, where he teaches courses in Software Engineering and Distributed Systems. In recent years, he has had close collaborative research with various academic and industrial organizations, among them Ghent University, Osaka University, Colorado State University, BT, IBM and HP research laboratories. His research activities are mostly aimed at using modelling to produce more dependable software and systems in shorter development cycles and at a lower cost. His current research projects are dealing with Formal methods, Model Analysis, Software Tools, Model Driven Development and Fault-tolerance in Service Oriented Architectures and Cloud.

Philip Weber Philip Weber received the BSc degree in Computer Science from Loughborough University, UK, in 1994, and the MSc in Advanced Computer Science from Birmingham University, UK, in 2009. Between these he worked in industry designing, analysing and implementing IT systems, and in systems administration. He is currently working towards the Ph.D. degree in Computer Science at the University of Birmingham, UK. His research interests include Process Mining, Machine Learning, Data Mining and information management.

Submission Guidance

About IJIS

International Journal of Informatics Society (ISSN 1883-4566) is published in one volume of three issues a year. One should be a member of Informatics Society for the submission of the article at least. A submission article is reviewed at least two reviewer. The online version of the journal is available at the following site: http://www.infsoc.org.

Aims and Scope of Informatics Society

The evolution of informatics heralds a new information society. It provides more convenience to our life. Informatics and technologies have been integrated by various fields. For example, mathematics, linguistics, logics, engineering, and new fields will join it. Especially, we are continuing to maintain an awareness of informatics and communication convergence. Informatics Society is the organization that tries to develop informatics and technologies with this convergence. International Journal of Informatics Society (IJIS) is the journal of Informatics Society.

Areas of interest include, but are not limited to:

Computer supported cooperative work and groupware
Intelligent transport system
Distributed Computing
Multi-media communication
Information systems
Mobile computing
Ubiquitous computing

Instruction to Authors

For detailed instructions please refer to the Authors Corner on our Web site, http://www.infsoc.org/.

Submission of manuscripts: There is no limitation of page count as full papers, each of which will be subject to a full review process. An electronic, PDF-based submission of papers is mandatory. Download and use the LaTeX2e or Microsoft Word sample IJIS formats.

http://www.infsoc.org/IJIS-Format.pdf

LaTeX2e

LaTeX2e files (ZIP) http://www.infsoc.org/template_IJIS.zip

Microsoft Word™

Sample document http://www.infsoc.org/sample_IJIS.doc

Please send the PDF file of your paper to secretariat@infsoc.org with the following information:

Title, Author: Name (Affiliation), Name (Affiliation), Corresponding Author. Address, Tel, Fax, E-mail:

Copyright

For all copying, reprint, or republication permission, write to: Copyrights and Permissions Department, Informatics Society, secretariat@infsoc.org.

Publisher

Address: Informatics Laboratory, 3-41 Tsujimachi, Kitaku, Nagoya 462-0032, Japan
E-mail: secretariat@infsoc.org

CONTENTS