Development of Digital Communication Assistant Application for Autistic Children
Ippei Torii, Kaoruko Ohtani, Takahito Niwa, and Naohiro Ishii

Micro Area Selection Framework for ICT Infrastructure Diffusion based on Commuting Flow
Motoi Iwashita, Takeshi Kurosawa, Akiya Inoue, and Ken Nishimatsu

Automatic Generation of SVG Program Documents with Animation Based on Attribute Graph Grammars
Takaaki Goto, Tomoo Sumida, Tadaaki Kirishima, Tetsuro Nishino, Takeo Yaku, and Kensei Tsuchida

Analysis of Software Fault Prediction Models Using Machine Learning Techniques
Golnoush Abaeia and Ali Selamata

Practice and Evaluation of Open Source Software Development Education
Based on the UEC Software Repository
Takaaki Goto, Kensei Tsuchida, and Tetsuro Nishino
Development of Digital Communication Assistant Application for Autistic Children

Ippei Torii\textsuperscript{a}, Kaoruko Ohtani\textsuperscript{a}, Takahito Niwa\textsuperscript{a}, Naohiro Ishii\textsuperscript{a}
Aichi Institute of Technology, Japan

Abstract

We developed a new AAC (Augmentative and Alternative Communication) application for personal digital assistant for autistic children, “Let’s Talk!”’. This new and remarkable application has many particular advantages comparing to existing AAC. We especially focused on an easy and simple manipulation. By tapping a symbol on a screen of a PDA (Personal Digital Assistant) with this application, a user can show his/her thoughts with pictures and sounds to others easily. There are two modes in the application that can be switched depending on different situations of users. It has 120 symbols based on daily life and a user can also create the original page with new icons made by pictures or sound. A user also can customize an original page by arranging icons he/she made or existing symbols. On the newest version of this application, we added Task Schedule System to stimulate motivations of children to do something by them. On the last part of this study, we show some case studies. We introduced this application to students in a school for handicapped children and corrected data.

Keywords: Autism mental retardation communication, VOCA (Voice Output Communication Aid), AAC (Augmentative and Alternative Communication)

1. Introduction

In the modern society that computer technology has developed, the variations and quantities of information have been increased explosively. In such situations, the visualization technology that tells the specific characteristics of data with visual expression by using 3DCG is regarded as important. Because of the popularization of mobile data terminals, it is required to develop a high-performance visualization application. We are proposing a method for observation, visualization and application of information being based on those social back ground. We are trying to fit the visualization of information method to real subjects. We study how the method can be put to practical use, find the problems, and apply it to the society. Especially we aim to develop a support application for PDA for handicapped people.

Many autistic children tend to have disability of verbal communication, so they need some support tools to express their thoughts or needs. Some assistant applications for PDA (Personal Digital Assistant), such as Drop Talks\cite{1}, Voice4u\cite{2}, Tap to Talk\cite{3}, aimed to help autistic children who have communication disorder have already introduced. Those communication assistant tools are called AAC (Augmentative and Alternative Communication) or VOCA (Voice Output Communication Aid). Although many studies about VOCA \cite{4}\cite{5} had been made and school educational fields have adopted these tools, they are not come into general use because of the high price and complicated operations. Therefore, we tried to develop a new communication assistant tool for PDA with simple and easy manipulation in low price as shown in Fig.1.

We spent a half-year to develop the application since October 2010 and finally released “Let’s Talk!” in April 2011 on iTunes Store. After we launched the application, we have modified and updated it 11 times within a year referring to data from schools for handicapped children or requests from users through ICT infrastructure.

2. Purpose of the study

There are some characteristics of autism as below;
1. Disorder to develop sociability
2. Disabilities to develop speech and communication ability
3. Repetition of same behavior and attachment to something

Because of the characteristic of disorder to develop sociability, it is very difficult for some autistic children to relate with others. They tend not to have ability to speech or repeat like a parrot even if they can say words. They cannot communicate with others because of these disabilities. Picture cards have been used to support communication for autistic children who have mental disorders. But those autistic children just point the card regardless of other people.

\textsuperscript{a}Department of Information Science, Aichi Institute of Technology, Toyota, Aichi, Japan
mac@aitech.ac.jp, kohani@comcast.net, x08062xx@aitech.ac.jp, ishii@aitech.ac.jp

\begin{figure}[h]
\centering
\includegraphics[width=0.3\textwidth]{Figure1.png}
\caption{A communication application tool for PDA}
\end{figure}
They do not understand they need to show the card to someone to whom they will fulfill their needs. In most cases, teachers show the cards and make the children to choose one. The autistic children did not get enough success experiences with this method, so their communication ability did not improved. In 1985, Dr. Andy Bondy and Mr. Lori Frost developed PECS (Picture Exchange Communication System)[6].

They pointed the problems of the existing picture indication system as “The system does not teach children to be interested in people but pictures.” and “The system ignores the approach for people which is a part of communication.” But PECS also has some problems. A user carries “a communication book”, which is a note with picture cards.

The more new words a user is gaining, the more picture cards he/she needs to carry. It would take much time to find a card he/she needs or to make new cards. If an autistic child uses PDA as a communication tool instead of those picture cards, he/she can reduce much time and effort to make or find cards. And it is much easier to carry around and to distribute. A user can create symbols by him/herself that are suitable his/her situation. It is a great advantage of the application to increase new vocabularies.

3. Construction of the system

3.1 Usability of the application

We focused on a simple manipulation without complicated explanation to develop this application. The reaction area is wide and the volume of the voice and sound is very clear and much enough to be able to hear outside or in a crowded place such as a classroom. These distinguished characters will let handicapped children to use those AAC more easily. They can communicate with others whenever and wherever by using pocket able PDA without carrying a special piece of equipment. It is also expected the unique contents of this application may create a chance to communicate with others. Autistic children will be accepted in the society with the application and it will lead to increase the quality of life of them.

3.2 Two modes of the application

There are two modes on this application. First, on "Supportive Mode", a supporter who helps an autistic child starts communication. Second, on "Self-use Mode", an autistic child can show his/her request by voice with this application. By switching the modes depending on different situations of users, more effective communication will be expected.

3.3 Supportive Mode

On Supportive Mode of the application, a supporter starts to pick symbols fitting to the situation and the condition of an autistic child who needs assistant for communication. A supporter can make a choice that is suitable for his/her request easily. When an autistic child touches the symbol what he/she wants to tell, a simple 2 words sentence comes out by voice sound. It is difficult for autistic children to choose what they really need to say among too many choices. So we limited the numbers of symbols picked by a supporter from 1 to 4 and make it easy to choose.

If a supporter guesses an autistic child want to eat something.
1) A supporter chooses “Eat” among 12 categories.
2) A supporter estimates what an autistic child wants to eat and pick “cereal”, "bread", "soup", "spaghetti” and shows it to a user.
3) A autistic child touches “cereal” and voice sound says “I want to eat cereal.”

3.4 Self-use Mode

If an autistic child understands how to manipulate this application, he/she can tell what he/she wants directly with icons in Self-use Mode. An autistic child may feel the joy of understanding by trying to tell his/her thoughts to others with this mode and it will turn into more effective communication. On this mode, we adopted some distinguished ideas such as the wide reacting area that makes it easy to operate for people with disabilities.

If an autistic child tries to tell he/she wants to eat something.
1) An autistic child chooses “Eat” among 12 categories.
2) An autistic child touches what he/she wants to eat ="cereal”.
3) The voice sound says, “I want to eat cereal.”

3.5 Original page

This application has about 120 icons which fit our daily life, but more icons may be required depending on autistic children’s situation. On the other hand, if there are too many icons, it will take much time to choose and avoid the easy operation. The "Make" page of this application will provide more effective communication without confusion. On this page, a user can make his/her original icons with a camera or a voice recorder the PDA has or using an illustration or sound source founded on web sites. It is sure to broaden the communication if there are some original icons which suitable with an autistic child’s dairy life, such as the places
he/she goes frequently, familiar people or the dairy activities. While keeping its simple operations, easier and quicker communication is expected by using those original icons. Original Page is customizable to put icons that are created by a user or existing symbols freely.

3.6 Stamp Mode

On the upgraded version of “Let’s Talk!”(iPhone / “Let's Talk!”AppVersionNumber:3.21 November 10, 2011), we added Stamp Mode. On this mode, a user can make a chart with a goal. Children get a stamp whenever they have done what they need to do. It is easy to understand how many stamps they need to reach a goal for autistic children. When a child achieves a goal, fireworks or fanfare will be appeared as a prize. It would be effective to give children the motivation to complete the chart to get these special prizes. Autistic children will get good influences through experiences of being able to do something by themselves repeatedly. It will develop autonomy and independence and lead motivations trying anything by their own efforts.

4. Design of the system

The main purpose for the development of this application is to make it possible for autistic children to communicate with others quickly and effectively. Therefore, it is important to figure out what they really need. We adopted ideas and opinions from the teachers of schools for handicapped children who contact with autistic children everyday.

4.1 Program constitution for design

We made several test manufactures and asked the teachers to use them in real classes with the children. In Self-use mode and Supportive mode, we do not use the font in PDA, but show the title of symbols as a picture to unify the design of the application. However, on Original Page, there are two ways to create a symbol; to put a new symbol that is made by a user, or to use an existing symbol. This system makes it possible to use two symbols that were created in different ways at the same time, but on the other hand, there will be a lack of unity of design if the title as a picture and the title as a font are mixed up together. Therefore, we adopted the method to show the letter information separately by trimming the symbol to regular size automatically, when an existing symbol is loaded on Original Page. The information of title and sound that is required when an existing symbol is loaded on Original Page will be picked up from the XML file where the information is described previously. Original Page and interaction of programming files and XML files are shown in Figure 2.

![Figure 2. Relation of original page and the XML file](image)

To load the sound file, pictures, and title form XML file (called it a plist in Integrated Development Environment), first the former picture and the name of sound file are pull out from the plist. The code is showed in Figure 2.

![Figure 3. The code of pulling out the former picture and the name of sound file form of the plist](image)

Second, a new name of the file of a symbol that is trimmed and prepared is created based on the obtained date and time. We show the code of programming in Figure 4.

![Figure 4. The code to create a new file name based on the obtained date and time](image)

We concentrated to keep good design qualities of this application. At the same time, modifying and using the pictures of Supportive mode and Self-use mode reduce the number of picture files. On iPhone App Store, an application that is more than 20MB cannot be uploaded to iPhone or iPad without connecting through PC or WiFi. The number of people who possess smart phones or high-performance music players is increased lately because of the popularity of smart mobile phone. But not all of the people who possess those high performance equipments are familiar with computer. Also it is
hard to say that the infrastructure to connect WiFi is widespread among the general public. Therefore, we considered it is important to make it possible to install applications to the devices conveniently and immediately by using 3G circuit. It is effective to keep the size of the application under 20MB by using this method. On the other hand, a subject child sometimes erased a symbol by mistake, or there were some children interested in erasing symbols in clinical trials. And the problem was found that pop-up alert was displayed repeatedly when switching the pages several times. It was happened because the reacting time of the delete button was too short. So we reset the length of the time to delete from 1 second to 3 seconds. But another problem was occurred that the liquid panel reacted to a slight movement or vibration of a finger, and it caused to block to delete. We prepared the flag variable and set the initial value to 0, and added a postscript as below:

1) When an icon is touched, set Flag to 1 and start to count the timer. The program code is shown below in Figure 5.

```c
-(void)longTouchStart:(id)sender
{
    Flag=1;
    [self performSelector:@selector(longTouch:) withObject:sender afterDelay:3.0f];
}
```

Figure 5. Code to start the timer

2) When a finger leaves from an icon, set Flag to 0 again. Flag also returns to 0 when a page moves by swiping. The program code is shown below in Figure 6.

```c
-(void)longTouchCancel:(id)sender
{
    Flag=0;
    [NSObject cancelPreviousPerformRequestsWithTarget:self
 selector:@selector(longTouch:) object:sender];
}
```

Figure 6. Code to cancel Flag

Third, the picture is trimmed and a new file is created. We show the code of programming in Figure 7.

```c
NSString* path = [NSBundle mainBundle]
    pathForResource:
    ofType:@"jpg";
UIImage* image = UIImage imageWithContentsOfFile: path;
NSString* toPath = [NSString stringWithFormat:
    DocumentsFolder, newFileName];
CGImageRef cgImage = CGImageCreateWithImageInRect
    (image.CGImage, CGRectMake(0, 0, 92, 92));
image = UIGraphicsGetImageFromCurrentImageContext();
writeToFile: toPath atomically:YES;
```

Figure 7. The code to trim and save the picture as a new file

Fourth, the voice sound file corresponding to the former picture is copied and used for a new symbol. We show the code of programming in Figure 8.

```c
NSFileManager *fileManager =
    [NSFileManager defaultManager];
path = [NSBundle mainBundle]
    pathForResource:symbol ofType:@"caf";
toPath = [NSString stringWithFormat:
    DocumentsFolder, newFileName];
NSError *error;
[fileManager copyItemAtPath:path toPath:
    toPath error:&error];
```

Figure 8. The code to copy the sound

At the same time, the file names of picture and voice sound are recorded on plist and the screen goes back to the Original page.

To use this method brings some benefits. One of the benefits is the file of Supportive mode or Self-use mode will not need to be changed. It is a big progress to make it possible to put the existing symbols to Original Page without damaging the appearance, since we concentrated to keep good design qualities of this application. At the same time, modifying and using the pictures of Supportive mode and Self-use mode reduce the number of picture files.

(3) If Flag is 1 when the count of the timer becomes 3 seconds, the alert to confirm to delete is displayed and Flag returns to 0 at the same time. The program code is shown below as Figure 9.

```c
-(void)longTouch:(id)sender
{
    if (Flag == 1){
        Flag=0;
        [self viewPopup:(id)sender];
    }
}
```

Figure 9. Code to show the alert to confirm to delete

4.2 Sound quality

“Let’s Talk!” is presumed to be used in noisy situations such as in a town or a classroom. Therefore, it is important that the sound quality is clear and loud enough in such circumstances. Many communication applications do not have enough volume or have the clipping noise that will prevent smooth communication in noisy places. Therefore, we tried one way after another to solve this problem. Usually 16 bits is suitable to record voice, but we adopted 24 bit instead. When we use 24 bits, we can enlarge or shrink the voice date in a wider range than 16 bits. The other reason to use 24 bits for recording is the sound quality does not deteriorate when we edit the voice data with an equalizer. To edit the voice data, we normalized the recorded data (Figure 10) first. Then we calculated the average level and
equalized all voice data. We show the process of normalization in Figure 11.

Many of autistic children have hyperacusis. They are over sensitive especially in a high register. Therefore, we cut a high register and tried to make the sound easy to listen for anyone. We applied an equalizer to change the frequency characteristics of normalized voice signal and cut a high register. We show this data in Figure 14 and the waveform of voice data that is used on the application in Figure 15.

The speakers of smart phones sometimes provide quite different sound compare with ordinary audio equipment. So unexpected noise will obstruct clear sound even we record a high quality voice data. Therefore, we prepared many samples of voice besides the data as stated above and modulated them. As a result, we heard many positive opinion such as, “I can listen the voice very clearly in the noisy place.

4.3 Selection of symbols

When we picked categories and symbols in this application, we tried to choose more suitable contents for autistic children by adopting opinions of teachers of schools for handicapped children. For example, there are traffic and a (ventilating) fan in “I want to watch/see” category. Those symbols proposed by the teachers. They said many autistic children tend to be attracted those things. And we have Japanese version of this application, “Nee Kite(ねえ、きいて。)”. The basic designs are the same but there are some differences between English version and Japanese one’s. For example, we use “railroad crossing” for Japanese version instead of “traffic” in “I want to watch/see” category. It is because people use train often and we have many railroad crossings around in Japan. We considered these cultural differences when we developed both English and Japanese versions.

4.5 Updating the application

After we released “Let’s Talk!”, we have collected suggestions and requests from users through ICT infrastructure and the data from schools for handicapped children. We modified and updated the application 15 times till September 2012 referring to those opinions and the data.

- 【iPhone / 「Let’s Talk!’ App Version : 1.0】 April 23, 2011
  - On sale

- 【iPhone / 「Let’s Talk!’ App Version : 1.1】 May 4, 2011
  - Correct the slow reacting buttons and category buttons / Relocate the information button / Add explanations / Change the voice / Make it return to the top page when restarts

- 【iPhone / 「Let’s Talk!’ App Version : 2.0】 May 18, 2011
  - Set up “Make” page to create original icons / Set up the functions to make new pictures / Set up the record mode and input the data of letters / Use animation for icons of categories
5. Experiments at school for handicapped children

Introducing the application on the experimental basis was carried out at Miai Yogo School, which is the school for handicapped children. This application attracted the attention of most of the children and they really enjoyed using it. They liked cartoon-like characters and were interested in the operation since it looked like a game for them. Some of the children who had never talked before started to communicate with teachers or parents with the application. It is obvious this application made great effect to those children with speech disorders. We made surveys about the reference between “Let’s Talk!” and the behavior of a child.

5.1 Case study 1

![Figure 16. The alteration of request behavior with iPod at lunch time (8-year-old boy)](image)

The alteration of request behavior with iPod at lunch time (8-year-old boy)

**<The subject of investigation>** 8-year-old boy with autism who does not have ability to speak.

This student tried to tell his teacher he wanted to have another plate at lunch with the application. His teacher divided his lunch into 12 small portions and gave one of them at once to urge him to ask a refill as a training of communication. Figure 16 shows how his behavior had changed when he started to use this application. The dark bar graph shows how many times the teacher thought him the manipulation of this application. The manipulation is divided into 5 levels.

1. To get the iPod
2. To release the lock of the iPod
3. To start “Let’s Talk!”
4. To choose an icon
5. To show it to someone

Autistic children will learn 1 to 4 quite easily, but they will not understand 5, which means they need to show the
screen to someone if they want their requests satisfied. The light bar graph shows how many times the teacher told him to get the attention of others. It is very difficult for them to understand that they need to get the attention of other people if they want to communicate. So the teacher’s support was needed more than 50% till 10th week for this boy. The Solid line graph shows how many times he actually used this application. The dashed line graph shows how many times he tried to get attention of his teacher.

Between 7 and 8 weeks, the school was closed for a summer break. He restarted to use iPod after a month and half on 8 week, so it seems he had forgot how to use the application. It is clear that his motivation to communicate with the teacher had been increased by using iPod with this application. His classroom teacher reported that he often had offensive movement such as scratching or spitting to other students since he got in the school. But those behaviors had been decreased dramatically after he started to learn how to express his thoughts to others in a proper way with this application. Lately, he even tried to start communication from him with PECS (Picture Exchange Communication System) and iPod to let others to know his requests or needs.

5.2 Case study 2

10 year-old boy with autism who does not have ability to speak
This student started to use iPad with “Let’s Talk!” when he tried to tell the teacher he want a refill, need help or reduce the quantity of the food at lunch time since the beginning of May 2011. According to his classroom teacher, he had learned how to manipulate this application quickly, and made his own icons by himself. He brings his own iPad all the time and uses it as a communication tool.

5.3 Summarization of case studies

In these two case studies, it is considered that those students were satisfied as their teachers understood what they needed, even their requirements had not been fulfilled. It is important that they know someone understands their requests and empathizes with them. Many teachers of the school said most of the children seemed to enjoy their life at school more after introducing this application.

6. Clinical observation

We also operated a clinical observation of “Let’s Talk!” with the cooperation of Miah Yogo School. The purpose of the observation is to estimate the effect of this application to an autistic child.

6.1 Background of observation

April 26, 2011 to February 8, 2012
A boy, called M in this report. Born in 2000. 11 years old when we started the observation.
A boy, called M in this report. Born in 2000. 11 years old when we started the observation.

6.2 Process of development

April, 2011
M touched iPad for the first time. We gave it to him with “Let’s Talk!” on the screen. He understood that the screen would change when he touched it immediately. And he also got the idea that there were some screens that created sound when he touched them.

May, 2011
He started to use iPod, which he preferred for convenience to carry around, for lunchtime. He became to show the words such as “Can I start?” “Reduce some of this, please.” “Can I have some more?” and so on with iPod that had some applications. He learned how to use those applications quickly when we taught him. In late May, we took pictures of lunch menu and saved them in “Let’s Talk!” as original symbols in ahead. M showed what he did not like to eat when he saw the pictures. At this point, he did not try to arouse someone’s attention.

June, 2011
He started to use iPod, which he preferred for convenience to carry around, for lunchtime. He became to show the words such as “Can I start?” “Reduce some of this, please.” “Can I have some more?” and so on with iPod that had some applications. He learned how to use those applications quickly when we taught him. In late May, we took pictures of lunch menu and saved them in “Let’s Talk!” as original symbols in ahead. M showed what he did not like to eat when he saw the pictures. At this point, he did not try to arouse someone’s attention.

September, 2011
He got used to the manipulation of the application. He even became to be able to create original symbols by himself. He added symbols on the way to go back home (a school bus or parents picking him up) by capturing images from the picture library.

November, 2011
He became to be able to use some other application besides “Let’s Talk!”. He worked on his project with confirming the process by “Task Schedule”.

February, 2012
M has been using “Let’s Talk!” less often at this point. He used his gestures to tell us simple requirements such as “Can I start to eat?” or “Reduce some of this, please.” If the person he tried to tell his request could not understand his gesture, he used pictures in iPod which he always carrying around or typed words on memo function. He seemed to choose the suitable way for the situation to communicate by him. He sometimes made voice sound to get someone’s attention that he had never done before this observation. It is a big progress he makes some actions to communicate with others from himself.

6.3 Summarization of observation
Before we stared this project, M could do something only with someone’s indication and reply only yes or no if he was asked something. We tried to introduce PECS (Picture Exchange Communication System) to him, but we could not continue the training for supporters and his level stayed on Fase 2 (it is the faze a child can request even he/she stays away from a supporter or picture cards). He could understand how to use “Let’s Talk!” and what he can do with it by himself without any special training in short period of time. He felt comfortable when someone sympathized with his emotion even if his request was not satisfied immediately while he learned the joy of communication with this application. He also became to be able to understand the situation when he needed to wait till someone finished what he/she was doing at that moment. “Let’s Talk!” was developed for people who had disabilities to communicate with others such as autistic children with mental handicap. M understood the function of communication quickly with this application and became to be able to raise his ability to communicate with others. To use PDA such as iPod or iPad makes it easy to understand the manipulation and the relation of cause and effect for mentally handicapped people because of their quick reactions. And also a person whom they try to communicate with easily understands their requests with voice sounds. We consider repeating the experiences to communicate with others stimulates the desire for communications and arouses initiative. It is obvious “Let’s Talk!” is useful to raise the communication ability.

7. Prospective study
It is urgently necessary for us to develop supporting systems for children with Limb/Trunk Dysfunction or bedridden old people who even cannot touch the screen. We consider the core categories and symbols for basic communication is established through the development of “Let’s Talk!”. We will apply the method for future studies. We are working to develop an alternative communication tool to use a blink as a source of launch or operate the application. A blink detected by the camera of iPhone will start the categories or symbols in the application, then transfer to the voice.

8. Conclusions
We developed a new communication assistant tool with PDA, “Let’s Talk!”, for autistic children. If autistic children feel the joy of communication by using this application, they will be strongly motivated to try to understand others thoughts.

On the report from the school for handicapped children introducing “Let’s Talk!” on an experimental basis, this application attracted the attention of most of the children and they really enjoyed using it. Some of the children who had never talked before started to communicate with teachers or parents with the application. Especially, it is very interesting that one of the children who had repeated self-injurious behavior tried to explain the problem with the application, which some of other children had made terrible fun of him, then he stopped hurting himself. The most important thing is the application brought a lot of smile to all children. It is obvious “Let’s Talk!” made great effect to those children with speech disorders.

This application may have much possibility to be used by not only autistic children but also people who have problems of communication because of some diseases, such as pharyngeal cancer, cerebral palsy from a stroke, or senile dementia. We believe this application will help all the people with or without any disabilities to live in better life. Communication is essential needs for human. If people with speech disorders can show their thoughts and feelings with “Let’s Talk!”, it will be possible that people can understand and relate each other more deeply. The most important aim of this application is to let autistic children to feel the joy of communication and encourage them to communicate with others. We consider it will lead the society where people support each other.
References


Ippei Torii received B.F.A. degree in sculpture of fine arts from Nihon University, Tokyo, Japan, in 1982. From 1982 to 1984 he was at the school of fine arts in Nihon University. From 2007 he has been an associate professor in Aichi Institute of Technology. He received the Ph.D degree in Management and Information Science from Aichi Institute of Technology in 2012. His research interests include computer graphics, HCI and visual design. He is a member of IEEE, ACIS and Design Research Association.

Kaoruko Ohtani received the BE degree in Education from Nanzan University in 1992.

Takahito Niwa is a bachelor course student in Aichi Institute of Technology.

Naohiro Ishii received the BE and ME degrees and the Ph.D degree in Electrical Communication Engineering from Tohoku University, in 1963, 1965 and 1968, respectively. He was a professor in Computer Science and Engineering, Nagoya Institute of Technology and in Information Science, Aichi Institute of Technology, respectively. He is now a visiting professor in Information Science, Aichi Institute of Technology. His research interest includes computer system, artificial intelligence, data mining and web applications. He is a member of IEEE and ACM.
Micro Area Selection Framework for ICT Infrastructure Diffusion based on Commuting Flow

Motoi Iwashita\textsuperscript{1}  
Chiba Institute of Technology, Chiba, Japan

Takeshi Kurosawa\textsuperscript{1}  
Tokyo University of Science, Tokyo, Japan

Akiya Inoue\textsuperscript{2}  
Chiba Institute of Technology, Chiba, Japan

Ken Nishimatsu\textsuperscript{1}  
NTT, Tokyo, Japan

Abstract

In Japan, social demands including a rapidly aging society, effective utilization of resources and energies, and environmental protection are on the rise. ICT infrastructures such as wired/wireless information networks are poised to play an important role in meeting these demands, and there is an urgent need to facilitate nationwide ICT infrastructure immediately. Such facilitation requires a large expenditure in terms of time and money, and the selection of which areas to target and in which order are important issues. Low-usage facilities can cause huge problems for businesses in terms of investment efficiency, and it takes time to select areas because we need to consider how to estimate the potential demand and how to diffuse the installation for thousands of municipal areas across the nation. In this paper, we propose framework for micro area selection with service diffusion and potential demand forecast. We constructed a target area selection method in which service reputation on an ICT infrastructure propagates based on commuting flow. A comparison with the performance of a WiMAX service currently in use demonstrates that the proposed method can function as the first step of facilitating nationwide ICT infrastructure.

Keywords: Data mining, area marketing, demand forecast, ICT services, service and infrastructure diffusion.

1. Introduction

Broadband access services can be rapidly deployed by asymmetric digital subscriber line (ADSL) penetration [1].

This has enabled consumer-generated media (CGM) such as social networking service (SNS) and YouTube to be widely used on the basis of a broadband access infrastructure. These multimedia services have dramatically changed the modern life-style and make it possible for individuals to obtain and share information with ease. Some local governments are trying to utilize ICT infrastructures for healthcare, nursing, and so on in their regions [2].

Fiber-to-the-home (FTTH), which has been provided in Japan since 2002, is an ultra high-speed broadband access infrastructure. Although the coverage rate of FTTH for telephone customers is very high at about 84\%, nationwide, in 2009 [2], the FTTH customer rate of telephone customers is low, at about 35\%. Moreover, the areas where FTTH is not installed tend to be those with small populations, according to data in 2005 [3], [4]. As for wireless broadband access, long-term evolution (LTE) for high-speed wireless access has been provided since December 2010. The LTE coverage rate for the population is about 25\% and the customer rate is about 4\% as of the end of FY 2011. The other high-speed wireless access, Worldwide Interoperability for Microwave Access (WiMAX), which has been provided since 2009, has a coverage rate in government-decreed cities those population is greater than 500 thousand of about 95\% and the total number of customers is about 1 million.

Business management is deeply affected by low usage of facilities because the cost of installing the infrastructure is very high. It is hard to find the areas with low usage when we focusing on the average data. This demonstrates the importance of not only considering macro areas but also micro areas when installing the infrastructure. We also need to ensure that the installation for ICT infrastructure such as wired and wireless broadband access facilities is both strategic and economic. Such installation strongly depends on our estimation of potential demand by area, so an efficient method for performing this estimation is really needed.

The smart-grid has been attracting a lot of attention in the ICT infrastructure field. Specifically, energy management systems (EMSs) can help us understand consumed power, control the electricity in the home, and more. As with ICT, EMS installation requires both economy and efficiency, and challenges concerning which areas to use are always present in facilitating infrastructure. In order to realize high efficiency for ICT services, a strategic framework with geographic service diffusion and potential demand forecast should be constructed.
The goal of exploiting area with ICT services is to determine the investment order of areas, as ICT infrastructure installation per area is both more effective and less expensive than on-demand installation. If we can come up with an efficient method, it should have a lasting impact on enterprise business.

In this paper, we first describe a framework of micro area marketing that consists of a target area selection and potential demand forecast. We then propose a micro area selection method for ICT infrastructure diffusion and compare its performance with that of WiMAX.

2. Related works

“Target area marketing” refers to the selection of areas to target for a sales promotion or for installing a service diffusion facility. Trade analysis [5] is commonly used for this type of market research. To give a practical example, it might be used to determine where a convenience store should be constructed in order to attract the most customers. A geographic information system (GIS) [6] is an effective tool for area determination. An empirical study using GIS for trade analysis has previously been proposed [7], and this and similar approaches have proven effective for deciding whether a shop should be established in a given area or not.

The main problem with these methods for ICT infrastructure installation is that it takes huge amount of time to select the areas. Since services based on ICT infrastructure have network externality features that make it more convenient to have more users [8], there is little effect if the installation is only for one area. Therefore, in terms of nationwide installation, we should focus on several areas simultaneously.

ICT infrastructure seems to be diffused by the effect of service reputation and usability in areas in which as ICT infrastructure already is provided. Individual behavior such cases has been described as “go type behavior” where individuals tend to go to issues/people when forming preferences [13], [14]. Therefore, it is important to consider who exactly pushes forward service diffusion. An innovative early adopter in a technology life cycle is characterized as an information source who has a strong influence on acquaintances from the innovative diffusion viewpoint [15]. Having a contact and then becoming interested in something through this contact are generally considered key diffusion factor.

![Type of contact with early adopter](image)

Figure 2. Type of contact with early adopter.

Although it is difficult to identify each person in an area from the viewpoint of technology life cycle (e.g., who is an early adopter), a city planning study that uses cell phones (life log data) to obtain mobility data by area has just been started from the viewpoint of mobility [16]. These researchers expect to analyze human behavior by utilizing the life log data, and if it works well, the results can be applied to various fields. However, at the moment, there are no studies on infrastructure diffusion considering human behavior in the determination of potential demand.

3. Hypothesis of facility diffusion among areas

In general, early adopters play an important role in the diffusion of innovation in that they tend to pass on information on items of interest to their acquaintances [15]. Here, we explain how ICT infrastructure diffuses as innovation. The reputation of audio-visual (AV) and digital equipment is constructed largely on the basis of its usefulness and on the impressions shared among users, and consumers tend to be most influenced by people who have the same preferences. The services based on ICT
infrastructure take the same behavior by sharing information among users. CGM is deeply involved in network externality; for example, a study on Internet video sites showed that the ratio of users who joined the sites after recommendations from friends/acquaintances is about 38% [17]. The propagation speed of Twitter is now even faster than that of Internet video sites such as YouTube, as shown in an investigation of Hotlink [18].

As for log data, since the diffusion of ICT infrastructure includes early adopter individuals. Therefore, we assume a possibility of service diffusion through contacts with many early adopters, the possibility that he/she will purchase the same way they respond to the service upgrade [20]. Since the ICT infrastructure is not a free service, we expect customers to respond to it in the same way they respond to software and service upgrade purchases.

Next, we explain the relationship between an early adopter and other individuals on the basis of personal behavior [20]. If an individual has many contacts with several early adopters, the possibility that he/she will demand the service is high, as shown in Fig. 1.

These considerations, along with the results in [15], show that the mobility of an early adopter induces a high possibility of service diffusion through contacts with many individuals. Therefore, we assume a commuting flow that includes early adopters in the service diffusion process, as shown in Fig. 2.

Here, we describe commuting flow in more detail. Since the diffusion of ICT infrastructure strongly depends on the application (SNS, Net-Game, etc.), it would be desirable to identify commuting flow on the basis of attribute (employee, student, etc.). In addition, if we could access life log data, we could perform an even more detailed analysis. As for how often commuting flow information should be collected, we assume that an average commuting flow per day or yearly commuting flow is used. This is because the interval of ICT infrastructure installation is not especially frequent (no real time installation).

Therefore, the following hypotheses are given for ICT infrastructure diffusion considering an early adopter (Fig. 3). In one type of service diffusion, the early adopters (Area-1, 2, and 3) gather in one area (Area-A) and have contact with individuals in the same area, as shown in the upper part of Fig. 3. In the other type of service diffusion, early adopters in one area (Area-A) visit the other areas (Areas-1 and -2) to make contact with individuals and then return to their own area (as shown in the lower part of Fig. 3).

Hypothesis 1): An area with high commuting inflow from several areas has a high possibility of diffusing ICT infrastructure via early adopters.

Hypothesis 2): An area with high outflow to the other areas has a high possibility of diffusing ICT infrastructure via early adopters.

The diffusion of ICT infrastructure considers region because the distance among areas deeply depends on in- and outflows.

4. Framework of micro area marketing

The framework of target area selection based on diffusion of ICT infrastructure and potential demand forecast is shown in Fig. 4. There are thousands of micro areas across the nation in the form of municipal units, some of which have an ICT infrastructure that has already been installed ("ICT provided area") and some of which do not ("ICT non-provided area"). Therefore, the key function of Step 1 is to efficiently select a high number of target areas that are geographically scattered. Here, the selection of areas means that the area has a big potential demand for an ICT infrastructure.
In Step 2, we forecast the expected number of demands for ICT infrastructure in non-provided areas. We perform this forecasting by analyzing the characteristics of areas in which an ICT infrastructure is already provided. These characteristics are determined by demographic data based on ICT infrastructure provided areas, and the potential demand in each area without providing ICT is estimated using this model through multi-regression analysis. We repeat these steps if necessary.

In Step 3 (the final step), the potential demand for an ICT infrastructure among various areas is compared and the target area is then decided. Since an efficient area selection method is required to enable network externality, we have mainly focused on the first step in this paper.

5. Representative area selection algorithm

In order to analyze in- and outflows among areas, we created a graph model in which an area is a node and a commuting flow among the areas is a link. Each link has an arrow to indicate the direction of commuting flow.

![Table of commuting in- and outflows among areas](image)

<table>
<thead>
<tr>
<th>In</th>
<th>Area 1</th>
<th>Area 2</th>
<th>Area 3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area 1</td>
<td>( f_{i1} )</td>
<td>( f_{i2} )</td>
<td>( f_{i3} )</td>
<td></td>
</tr>
<tr>
<td>Area 2</td>
<td>( f_{21} )</td>
<td>( f_{22} )</td>
<td>( f_{23} )</td>
<td></td>
</tr>
<tr>
<td>Area 3</td>
<td>( f_{31} )</td>
<td>( f_{32} )</td>
<td>( f_{33} )</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Classification policy](image)

Let G is a graph such that \( G = (N, L, F) \), where N is a set of nodes \( (n_i \in N) \), L is a set of links \( (l_{ij} \in L) \), and F is a set of link flows \( (f_{ij} \in F \text{ on } l_{ij}) \).

An example of the graph model construction based on commuting flow table in action is shown in Fig. 5. A table of commuting in- and outflows among areas was prepared in advance considering the direction from one area to another. A link was added among the specified areas if in/outflow was greater than the given threshold, \( \alpha \). In other words, we selected links that contained a high average number of commuters. This procedure is repeated until there are no longer any links that are greater than the threshold.

The procedure for selecting an area (shown in Fig. 6) is as follows.

![Representative area selection algorithm](image)

1) Algorithm of graph construction part

Step 1: \( f_d = \max \{ f_{ij} \} \in F \) by descending order. Set \( f_d, \alpha \in F \) and \( G^* = (N^*, L^*, F^*) \) is an empty set.

Step 2: If \( f_d \geq \alpha \) then the following procedures continue.
- Step 2-1: Add arrowed link \( l_{ij} \) from node “i” to node “j”, add nodes “i” and “j” to \( N^* \), add link \( l_{ij} \) to \( L^* \), and add flow \( f_{ij} \) to \( F^* \). If several arrowed links exist, all corresponding nodes, links and flows should be added to \( N^*, L^* \), and \( F^* \), respectively.

Step 2-2: Replace \( F \) with \( F \setminus \{ f_d \} \)

Step 2-3: \( f_d = \max \{ f_{ij} \} \in F \). Finally, graph \( G^* = (N^*, L^*, F^*) \) is obtained.

2) Algorithm of node degree calculation part

Step 3: Give threshold values \( \beta \) and \( \gamma \)

Step 4: Find node “i” in \( N^* \) with its in-degree \( \geq \beta \), and select node “i” in \( G^* \).

Step 5: Find node “i” in \( N^* \) with its out-degree \( \geq \gamma \) and select node “i” in \( G^* \).
6. Application to ICT infrastructure

6.1 Estimation of commuting flows among areas

If we use life log data flexibly, it is possible to determine the movement of people both efficiently and precisely. However, at the moment it is still difficult to access such data, so in this study we used census data as the approximate information. Although this enabled us to determine the total flows to and from a given area, it was impossible to determine the in- and outflows between specified areas, especially when using the latest census data. We therefore had to estimate the in- and outflows (Fig. 7).

The gravity model defined in (1) is used in this paper. Let $f_{ij}$ be a flow from area “$i$” to area “$j$”, then

$$f_{ij} = K \cdot r_{ij} \cdot p_i^a \cdot q_j^b / d_{ij}$$

(1)

where $K$, $a$, $b$, and $c$ are constants, $r_{ij}$ is represented as a commuting effect from area “$i$” to area “$j$” defined by the total employees in each area, $p_i$ is the total commuting outflows from area “$i$”, $q_j$ is the total commuting inflows to area “$j$”, and $d_{ij}$ is the distance between “$i$” and “$j$”.

We considered 97 municipal areas in one prefecture (Fukuoka). The in- and outflows between all 97 areas were calculated using census data from 2007.

We used the census data to obtain the total flow difference of an area and then estimated the in- and outflow results, as shown in Fig. 8 (a). Although some areas have a marked difference in outflow, the RRD as defined in (2) is 4.4%.

$$RRD = \left[ \sum_i \left( f_i - r_i \right)^2 / r_i \right]^{1/2}$$

(2)

where $f_i$ is the estimated total outflows from node “$i$” and $r_i$ is the real value.

In terms of the difference in inflow, the RRD is about 10%, as shown in Fig. 8 (b). Since the difference did not exceed about 10% in either result, we used this gravity model to calculate the in- and outflows between specified areas.

6.2 Feature of the proposed algorithm

First, we briefly discuss the diffusion of WiMAX [21] and evaluate its effectiveness using the hypothesis given in this paper.

WiMAX is a wireless broadband access service that has been in place in Japan since 2009. We looked at 97
municipal areas in the same prefecture and used the same 2007 census data.

The areas selected by the algorithm are shown in Fig. 9 under the condition of $\beta = 3$ and $\gamma = 1$ as thresholds. Areas 9 and 10 are connected to the other 12 areas in the case of $\alpha = 10,000$ (Fig. 9 (a)). All areas except areas 30, 34, 39, 42, and 46 in this graph are the selected areas; the other 11 areas are added if $\alpha = 5,000$ (Fig. 9 (b)). In this second case, all areas except areas 4, 16, 33, 35, 36, 37, 39, 40, 41, and 43 are selected. This is because area 4 only has an inflow link from area 5, for example. These results strongly depend on the value setting of $\alpha$. This $\alpha$ is not a fixed value, but has to be adjusted in consideration of service penetration in real field operation. That is to say, $\alpha$ is given a high value at the first stage and then the value decreases gradually. Therefore, we need a lot of case studies to determine the optimum threshold.

![Figure 9. Results of area selection in prefecture A.](image)

Let us consider the effects of thresholds $\beta$ and $\gamma$, as these thresholds also have a strong influence on the area selection. An area selection result when the condition was changed to $\alpha = 10,000$, $\beta = 2$, and $\gamma = 0$ is shown in Fig. 10. Area 9 is connected to the other 11 areas. Since all 12 areas are the selected areas, the number of selected areas in this condition is greater than the one in the case of $\beta = 3$ and $\gamma = 1$ (Fig. 9 (a)) due to the weakened area selection condition. Since it is very difficult to decide the values of $\beta$ and $\gamma$, it is necessary to evaluate many combination sets of $\beta$ and $\gamma$ and then cross-reference the results with the obtained field data.

![Figure 10. Results of area selection with condition $\beta = 2$ and $\gamma = 0$.](image)

6.3 Area comparison for WiMAX

We compared the performance of the proposed algorithm with that of a provider and the conventional method in order to determine the feasibility of the proposed algorithm. The results of the provider (WiMAX evolution) and the proposed algorithm (proposed method with $\beta = 3$ and $\gamma = 1$ as thresholds) are shown in Table 1.

In this comparison, the concordance ratio (CR) in October 2011 was 84% (= 26/31). The ratio of the number of selected areas between WiMAX evolution and the proposed method is obtained by the following equation.

$$CR = \frac{\text{(Number of the selected areas matching WiMAX evolved areas)}}{\text{(Number of WiMAX evolved areas)}}$$

The CR grew rapidly: from 50% in August 2010 to 80% in December 2010. The order of the WiMAX infrastructure installation areas is also shown in Table 1. This is because, in 2010, the area selected interval was one month. Since introducing the ICT infrastructure takes a long time, ideally the interval should be several months. In such cases, the concordance ratio is considered to be 81% in 2010 and 84% in 2011.

6.4 Comparison with the conventional method

In this section, we compare and evaluate the proposed and conventional methods by considering the difference between the WiMAX evolution results. Table 2 shows the compared results. Here (a), (b), (c), and (d) are the selected area number in the WiMAX evolution, the selected area
number by the proposed algorithm with $\beta = 3$ and $\gamma = 1$, the selected area number by the proposed method with $\beta = 2$ and $\gamma = 0$, and the selected area number by population order, respectively. The concordance ratio of each case is described below in the cells of (b), (c), and (d).

<table>
<thead>
<tr>
<th>WiMAX evolution</th>
<th>Proposed method</th>
<th>Concordance ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area No.</td>
<td>Area No.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Area Selection Order Of Wimax Introduction

The comparison of (b) and (c) is described from the viewpoint of threshold value sensitivity. The CR of (b) is greater than that of (c). The proposed algorithm in the case of (c) tends to select the areas with $\gamma = 0$ where many commuters are concentrated on moving to only one specified area. Hence, the gap between (a) and (c) is greater if the specified area has very little communication with the other areas. Therefore, thresholds $\beta$ and $\gamma$ should be adequately adjusted.

Finally, we discuss the comparison of (b) and (d). The concordance ratio of (b) is greater than or equal to that of (d) in Table 2. This means that the evolution strategy of the WiMAX provider is not as simple as considering the population order, suggesting that potentiality might include other factors that can be used to forecast potential demand. Highly populated areas seem to tend to have many early adopters, but merely existence of early adopters gives little effect for increasing potential demand. The movement of early adopters will provide the stimulus for potential demand like our hypothesis.

Table 2. Area Selection Order In Prefecture A

<table>
<thead>
<tr>
<th>Year</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</td>
<td>2, 4, 5, 6, 7, 8, 9, 10</td>
<td>5, 7, 8, 9, 10</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</td>
</tr>
<tr>
<td>2011 Jan.-Oct.</td>
<td>15, 17, 18, 19, 29</td>
<td>18, 29</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2011 Nov.-Dec.</td>
<td>11, 12, 13, 14, 16</td>
<td>11, 12, 13, 14, 16, 18, 29, 30</td>
<td>31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 46</td>
<td>31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 46</td>
</tr>
</tbody>
</table>

6.5 Comparison with other areas

We evaluated the proposed algorithm from the viewpoint of prefecture size. To determine the difference of results depending on region, three prefectures are considered: A (Fukuoka), B (Ibaraki), and C (Kagawa). The features are shown in Table 3.

<table>
<thead>
<tr>
<th>Year</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</td>
<td>2, 4, 5, 6, 7, 8, 9, 10</td>
<td>5, 7, 8, 9, 10</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</td>
</tr>
<tr>
<td>2011 Jan.-Oct.</td>
<td>15, 17, 18, 19, 29</td>
<td>18, 29</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2011 Nov.-Dec.</td>
<td>11, 12, 13, 14, 16</td>
<td>11, 12, 13, 14, 16, 18, 29, 30</td>
<td>31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 46</td>
<td>31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 46</td>
</tr>
</tbody>
</table>

Table 3. Conditions of Prefectures

<table>
<thead>
<tr>
<th>Prefecture</th>
<th>Pref. A</th>
<th>Pref. B</th>
<th>Pref. C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>5,500,000</td>
<td>2,980,000</td>
<td>1,012,000</td>
</tr>
<tr>
<td>Number of areas</td>
<td>97</td>
<td>54</td>
<td>34</td>
</tr>
<tr>
<td>Average population per area</td>
<td>56,701</td>
<td>55,185</td>
<td>29,764</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>62,754</td>
<td>53,655</td>
<td>58,220</td>
</tr>
<tr>
<td>Number of customers</td>
<td>71,700</td>
<td>26,000</td>
<td>9,600</td>
</tr>
</tbody>
</table>

The population of A is the largest, that of B is in the middle, and that of C is the smallest. A contains 97 areas, B contains 54, and C contains 34. Although prefectures A and B have almost the same average population per area, the standard deviation of A is greater than that of B.
prefecture C has two extremely big areas, the standard deviation is very large. Moreover, prefecture B has a large commuting flow to Tokyo, which is considered in the algorithm.

**Table 4. Area Selection Order In Prefecture B**

<table>
<thead>
<tr>
<th>Year</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>1, 2, 3, 4, 15, 16, 17</td>
<td>1, 2, 3, 15, 16, 17</td>
<td>1, 2, 3, 5, 7, 14, 16</td>
<td>1, 2, 3, 4, 16, 17</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>26</td>
<td>[86%]</td>
<td>[71%]</td>
</tr>
<tr>
<td>2011</td>
<td>5, 6, 7, 8, 9, 10, 11, 14, 18, 20, 22, 26, 30</td>
<td>2, 4, 5, 6, 7, 9, 10, 11, 18, 19, 20, 21, 22, 23, 24, 26</td>
<td>5, 6, 7, 8, 9, 10, 11, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29</td>
<td>5, 6, 7, 8, 9, 10, 11, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29</td>
</tr>
<tr>
<td></td>
<td>[75%]</td>
<td>[67%]</td>
<td>[79%]</td>
<td>[71%]</td>
</tr>
</tbody>
</table>

Tables 4 and 5 show the results of area selection order in the cases of prefectures B and C, respectively. The concordance ratio of (b) is greater than those of (c) and (d) for both cases, with a similar result for prefecture A.

**Table 5. Area selection order in Prefecture C**

<table>
<thead>
<tr>
<th>Year</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>1, 2, 3, 6, 11, 20</td>
<td>1, 2, 3, 6, 7, 12, 13, 25, 29, 30</td>
<td>1, 2, 3, 5, 6, 7</td>
<td>1, 2, 3, 5, 6, 7</td>
</tr>
<tr>
<td></td>
<td>[83%]</td>
<td>[67%]</td>
<td>[67%]</td>
<td>[67%]</td>
</tr>
<tr>
<td>2012</td>
<td>4, 5, 7, 17, 18, 22, 26, 31, 32, 34</td>
<td>4, 5, 11, 12, 14, 19, 22, 24, 26, 29, 30</td>
<td>3, 4, 5, 7, 12, 20, 4, 9, 11, 12, 14</td>
<td>3, 4, 5, 7, 12, 20, 4, 9, 11, 12, 14</td>
</tr>
<tr>
<td></td>
<td>[69%]</td>
<td>[63%]</td>
<td>[63%]</td>
<td>[63%]</td>
</tr>
</tbody>
</table>

### 6.6 Consideration of gapped areas

In terms of installation, there are selection order gaps of more than 12 months for area 44 in A and areas 32, 36, and 48 in B. In reality, there are many features in area 44 in A related to commuting flows to areas 9 and 10. However, the census data from 2007 could not corroborate this situation. This was in contrast to areas 32, 36, and 48 in B, all of which were merged with the other areas after 2007. This suggests that the provider might use the area data after merging areas. Further analysis of these facts will be the objective of our future work.

In the real world, results have been interpreted through various lenses, such as economy and service competition, as well as potential demand forecast. Therefore, the results obtained by the proposed method do not necessarily always fit with that in the real world. However, a first step is our consideration of the effect of commuting on the area selection.

### 6.7 LTE case

We also applied our method to LTE [22], which is a wireless broadband access service provided for cell phones. LTE started about 2 years ago, in December 2010. At present, the concordance ratio of the areas selected by the proposed method with those for LTE evolution is about 73% (= 29/40) in prefecture A. Although it is necessary to evaluate continuously, at the moment it is possible to apply other ICT infrastructures.

### 7. Conclusion

It is tremendously important to select the area in which an ICT infrastructure is introduced in order to ensure the quick and economical development of an advanced information society. Area selection has a strong dependence on potential demand, and one of the main features of ICT infrastructure is network externality, so a huge amount of time and labor is required to select specified areas from among a large number of candidate areas.

In this paper, we modeled the movement of early adopters as commuting flow among areas, provided a framework of micro area marketing for ICT infrastructure installation, and proposed an efficient area selection method. We evaluated the proposed method using real field data and obtained empirically valid results. However, we require much more micro area data to definitively verify the method and to obtain the characteristics of each micro area.

Our future work will involve obtaining detailed movement data such as life logs and verifying/improving the method. We also need to consider the second step in a micro area marketing framework in order to improve the application of commuting flow as an important factor in multi-regression analysis.
References


[22] http://www.3gpp.org/Technologies/Keywords-Acronyms/LTE.

Motoi Iwashita joined NTT Corporation in 1985. He received his B.S. and M.S. in Mathematics and Dr. Eng. degrees in Electrical Engineering from Waseda University, Tokyo, in 1983, 1985, and 1999, respectively. He has engaged in research on modeling and evaluation method of optical access networks, telecom operation support system development, and multi-service teletraffic network design. In 1992 and 1993, he was an Exchange Researcher at the British Telecom Research Laboratories, England. He is currently an Associate Professor at the Department of Management Information Science, Chiba Institute of Technology. He is a member of COM Society of IEEE.

Akiya Inoue joined NTT Public Corporation in 1981. He received his B.E., M.E. and D.E. degrees in Electrical Engineering from Nihon University, Tokyo, in 1979, 1981, and 1995, respectively. He has been engaged in research on network control and management methods and on new service demand forecasting based on user behavior modeling. From 2004 to 2005, he was an Executive Researcher, Communication Traffic & Service Quality Project, NTT Service Integration Laboratories. He is currently a Professor, Department of Management Information Science, Faculty of Social Systems Science, Chiba Institute of Technology. He is a member of COM and SMC Society of IEEE, IEICE of Japan, and the Operations Research Society of Japan.

Takeshi Kurosawa received his B.S., M.S. and Dr. degrees in Mathematics from Keio University, Tokyo, Japan, in 2001, 2003, and 2009, respectively. He joined NTT (Research and Development) in 2003. He was engaged in research on demand analysis for telecommunications and internet services and on a service-choice behavior model using discrete choice theory from 2003 to 2010 in NTT Laboratories. He is currently a Lecturer of Department of
Mathematical Information Science, Tokyo University of Science. He is a member of IEICE of Japan, and the Operations Research Society of Japan.

Ken Nishimatsu got his B.E., M.E. and D.E. degrees from Waseda University, Tokyo, Japan, in 1995, 1997, respectively. He joined NTT in 1997. He is engaged in research on demand analysis for telecommunications and internet services. He is currently a Senior Research Engineer of the Traffic Solution Group, NTT Network Technology Laboratories. He is a member of IPSJ, IEICE of Japan, and the Operations Research Society of Japan.
Automatic Generation of SVG Program Documents with Animation Based on Attribute Graph Grammars

Takaaki Goto†
The University of Electro-Communications, Japan

Tadaaki Kirishima*
Oberlin University, Japan

Takeo Yaku**
Oberlin University, Japan

Tomoo Sumida†
Toyo University, Japan

Tetsuro Nishino‡‡
The University of Electro-Communications, Japan

Kensei Tsuchida‡‡
Toyo University, Japan

Abstract

We have developed a software development environment using the program diagram language Hichart [1, 2, 3]. In our research, we implemented a graphical editor for Hichart. We added new features and capabilities to the editor, including calculating the cyclomatic complexity and generating SVG files for a given Hichart diagram. In this paper, we present a method for automatically generating an SVG file for a given program diagram based on attribute graph grammars. The SVG file can animate a process for calculating the cyclomatic complexity of a given Hichart diagram on any readily available Web browser. We also describe how to automatically generate an SVG file that can be displayed when diagrams are drawn.

Keywords: program diagrams, Hichart, graph grammar, SVG, animation.

1. Introduction

We have developed a software development environment based on graph theory that includes graph drawing theory and graph grammars [1, 2, 3, 4]. In this environment, the program diagram called Hichart (Hierarchical flowCHART description language) is used. We have already developed bidirectional translators that translate Pascal, C, and DXL sources into Hichart and translate Hichart into Pascal, C, or DXL [2, 3]. Concerning DXL, the Diagram eXchange Language for tree-structured charts DXL was specified by the ISO in 1997 [5]. Hichart is used as an example. We have implemented a Hichart editor for DXL based on attribute graph grammars.

In the software development environment, features that support programming development should be provided. Therefore, we added a new feature to the editor to calculate the cyclomatic complexity [6] for a given Hichart diagram. However, animation for describing programs on the Web is a main theme in software visualization. Many studies have investigated this theme [7, 8, 9, 10, 11, 12]. However, automatic animation generation based on a formal method has not yet been researched.

We generated automatic animations by using the semantic rules of attribute graph grammar. Our approach in realizing this theme is as follows. First, we introduce an attribute for generating SVG text [13]. Next, we define semantic rules to satisfy the constraints of the intended animation, by referring to other attributes. Then, we evaluate the attributes while traversing the derivation tree for a given Hichart diagram. Finally, we obtain the SVG source file as a value of the attribute. The SVG file can animate a process that computes the cyclomatic complexity on any readily available Web browser.

This paper is organized as follows: In Section 2, we discuss preliminaries. In Section 3, we explain our method for automatically generating SVG documents. We conclude in Section 4.

2. Preliminaries

2.1 Attribute edNCE graph grammar [4, 14]

Definition 1 Let $\Sigma$ be an alphabet of node labels and $\Gamma$ be an alphabet of edge labels. A graph over $\Sigma$ and $\Gamma$ is a tuple $H = (V, E, \lambda)$, where $V$ is the finite set of nodes,

$$E \subseteq \{(v, \gamma, w) : v, w \in V, v \neq w, \gamma \in \Gamma\}$$

is the set of edges, and

$$\lambda : V \rightarrow \Sigma$$

is the node labeling function.

$$E(v, w) = \{\gamma \in \Gamma | (v, \gamma, w) \in E\}.$$ The label tuple of two nodes $v, w \in V$ is

$$lab(v, w) = (\lambda(v), E(v, w), E(w, v), \lambda(w)).$$

Definition 2 Two graphs $H$ and $K$ are isomorphic if there is a bijection $f : V_H \rightarrow V_K$ such that
Definition 3 The set of all concrete graphs over $\Sigma$ and $\Gamma$ is denoted as $GR_{\Sigma,\Gamma}$, and the set of all abstract graphs is denoted as $[GR_{\Sigma,\Gamma}]$. A subset of $[GR_{\Sigma,\Gamma}]$ is called a graph language.

Definition 4 A graph with (neighborhood controlled) embedding over $\Sigma$ and $\Gamma$ is a pair $(H,C)$ with $H \in GR_{\Sigma,\Gamma}$ and $C \subseteq \Sigma \times \Gamma \times \Gamma \times V_H \times \{\text{in, out}\}$. $C$ is the connection relation of $(H,C)$, and each element $(\sigma, \beta, \gamma, x, d)$ of $C$ with $\sigma \in \Sigma, \beta, \gamma \in \Gamma, x \in V_H$ and $d \in \{\text{in, out}\}$ is a connection instruction of $(H,C)$. A connection instruction $(\sigma, \beta, \gamma, x, d)$ will always be written as $(\sigma, \beta / \gamma, x, d)$. Two graphs with embedding $(H,C_H)$ and $(K,C_K)$ are isomorphic if there is an isomorphism $f$ from $H$ to $K$ such that $C_K = \{(\sigma, \beta / \gamma, f(x), d) | (\sigma, \beta / \gamma, x, d) \in C_H\}$. The set of all graphs with embedding over $\Sigma$ and $\Gamma$ is denoted as $GRE_{\Sigma,\Gamma}$.

Definition 5 An edNCE graph grammar is a tuple $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$, where $\Sigma$ is the alphabet of node labels, $\Delta \subseteq \Sigma$ is the alphabet of terminal node labels, $\Gamma$ is the alphabet of edge labels, $\Omega \subseteq \Gamma$ is the alphabet of final edge labels, $P$ is the finite set of productions, and $S \in \Sigma - \Delta$ is the initial nonterminal.

A production is of the form $X \rightarrow (D,C)$ where $X$ is a nonterminal node label, $D$ is a graph over $\Sigma$ and $\Gamma$, and $C \subseteq \Sigma \times \Gamma \times \Gamma \times V_D \times \{\text{in, out}\}$ is the connection relation, which is a set of connection instructions. A pair $(D,C)$ is a graph with embedding over $\Sigma$ and $\Gamma$.

Definition 6 A copy($P$) denotes the infinite set of all productions that are isomorphic to a production in $P$; an element of copy($P$) is called a production copy of $GG$.

Definition 7 Let $(H,C_H)$ and $(D,C_D)$ be two graphs with embedding in $GRE_{\Sigma,\Gamma}$, such that $H$ and $D$ are disjoint, and let $v$ be a node of $H$. The substitution of $(D,C_D)$ for $v$ in $(H,C_H)$, denoted as $(H,C_H)[v/(D,C_D)]$, is the graph with embedding $(V,E,\lambda,C)$ in $GRE_{\Sigma,\Gamma}$ such that $V = (V_H - \{v\}) \cup V_D$.

Definition 8 Let $G = (\Sigma, \Delta, \Gamma, \Omega, P, S)$ be an edNCE graph grammar. Let $H_{i-1} = (V_{H_{i-1}}, E_{H_{i-1}}, \lambda_{H_{i-1}})$ and $H_i = (V_{H_i}, E_{H_i}, \lambda_{H_i})$ be graphs in $GRE_{\Sigma,\Gamma}$. In addition, let $v_i \in V_{H_{i-1}}$, and $p_i : X \rightarrow (D_i', C_i')$ be a production copy $P$ of $G$ such that $D_i'$ and $H_{i-1}$ are disjoint. $s_i = (p_i', v_i, D_i', b_i')$ is a derivation specification of $G$ if $p_i' \in \text{copy}(P)$.

Let $\lambda_{H_{i-1}}(v_i) = X, D_i' \subseteq D, b_i' : V_{D_i'} \rightarrow V_D$. We write $H_{i-1} \rightarrow_{s_i} H_i$ or just $H_{i-1} \rightarrow_h H_i$, if $\lambda_{H_{i-1}}(v_i) = X$ and $H_i = H_{i-1}[v_i/(D', C')]$. $H_{i-1} \rightarrow_h H_i$ is called a derivation step, and a sequence of such derivation steps is called a derivation.

Definition 9 An Attribute edNCE Graph Grammar[4] is a tuple $AGG \ll GG, Att, F >$, where

1. $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$ is called an underlying graph grammar of AGG. Each production $p$ in $P$ is denoted by $X \rightarrow (D,C)$. $\text{Lab}(D)$ denotes the set of all occurrences of the node labels in graph $D$.
2. Each node symbol $Y \in \Sigma$ of $GG$ has two disjoint finite sets $\text{Inh}(Y)$ and $\text{Syn}(Y)$ of inherited and synthesized attributes, respectively. The set of all attributes of symbol $X$ is defined as $\text{Att}(X) = \text{Inh}(X) \cup \text{Syn}(X)$.
3. Associated with each production $X_p \rightarrow (D,C) \in P$ is a set $F_p$ of semantic rules that define all the attributes in $\text{Syn}(X_0) \cup \cup_{X \in \text{Lab}(D)} \text{Inh}(X)$. A semantic rule defining an attribute $a(X_{i0})$ has the form $a_0(X_{i0}) := \alpha_0(X_{i0}) \ldots a_m(X_{im})$. Here $f$ is a mapping from $V(a_1(X_{i1})) \ldots \chi V(a_m(X_{im}))$ into $V(a_0(X_{i0}))$. In this situation, we say that $a_j(X_{i0})$ depends on $a_j(X_{ij})$ for $0 \leq j \leq m$ in $p$. The set is called the set of semantic rules of $G$.

2.2 Hichart

Hichart[1] is a program diagram language with the fol-
lowing characteristics. (1) A diagram is a tree-flowchart that has the flow control lines of a Neumann program flowchart. (2) The nodes of the different functions in a diagram are represented by differently shaped cells. (3) The hierarchy of the data structure represented by the diagram and the control flow are simultaneously displayed on a plane. Figure 1 shows examples of Hichart symbols. In Figure 1, (A), (B), (C), and (D) indicate “process,” “pre-defined process,” “caption,” and “exclusive selection,” respectively.

![Hichart symbols](image)

**Figure 1** Examples of Hichart symbols

The Diagram eXchange Language for tree-structured charts DXL was specified in 1997 by the ISO [3]. The primary purpose of DXL is to provide a means for exchanging data between different CASE tools for different program flow charts. This allows various CASE tools to use specifications made in the past. Figure 2 shows an example of a Hichart for DXL that describes an example of a DXL code in ISO/IEC 14568 [3].

![Example of a Hichart for DXL](image)

**Figure 2** Example of a Hichart for DXL

The Hichart editor is a graphical hybrid editor that supports syntax-directed editing and free-hand editing.

In syntax-directed editing, a Hichart diagram is generated by a top-down process using the productions of the graph grammar for Hichart/DXL (Hichart for DXL). The grammar provides graph grammatical definitions and attribute rules. The graph grammatical part is defined by a context-free edNCE graph grammar [14]. In free-hand editing, users are allowed to directly manipulate a diagram, so that the diagram generated can be analyzed by a parser based on the grammar. Figure 3 illustrates an example of a Hichart editor screen in free-hand editing.

![Example of Hichart editor screen in free-hand editing](image)

**Figure 3** Example of Hichart editor screen in free-hand editing

### 2.3. An Attribute Graph Grammar for Hichart/DXL

In this section, we briefly describe an attribute edNCE graph grammar that defines Hichart/DXL (Hichart for DXL) Diagrams. The grammar is called PGGHD (Precedence Graph Grammar for Hichart/Dxl) [15]. PGGHD provides production and semantic rules.

A graph grammar for Hichart/DXL is a tuple $\text{PGGHD} = (\Sigma_{\text{HD}}, \Delta_{\text{HD}}, \Gamma_{\text{HD}}, \Omega_{\text{HD}}, \lambda_{\text{HD}}, \phi_{\text{HD}}, S_{\text{HD}})$, where $\Sigma_{\text{HD}}$ is the alphabet of node labels, $\Delta_{\text{HD}} \subseteq \Sigma_{\text{HD}}$ is the alphabet of terminal node labels, $\Gamma_{\text{HD}}$ is the alphabet of edge labels, $\Omega_{\text{HD}} \subseteq \Gamma_{\text{HD}}$ is the alphabet of final edge labels, $\lambda_{\text{HD}}$ is the finite set of productions, and $S_{\text{HD}} \in \Sigma - \Delta$ is the initial graph.

PGGHD includes 70 productions and 888 attribute rules. In PGGHD, [ ] and “ ” denote a nonterminal node and a terminal node, respectively. For example, the node with the “profile” label in Figure 4 means a terminal node; similarly, the node with the [explanation] label indicates a nonterminal node. Nonterminal nodes can be substituted by applying productions; however, terminal nodes cannot be substituted. The nonterminal node with the [module_packet] label is the initial nonterminal.

PGGHD generates a graph that indicates the internal structure of a Hichart diagram. The Hichart diagram is drawn in an undirected graph. However, the internal structure of Hichart is a digraph. PGGHD rewrites edge labels from empty labels to empty labels.
Figure 4 Example showing PGGHD production and semantic rules

Figure 4 shows an example of PGGHD production and semantic rules. A rewriting step by this production consists of removing a nonterminal node labeled [profile] from a given host graph and substituting the graph consisting of "profile" and [explanation] as shown in Figure 5. Note that the numbers above the nodes indicate the node ID to specify the nodes.

Figure 5 Example of applying production using the production from Figure 4

2.4 Derivation of PGGHD

We now provide an example of derivation. Figure 6 illustrates a derivation from the initial nonterminal node with the [module_packet] label. \( G_0 \) is the initial graph.

First, we remove the [module_packet] node in the host graph \( G_0 \), and apply production 1 (abbr. \( P_1 \)), that is, in which the nodes with the “m_packet” and [profile_module_list] label are put. Then we can obtain the graph \( G_1 \).

Next, we remove the [profile_module_list] node and edges that connect with the [profile_module_list] node from host graph \( G_1 \). After that, we embed the daughter graph, including the [profile] node and the [module_list] node, by applying \( P_2 \). Then we establish edges between the nodes of the daughter graph and the nodes that were connected to the [profile_module_list] node using the connection instructions on \( P_3 \).
We can finally obtain a Hichart diagram after applying some productions. Figure 7 shows an example of the Hichart diagram.

![Figure 7 Example of the Hichart diagram](image)

### 2.5 Parsing of PGGHD

In this section, we show the parsing method using graph grammar PGGHD. This parser uses a stack and precedence relation.

Figure 8 illustrates part of the parsing. The parser continues shifting as long as possible using precedence relations. Then a precedence handle is computed. In Figure 8 (a), the "explanation" node is the handle. Next, the parser searches for a production in which the right-hand side is isomorphic to the handle. The parser next reduces the handle to the left-hand side of the production. Figure 8 (b) illustrates a diagram after reduction. The above operations are repeated until the graph becomes the initial graph with the [module_packet] label.

![Figure 8 Parsing example](image)

### 2.6 Features of the Hichart editor

The main features of the Hichart editor are as follows:
1. Checks the correctness of the Hichart diagrams with the parser (in free-hand editing).
2. Draws Hichart diagrams.
3. Calculating the cyclomatic complexity for a given Hichart diagram.
4. Generates an SVG file for a given program diagram and that the display of the SVG file corresponds directly to the diagram in the editor.
5. Generates an SVG file for a given program diagram and that the SVG file can animate a process for computing the cyclomatic complexity on popular Web browsers (e.g., IE).

### 3. Automatic generation of SVG documents

In this section, we describe the methods with which to build the features of (4) and (5) in Section 2.6 by using illustrations.

#### 3.1. Generation of SVG documents with aesthetic drawings

In addition to attributes such as $x$ (x-coordinate), $y$ (y-coordinate), $w$ (width) and $h$ (height), we introduce new attribute $S_{SVG}$, which contains SVG source codes, as the value and representation corresponding to the Hichart diagram. Each node has an $S_{SVG}$ attribute.

Then, we define the semantic rules to satisfy the constraints for drawing Hichart diagrams. The SVG source codes are generated by evaluating $S_{SVG}$. The attributes are evaluated in the bottom-up manner on derivation trees that can be obtained after parsing using semantic rules. Figure 9 gives an example of semantic rules with the attribute $S_{SVG}$ of the production described in Figure 4.

![Figure 9 Examples of semantic rules with the attribute S_{SVG}](image)

In Figure 9, the number in parentheses indicates the node ID; for example, node 0 means a node with the [profile] label shown in Figure 4. Thus, $S_{SVG}(0)$ represents the SVG attribute for node 0. Similarly, $S_{SVG}(1)$ indicates the SVG attribute for node 1 with the "profile" label, and stores the SVG source code that describes the diagram. $S_{SVG}(2)$, the SVG attribute for node 2 with [explanation], stores the SVG source code for all nodes that can be generated from node 2. $S_{SVG}(0)$ stores all SVG source code in $S_{SVG}(1)$ and $S_{SVG}(2)$.

Figure 10 is a derivation tree added with the SVG attributes evaluation process. In this figure, the derivation tree is described in a black line, and the attribute evaluation process is drawn in a bold gray line. In this case, the attribute evalu-
ation for SVG begins from the semantic rules on Production 14. Production 14 has two semantic rules for evaluating the SVG attribute. For example, $S_{SVG}(15)$ sets the drawing information for node 15 in such a rule that `<rect x="x(15)" , y="y(15)" , width="w(15)" , height="h(15)" , fill="white" , fill-opacity="1" , and stroke="black"></rect>$. $S_{SVG}(14)$ is computed in this rule such that $S_{SVG}(14) = S_{SVG}(15)$. That is, $S_{SVG}(14)$ stores drawing information for node 15. In a similar way, the attributes are evaluated in the following order: Production 14 $\rightarrow$ Production 11 $\rightarrow$ Production 8 $\rightarrow$ Production 7 $\rightarrow$ Production 5 $\rightarrow$ Production 10 $\rightarrow$ Production 4 $\rightarrow$ Production 3 $\rightarrow$ Production 1. Finally, $S_{SVG}(1)$ stores all the information that describes the diagram.

Figure 10 Part of the derivation tree with SVG attributes
Figure 11 is an example of the display of a Hichart diagram in SVG. This example illustrates the output of an SVG file in which the diagram in Figure 2 is used as the input. In the Hichart editor, we input the diagram in Figure 2. Then we execute parsing and attribute evaluation of the SVG file. Finally, we obtain the SVG file.

3.2. Generation of SVG documents with animation

Animation is important for understanding the SVG document. We tried to generate an SVG document with an animation that illustrates the computation process for evaluating the cyclomatic complexity attributes on derivation trees.

Generating SVG documents with animation is realized in a manner similar to that described in Section 3.1. We introduce the new attribute $DT_{SVG}$ that contains SVG source codes that contain animation information. Next, we define the semantic rules to satisfy the constraints of the intended animation, by referring to other attributes. Figure 12 is an example of semantic rules with the attribute $DT_{SVG}$. To generate the animation SVG file, the animation start time is needed for each object. Therefore, we previously compute the time using the time attribute.

$DT_{SVG}(1) = DT_{SVG}(0) < ellipse cx="dx0" cy="dy0" rx="nodeSize" ry="nodeSize" stroke="black" > <animateColor attributeName="fill" attributeType="CSS" from="black" to="red" begin="time ....."

Figure 12 Example of semantic rules with the attribute $DT_{SVG}$

Figure 13 and Figure 14 are screen shots of animating a process for calculating the cyclomatic complexity for a given Hichart diagram. In Figure 13, the tree is a derivation tree of a given Hichart diagram, and p1 or p3 indicates the production number. In the animation, the value for the cyclomatic attribute of each node is displayed, and the color of the node changes from black to red according to the order in which the attribute is evaluated.

Figure 13 Screen shot of animating a process for calculating the cyclomatic complexity for a Hichart diagram (in execution animation)

Figure 14 Screen shot of animating a process for calculating the cyclomatic complexity for a Hichart diagram (finish animation)
4. Conclusions

In this paper, we described a method for automatically generating an SVG file for a given program diagram based on attribute graph grammars. We also described the SVG method with animation. The SVG file obtained can animate a process for calculating the cyclomatic complexity of a given Hichart diagram on any readily available Web browser.

The method employed in this paper is a novel approach that automatically generates multimedia-related documents by using animation to describe programs that are platform-independent on the Web.

References


Takaaki Goto graduated with a Doctor of Engineering degree from Toyo University in 2009. He is a Project Assistant Professor at the Graduate School of Informatics and Engineering at The University of Electro-Communications. His main research interests are applications of open source software development, software development education, graph grammars, visual languages, and software development environments. He is a member of IEICE Japan, IPSJ, and IEEE.

Tomoo Sumida graduated with a Doctor of Engineering degree from Toyo University in 2009. He has been a Part-time Lecturer at Toyo University since 2009. His main research interests are network protocol, graph visualization, and software development. He is a member of IPSJ, IEICE Japan, and IEEE.
**Tadaaki Kirishima** received M.E. and D.E. degrees from Toyo University in 2002, and 2011, respectively. He is currently a Part-time Lecturer at J. F. Oberlin University and Toyo University. His main research interests are applications of graph grammars, and graph drawing.

**Tetsuro Nishino** graduated from department of Mathematics, Waseda University, and continued his research and obtained a D.Sc. in Mathematics in 1991. He was a researcher at Tokyo Research Laboratory, IBM Japan during 1984-1987. He was an Associate Professor at School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku during 1992-1994. In 1994, he joined at Department of Communications and Systems Engineering, The University of Electro-Communications as an Associate Professor and in 2006 he became a Professor. He received The Funai Information Technology Prize (2003) and IBM Faculty Award (2008).

**Takeo Yaku** received his M.Sc. and D.Sc. degrees from Waseda University in 1972 and 1977, respectively. He has been a Professor at the Department of Computer Science and System Analysis of Nihon University since 1992. His research interests include software visualization, human interface, graph languages, and graph algorithms. He is a member of IEICE Japan, IEEE Computer Society, and ACM.

**Kensei Tsuchida** received M.S. and D.S. degrees in mathematics from Waseda University in 1984 and 1994, respectively. He was a member of the Software Engineering Development Laboratory, NEC Corporation in 1984-1990. From 1990 to 1992, he was a Research Associate of the Department of Industrial Engineering and Management at Kanagawa University. In 1992 he joined Toyo University, where he was an Instructor until 1995 and an associate professor from 1995 to 2002, and a Professor from 2002 to 2009 at the Department of Information and Computer Sciences, and since 2009 he has been a Professor of Faculty of Information Sciences and Arts. He was a Visiting Associate Professor of the Department of Computer Science at Oregon State University from 1997 to 1998. His research interests include software visualization, human interface, graph languages, and graph algorithms. He is a member of IPSJ, IEICE Japan and IEEE Computer Society.
Analysis of Software Fault Prediction Models Using Machine Learning Techniques

Golnoush Abaei*, Ali Selamat*
University Technology Malaysia

Abstract

One part of the Software Engineering newly interest are software quality assurance activities such as testing, verification and validation, fault tolerance and fault prediction. When there is not sufficient time and budget for complete testing of an entire system, a professional manager can use predictors for software quality in order to focus the testing on parts of the system that seem more defect prone. Attempt to identify and use high performance fault prediction algorithms based on distinct machine learning and statistical techniques followed by defining the different factors that could have influence on these predictions is one of researcher’s challenges for so long. In this paper, software fault prediction has been studied based on different machine learning techniques such as decision trees, decision tables, random forest, neural network, naïve bayes and distinctive artificial immune systems classifiers. According to this study, random forest performs best on small as well as big datasets and AIRSParallel is the finest algorithm in AIS category regardless of dataset size.

Keywords: Software Fault Prediction, Machine Learning Techniques, AIRSParallel, Random Forest

1. Introduction

As today’s software grow rapidly in size and complexity, the prediction of software reliability plays a crucial role in a software development process [1]. Software fault is an error situation of the software system, which is caused by explicit and potential violation of security policies at runtime because of wrong specifying and inappropriate development of configuration [3]. Defective modules pose a considerable risk by decreasing customer satisfaction and by increasing development and maintenance cost [2].

Identifying faults and defects in any software projects is a difficult work especially when project sizes grow, this task becomes expensive with sophisticated testing and evaluation mechanisms. On the other hand, measuring software in a continuous and disciplined manner brings many advantages such as accurate estimation of project costs and schedules, and improving product and process qualities [9].

When budget does not allow for full testing verification of an whole system, software managers can use predictors for software quality to focus the testing on parts of the system that seem defect prone and the defect prone trouble spots can then be examined in more detail by model checking, intensive testing and so on. Software fault prediction approaches use previous software metrics and fault data to predict fault-prone modules for the next release of software. According to [13] one member of the review team may inspect 8-20 lines of code per minute and all the member of a team consisting of 4-6 members performs this process. Therefore, software fault prediction approaches are much more cost effective to detect software faults compared to software reviews [9].

Prediction of fault prone software components is one of the most researched problems in software engineering [1]. Although many statistical and machine learning techniques have been proposed so far, but there is no agreement on the methodology to select the best model. Choosing the “best” model among many available models involves performance assessment and detailed review, but these comparisons are not simple due to many performance evaluation metrics. According to [13] the space of “best” predictors is brittle; minor change in a data can make different attributes appear most useful for defect prediction and some models more acceptable than others. In addition a modeling methodology is good if it is able to perform well on all data sets, or at least most of them.

A study by the National Institute of Standards & Technology [29] found that “the national annual cost of inadequate infrastructure for software testing is estimated to range from $22.2 to $59.5 billion”. Testing goals could be, faults identifications, so we should find out, what fault caused the failure? Then we should search for the remedy, how we can fix the error, or how we can delete the fault. In fact, Data mining and working on different types of machine learning technique to retrieve reliable predictions becomes a very interesting and popular topic recently in any aspects of industry. Building a fault prediction model becomes many researchers interest these past few years.

In this paper, the experiments have been done on three NASA datasets with different population size by applying Correlation Feature Selection (CFS) techniques. The predictability accuracy based on method-level metrics investigated in this paper. Table 1 shows the research done

---

* Intelligent Software Engineering Laboratory,
Faculty of Computer Science & Information Systems
University Technology Malaysia
golnoosh.abaee@gmail.com, aselamat@utm.my
in this study. Three different performance evaluation metrics were used together, Area Under receiver operating Characteristic Curve (AUC), Probability of Detection (PD), and Probability of False alarm (PF), to give more reliable prediction analysis. Although we calculated accuracy along with above metrics, it is only for sake of comparison.

Table 1. The studies done in this paper

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M = 10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>All = 21</td>
<td># 21 method-level</td>
</tr>
<tr>
<td>3</td>
<td>DATAS = (CM1 KC1 JM1)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>FST = (CFS)</td>
<td># feature selection techniques</td>
</tr>
<tr>
<td>5</td>
<td>LEARNERS = (J48, RF, NB, NN (back propagation), Decision Table, AIRS1, AIRS2, AIRSParallel, Immunos1, Immunos2, Immunos99, CLONALG, CSCA)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>for data in DATAS</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>for fst in FST</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>data’ = fst(data)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>for i in 1 to M</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>tests = bin[i]</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>trainingdata = data’ - tests</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>for learners in LEARNERS</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>METHOD = (cfs learner)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Predictor = learner (trainingdata)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>RESULT (METHOD) = apply predictors to test</td>
<td></td>
</tr>
</tbody>
</table>

The rest of the paper is structured as follows. The following section presents related work; Methodologies are described in section 3. Section 4 discusses the experimental description follows by results and comparison in section 5.

2. Related Works

According to a survey carried out by the Standish Group, an average software project exceeded its budget by 90 percent and its schedule by 222 percent [8]. This survey took place in mid 90s and contained data from about 8000 projects. These statistics show the importance of measuring the software early in its life cycle and taking the necessary precautions before these results come out. According to Catal [7], software fault prediction became one of the noteworthy research topics since 1990, and the number of research papers almost doubled until year 2009. Different techniques were used for software fault prediction such as genetic programming [10], decision trees [11] neural network [12], naïve bayes [13], case-based reasoning [14], fuzzy logic [15] and the artificial immune recognition system algorithms in [16, 17, 18]. Koru and Liu [21] applied the J48, K-Star and random forest algorithms on public NASA datasets in order to construct fault prediction model based on 21 method-level, which was converted into 84 class-level metrics, they used F-measures as an evaluation performance metrics. Menzies et al. [19] have conducted an experiment with the help of several data mining algorithms on again public NASA datasets using method-level metrics and evaluated the results by Probability of Detection (PD), Probability of False alarm (PF) and balance parameter metrics. They used log-transformation with Info-Gain filters before applying the algorithms and they claimed that fault prediction using naïve bayes performed better than the J48 algorithm. Menzies et al. [22] argued that since some models with low precision performed well, using it as a reliable parameter for performance evaluation is not good. Shafi et al. [23] used two other datasets from PROMISE repository, JEditData and AR3, they applied 30 different techniques on them, and showed that classification via regression and Locally Weighted Learning (LWL) are better than the other techniques, they chose precision, recall and accuracy as an evaluation performance metrics. Catal and Diri [28] used some machine learning techniques such as random forest and artificial immune recognition system with five NASA dataset; they evaluated the results with accuracy and Area Under receiver operating Characteristic Curves (AUC), Turhan and Bener used Probability of Detection (PD), Probability of False alarm (PF) and balance parameter [24], the results indicate that independence assumption in naïve bayes algorithm is not harmful with Principal Component Analysis (PCA) pre-processing. Alsmadi and Najadat [25] developed the prediction algorithm based on studying statistics of the whole dataset and each attributes, they proposed a technique to evaluate the correlation between numerical values and categorical variables of fault prone dataset in order to automatically predict faulty modules based on attribute's values. Parvinder S. Sandhu et al. [26] claimed that, the prediction of different level of severity or impact of faults in object oriented software systems with noise can be done satisfactory by using density-based spatial clustering; they used KC1 from NASA public dataset. Burak Turhan et al. [27] analyzed 25 project of the largest GSM operator in Turkey, Turkcell, to predict defect before the testing phase, they used a defect prediction model that is based on static code attributes such as lines of code, Halstead and McCabe. They suggested that at least 70% of the defects could be detected by inspecting only 6% of the code using a naïve bayes model and 3% of the code using Call Graph Based Ranking (CGBR) framework.

3. Methodologies

In this section, some of software prediction algorithms are reviewed briefly with some of their features.

3.1. Artificial Immune System

Artificial immune system first introduced as a branch of AI in 1991. Artificial immune system is a technique to the scene of biological inspired computation and artificial intelligence based on the metaphor and abstraction from theoretical and empirical knowledge of the mammalian immune system. The immune system is known to be
distributed in terms of control, parallel in terms of operation, and adaptive in terms of functions, all of which are features desirable for solving complex or intractable problems faced in the field of artificial intelligence [30]. In the artificial immune system, the components are artificial cells or agents which flow through a computer network and process several tasks in order to identify and prevent attacks from intrusions. There are several classifiers based on AIS paradigm available, some of them are briefly described as follows:

AIRS (Artificial Intelligence Recognition System) is one of the first AIS techniques designed specifically and applied to classification problems. It has five steps: Initialization, antigen training, competition for limited resources, memory cell selection, and classification [27, 29]. First, the dataset is normalized. Then based on the Euclidean formula, distances between antigens are calculated which is called affinity. Antibodies that are present in the memory pool are stimulated with a specific infected antigen, and the stimulated value is assigned to each cell. The cell that has the highest stimulation value is chosen as the finest memory cell. Afterwards, the best match from the memory pool is selected and added to the ARB (Artificial Recognition Ball) pool. This pool contains both antigen and antibodies with their stimulation and some other information related to them. Next the numbers of clones are calculated and cloning starts. These clones also are added to the ARB pool. After that, competition for the finite resources begins. Again, ARB pool is stimulated with the antigens, and limited resources are assigned to them based on derived stimulation values. This is a recursive task until the stopping condition happens, that is, if the stimulation level between ARB and antigen is less than the affinity threshold, it stops otherwise; it goes on. After that, the ARB with the highest stimulation value is selected as a candidate to be a memory cell. The stimulation value is compared to the best previous matching value and if the new value is better, it will be replaced by the old one.

CLONALG (CLONal selection ALGorithm) is another AIS classifier, which is inspired by the clonal selection theory of acquired immunity, previously known as CSA. The theory specifies that the organism have a pre-existing pool of heterogeneous antibodies that can recognize all antigens with some level of specificity [30]. A new clonal selection inspired classification algorithm is called Clonal Selection Classification Algorithm (CSCA). The goal here is to develop a memory pool contains best antigen matching antibodies that represent a solution to engineering problems. Most of the issues that described in AIS are close to biological metaphor, which is useful in architectural aspect, but the goal for Immunos-81 is to reduce this part and focus on the practical application. This algorithm has three main steps, Initialization, Training, and Classification [31]. There are two basic implementations for Immunos-81 that are known as naïve immuno algorithms, which are also called, Immunos-1 and Immunos-2. There is no data reduction in Immunos1, and it’s similar to the k-nearest neighbors. The primary difference is obviously that the training population is partitioned and k is set to one for each partition; multiple problem support can be provided with simpler mechanism that uses classifier for each problem, and each classifier has its own management mechanism. The Immunos2 implementation is the same as Immunos1 only it seeks to provide some form of primary generalization via data reduction, so the closer representation to basic Immunos [31].

3.2. Back Propagation, Neural Network

Back Propagation, which is used in supervised learning, is one of the famous artificial neural network methods. The neural network has a different level of layers, when they receive inputs, the neuron in the first layer received them, and it applies the linear combiner and the activation function to the inputs and produces the Output. This output, could be the input for the specific neuron in the following layer. Therefore, the next layer will feed forward the data, to the succeeding layer. And so on, until the last layer is reached. For example, when we had a data from yesterday, and we want to do some prediction about today’s data, we know what will be the expected output. Then the real output is compared with the desired output and error rate is calculated. In order to do that we may do some input adjustments to improve the error rate. Once we decide what adjustment we should apply, we can do the reverse work (back propagate the changes to the previous layers of the network). In order to reduce the number of errors. Adjustment will change weights of the input nodes of the neurons in the output layer. Generally, there are two sets of data, one for testing the data, which is called training set and the other for checking to see whether the learning and prediction are correct or not, the second one is called testing set. After the input pattern was presented to the network and processed by all layers, we have errors that can be used for adjusting the network.

3.3. Naïve Bayes

It is probabilistic classifier, which has this idea that every attributes that are used for identifying any class are independent and there is no correlation between them. This model train and construct a predictor by analyzing historical data of any software modules based on predictors, and then it will decide whether new modules are defected or not. It can be applied to many different complex problems, and it needs small amount of training data in order to do variable estimation, which is used for classification. It is a simplified version of Bayes formula.
that identifies which class a novel instance belongs. Although this method is suited when the dimensionality of the data is high, but the na"\i ve Bayesian classifier’s predictive performance can be adversely affected by the presence of redundant attributes in the training data that could perform better if the redundant features are removed.

3.4. Decision Tree, C4.5, J48

It is an algorithm that summaries training data in the form of a decision tree. Features are nodes, associated values are shown in branches, and classes are leaves. For classifying any new instance, one simply examines the features tested at the nodes of the tree and follows the branches corresponding to their observed values in the instance. Upon reaching a leaf, the process terminates, and the class at the leaf is assigned to the instance. It can accept both numerical and nominal values. Decision Tree is a robust algorithm, which can produce accurate and fast results and summarize the knowledge in an understandable structure. It does not have any problem with redundant data, but it works better and understandable when the size of the tree is smaller.

C4.5 builds the decision tree that could be used for classification as a statistical classifier. C4.5 is an extension of Quinlan’s earlier ID3 algorithm, which made a number of improvements to it. It can handle both discrete and continuous values and could work with missing values in training data. J48 is an open-source java implementation, which could be used in simulators in order to run C4.5. J48 has two different pruning methods [32, 33]. Subtree replacement, here nodes in a decision tree may replace with a leaf and reduce the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. The second approach is Subtree raising, a node may moved upwards towards the root of the tree and replace other nodes along the way. It has a negligible effect on decision tree models and can be somewhat computationally complex.

3.5. Random Forest

Random Forests consists of more than one decision trees. Suppose there is N number of training cases and M number of variables in the classifiers. Assume that, there is m number of input variables that could be used to determine the decision at any node of the tree. Training set is chosen n times with replacement from all N available training cases, the rests of the cases are used to estimate the error of the tree, by predicting their classes. For each node of the tree, randomly choose m variables on which to base the decision at that node [34]. The best match is calculated based on these m variables in the training set. No pruning is done here. Whenever the prediction needs to be done, sample is pushed down to the tree with a training sample label in the terminal node it ends up with. The task is repeated through all trees in the ensemble, the average vote is calculated and reported for all trees and is called random forest prediction. These algorithms work with high data dimensionality; it means, it does not need any data reduction. It could identify what are the important variables in the classification; it also can estimate missing data. It is used as one of the most accurate classifiers, but it has some drawbacks also such as, according to [35]; they may over fit for some datasets with noisy classification or regression tasks.

4. Experimental Description

One of the main challenges in this area is how to get the data. Some researchers like in [6] use the data provided by specific company, so the results are not fully trustable. The work will be more valuable and accurate when the public datasets are used. After 2005, with the help of PROMISE repository, the usage of public datasets reached to half [7]. Many researchers [15,16,17,18,19,21,22,23,26] used NASA datasets. Since we wanted to show the comparative study to their research, we also selected the dataset from PROMISE repository of NASA [20]. The largest dataset is JM1 with 10885 rows. It belongs to real time predictive ground system project, 19% of these data are defected. The smallest dataset is CM1, which has only 498 modules, and 10% of the data are defected, CM1 belong to NASA spacecraft instrument project. KC1 belongs to storage management project for receiving and processing ground data; it has 2109 numbers of rows and 15% of the data are defected. The above datasets are preferred because they all have same 21 attributes, and they are distinctive in number of modules (rows) and defect rates.

All 21 static metrics are listed as follows. McCabe’s metrics: cyclomatic complexity (v(g)), essential complexity (ev(g)), design complexity (iv(g)). Halstead’s metrics: total operators + operands (N), volume (V), program length (L), difficulty (D), intelligence (I), effort (E), delivered bugs (B), time estimator (T), line count (IOCode), count of lines of comments (IOComment), count of blank lines (IOBlank). Rest of metrics are: lines of code and comments (IOCodeAndIOComment), unique operators (uniq_op), unique operands (uniq_opnd), total operators (total_op), total operand (total_opnd), branch count of the flow graph (branchCount), line count of code (LOC).

All experiments have been done in WEKA, which is open-source software and implemented in JAVA; it developed in university of Waikato, and it is used for machine learning studies [4]. 10-fold cross-validation is used for experimenting.
Evaluation metrics are accuracy, Probability of Detection (PD), Probability of False alarm (PF) and Area Under receiver operating Characteristic Curve (AUC). PD or recall is the probability of the correct classification of faulty modules. PF is a portion of fault-free modules that classified erroneously. Accuracy is overall probability of classifying module correctly. Since the accuracy and precision parameters are not completely reliable, Area Under receiver operating Characteristic Curve (AUC) values is used for benchmarking especially when the dataset is unbalanced. Table 2, and Equation 1, 2 and 3 shows the calculation of this metrics. FP (False Positive), classifies non-faulty units as faulty. FN (False Negative), classifies faulty units as non-faulty. TP (True Positive), modules correctly classified as faulty. TN (True Negative), modules correctly classified as non-faulty. The best classifier has high TP rate and low FP rate [25].

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad \text{Equation (1)}
\]

\[
\text{Recall (PD)} = \frac{TP}{TP + FN} \quad \text{Equation (2)}
\]

\[
PF = \frac{FP}{FP + TN} \quad \text{Equation (3)}
\]

In this experiment all redundant attributes should be eliminated, so if any features’ prediction ability could be covered by another, then, it can be removed therefore correlation-based feature selection was applied on the data set before they are used for the experiment. According to research on feature selection experiments, irrelevant features should be removed along with redundant information. A feature is said to be redundant if one or more of the other feature are highly correlated with it [36]. CFS computes a heuristic measure of the “merit” of a feature subset from pair-wise feature correlations and a formula adapted from test theory. Heuristic search is used to traverse the space of feature subsets in reasonable time; the subset with the highest merit found during the search is reported. This method also needs descretizing the continuous features. In fact, CFS is categorized as a filter. According to [37], feature \( V_i \) is said to be relevant. If there exists some \( v_i \) and \( c \) for which \( p(V_i = v_i) > 0 \) such that in Equation 4.

\[
p(C = c | V_i = v_i) = p(C = c) \quad \text{Equation (4)}
\]

5. Results and Comparison

In this paper, we built fault prediction models based on different machine learning techniques after we applied correlation-based feature selection (CFS) on each dataset. Since the accuracy rate is not the reliable metrics for performance evaluation, three other metrics were used. After applying CFS (best first) on three dataset, some of the attributes were eliminated. The remaining attributes form JM1, KC1 and CM1 after applying CFS (best first) are listed below. Results in Table 3, Figure 1, Figure 2 and Figure 3 show the comparison between different evaluation metrics for this study.

**Table 2. Confusion Matrix**

<table>
<thead>
<tr>
<th></th>
<th>No (Actual)</th>
<th>Yes (Predicted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No (Actual)</td>
<td>TN</td>
<td>FP</td>
</tr>
<tr>
<td>Yes (Actual)</td>
<td>FN</td>
<td>TP</td>
</tr>
</tbody>
</table>

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \quad \text{Equation (1)}
\]

\[
\text{Recall (PD)} = \frac{TP}{TP + FN} \quad \text{Equation (2)}
\]

\[
PF = \frac{FP}{FP + TN} \quad \text{Equation (3)}
\]

According to Figure 1, JM1 results, random forest, and decision table, both have high AUC value but with considering PD and PF values, random forest has better combination of low PF and high PD. Among AIS algorithms, although Immunos1 has higher AUC and PD compared to the others but the value of PF is high and not acceptable. Therefore, AIRSParallel has a better performance. Figure 2, shows KC1 results. Random forest, naïve bayes and decision tree have higher AUC value, but if we consider PF and PD as well, naïve bayes is a finest algorithm. In AIS algorithms, both AIRSParallel and CSCA accomplished preferred results. According to Figure 3, random forest follows by naïve bayes have highest AUC values for the smallest dataset, CM1, but naïve bayes performed better when PD and PF values are also considered. AIRS1 has a better performance compared to the other AIS algorithms.

From Table 3, we can conclude that, if we consider high AUC and PD along with low PF as a well-performed benchmark, random forest performs best on both small and big datasets. With more detailed look, random forest and decision table are the best when the size of the dataset is not that very small, of course naïve bayes is the safe algorithm as it performs well among the others for all three datasets. In AIS classifiers, AIRSParallel is the finest algorithms regardless of dataset size, and CSCA works well when the size of dataset is not small. It seems that if the algorithms perform best in the big datasets, there is a high chance that could performs well for smaller datasets as well.
Table 3. Evaluation results based on each algorithm

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>JM1</th>
<th>KC1</th>
<th>CM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision tree, J48</td>
<td>81.01</td>
<td>84.68</td>
<td>89.31</td>
</tr>
<tr>
<td>AUC</td>
<td>0.664</td>
<td>0.705</td>
<td>0.542</td>
</tr>
<tr>
<td>PD</td>
<td>0.148</td>
<td>0.175</td>
<td>0.000</td>
</tr>
<tr>
<td>PF</td>
<td>0.031</td>
<td>0.030</td>
<td>0.009</td>
</tr>
<tr>
<td>Random forest</td>
<td>80.28</td>
<td>84.83</td>
<td>88.15</td>
</tr>
<tr>
<td>AUC</td>
<td>0.710</td>
<td>0.786</td>
<td>0.615</td>
</tr>
<tr>
<td>PD</td>
<td>0.243</td>
<td>0.282</td>
<td>0.102</td>
</tr>
<tr>
<td>PF</td>
<td>0.063</td>
<td>0.048</td>
<td>0.033</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>80.41</td>
<td>82.41</td>
<td>86.55</td>
</tr>
<tr>
<td>AUC</td>
<td>0.665</td>
<td>0.785</td>
<td>0.691</td>
</tr>
<tr>
<td>PD</td>
<td>0.223</td>
<td>0.365</td>
<td>0.306</td>
</tr>
<tr>
<td>PF</td>
<td>0.056</td>
<td>0.092</td>
<td>0.073</td>
</tr>
<tr>
<td>Back Propagation, NN</td>
<td>80.65</td>
<td>84.54</td>
<td>90.16</td>
</tr>
<tr>
<td>AUC</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>PD</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>PF</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Decision Table</td>
<td>80.81</td>
<td>84.92</td>
<td>89.16</td>
</tr>
<tr>
<td>AUC</td>
<td>0.701</td>
<td>0.781</td>
<td>0.626</td>
</tr>
<tr>
<td>PD</td>
<td>0.108</td>
<td>0.178</td>
<td>0.000</td>
</tr>
<tr>
<td>PF</td>
<td>0.024</td>
<td>0.028</td>
<td>0.011</td>
</tr>
<tr>
<td>AIRS1</td>
<td>66.76</td>
<td>76.34</td>
<td>84.54</td>
</tr>
<tr>
<td>AUC</td>
<td>0.567</td>
<td>0.602</td>
<td>0.569</td>
</tr>
<tr>
<td>PD</td>
<td>0.402</td>
<td>0.368</td>
<td>0.224</td>
</tr>
<tr>
<td>PF</td>
<td>0.269</td>
<td>0.184</td>
<td>0.087</td>
</tr>
<tr>
<td>AIRS2</td>
<td>73.36</td>
<td>77.34</td>
<td>82.53</td>
</tr>
<tr>
<td>AUC</td>
<td>0.565</td>
<td>0.591</td>
<td>0.530</td>
</tr>
<tr>
<td>PD</td>
<td>0.290</td>
<td>0.328</td>
<td>0.163</td>
</tr>
<tr>
<td>PF</td>
<td>0.160</td>
<td>0.145</td>
<td>0.102</td>
</tr>
<tr>
<td>AIRS Parallel</td>
<td>70.17</td>
<td>79.47</td>
<td>86.14</td>
</tr>
<tr>
<td>AUC</td>
<td>0.564</td>
<td>0.588</td>
<td>0.488</td>
</tr>
<tr>
<td>PD</td>
<td>0.301</td>
<td>0.301</td>
<td>0.041</td>
</tr>
<tr>
<td>PF</td>
<td>0.173</td>
<td>0.125</td>
<td>0.065</td>
</tr>
<tr>
<td>Immunos1</td>
<td>59.99</td>
<td>49.98</td>
<td>69.88</td>
</tr>
<tr>
<td>AUC</td>
<td>0.600</td>
<td>0.678</td>
<td>0.697</td>
</tr>
<tr>
<td>PD</td>
<td>0.600</td>
<td>0.936</td>
<td>0.694</td>
</tr>
<tr>
<td>PF</td>
<td>0.400</td>
<td>0.508</td>
<td>0.301</td>
</tr>
<tr>
<td>Immunos2</td>
<td>80.65</td>
<td>80.23</td>
<td>90.16</td>
</tr>
<tr>
<td>AUC</td>
<td>0.500</td>
<td>0.491</td>
<td>0.500</td>
</tr>
<tr>
<td>PD</td>
<td>0.000</td>
<td>0.040</td>
<td>0.000</td>
</tr>
<tr>
<td>PF</td>
<td>0.000</td>
<td>0.058</td>
<td>0.000</td>
</tr>
<tr>
<td>Immunos99</td>
<td>65.02</td>
<td>62.21</td>
<td>76.51</td>
</tr>
<tr>
<td>AUC</td>
<td>0.594</td>
<td>0.705</td>
<td>0.679</td>
</tr>
<tr>
<td>PD</td>
<td>0.502</td>
<td>0.825</td>
<td>0.571</td>
</tr>
<tr>
<td>PF</td>
<td>0.314</td>
<td>0.415</td>
<td>0.214</td>
</tr>
<tr>
<td>CLONALG</td>
<td>72.92</td>
<td>79.28</td>
<td>87.95</td>
</tr>
<tr>
<td>AUC</td>
<td>0.512</td>
<td>0.522</td>
<td>0.497</td>
</tr>
<tr>
<td>PD</td>
<td>0.159</td>
<td>0.129</td>
<td>0.020</td>
</tr>
<tr>
<td>PF</td>
<td>0.134</td>
<td>0.086</td>
<td>0.027</td>
</tr>
<tr>
<td>CSCA</td>
<td>79.55</td>
<td>83.21</td>
<td>87.75</td>
</tr>
<tr>
<td>AUC</td>
<td>0.575</td>
<td>0.590</td>
<td>0.505</td>
</tr>
<tr>
<td>PD</td>
<td>0.217</td>
<td>0.239</td>
<td>0.041</td>
</tr>
<tr>
<td>PF</td>
<td>0.066</td>
<td>0.059</td>
<td>0.031</td>
</tr>
</tbody>
</table>

Figure 1. Comparison between different evaluation metrics for project JM1

Figure 2. Comparison between different evaluation metrics for project KC1

Figure 3. Comparison between different evaluation metrics for project CM1
References


[25] Izzat Alsmadi and Hassan Najadat, “Evaluating the change of software fault behavior with dataset attributes based on categorical correlation”, Advances in Engineering Software


Golnoush Abaei received her B.E. in Software engineering from Islamic Azad University, Central Branch, Tehran, Iran and the master’s degree, MCA (Master of Computer Application) from University of Mysore, Mysore, Karnataka state, India. Now she is a Ph.D candidate in University Technology Malaysia. She had worked in several software companies as a programmer, project manager and system analyst for almost 4 years before she persuade her master’s degree. She worked in faculty of software engineering in Shahab e Danesh institute of higher education in Iran and also she was a lecturer in Islamic Azad University, Roodehen Branch. Her research interests are software testing, software defect detection and prediction and data mining.

Ali Selamat has received a B.Sc. in IT from Teesside University, U.K. and M.Sc. in Distributed Multimedia Interactive Systems from Lancaster University, U.K. in 1997 and 1998, respectively. He has received a Ph.D. degree from Osaka Prefecture University, Japan in 2003. Currently, he is a professor and Research Dean of K-Economy Research Alliance in UTM. His research interests include software engineering, software agents, web engineering, information retrievals, genetic algorithms, neural networks and soft computing.
Practice and Evaluation of Open Source Software Development Education 
Based on the UEC Software Repository

Takaaki Goto†
The University of Electro-Communications, Japan

Kensei Tsuchida†
Toyo University, Japan

Tetsuro Nishino‡‡
The University of Electro-Communications, Japan

Abstract

Many efforts and much research has been done on practical software development education over the years. Practical software development education mainly targets project based learning, therefore it consists of a small group of a few members developing systems from requirements. The goal of project based learning is to learn the development style that is actually taking place in a company. On the other hand, Open Source Software (OSS) is a kind of published software that we can use, and also modify its source code, subject to license. Each type of OSS has its own community consisting of some members to develop the software. However, there are not very many Japanese OSS developers, therefore a practical education program for OSS development is needed. We started with practical software development education in 2009, and we established the OSS development subject in graduate school in 2011. In this subject, students develop OSS software along with their ideas or presented material (including development themes) for publishing the software on the UEC (The University of Electro-Communications) software repository. The UEC software repository is the database that stores software developed in our university. Students can learn the total process of software development through OSS development. In this paper, we provide an outline and describe the current state of the UEC Software Repository, and report the details and evaluation of our OSS development subject.

Keywords: software repository, software development education, open source software, practical education

1. Introduction

Many efforts and much research has been done on practical software development education over the last few years. Advances in telecommunications technology are turning the software industry into one of the key industries of the 21st century. In Japan, today’s information systems graduates have not been provided with a sufficient education with regard to the knowledge, techniques, and experience needed for the usual software development process. This includes the actual software development businesses from project planning through execution and management (requirements analysis, development, implementation, testing, etc.), and the lack of the basic skills for handling actual business situations.

On the other hand, Open Source Software (OSS) is a kind of software where we can use published software and also modify its source code, subject to license. Each OSS has its own community consisting of members to develop software, however there are not very many Japanese OSS developers yet, therefore an education program for OSS developers is needed.

One of the repositories that can handle software is SourceForge [1]. SourceForge is a repository used for open-source software that not only provides facilities for publishing computer programs, but also supports software development by offering additional features for such as managing version control systems and operating bulletin boards. Github also supports collaborative software development [2].

In this paper, we provide an outline and describe the current state of the UEC software repository, and report the details and evaluation of our OSS development subject. This paper is organized as follows. Section 2 outlines the Education Program for Practical Software Development in the University of Electro-Communications. In Section 3, we review the UEC software repository, which is a software database. Section 4 explains the practice and evaluation of the foundations of open source software development. And finally, we conclude in Section 5.

2. Education Program for Practical Software Development in the University of Electro-Communications

We are working on a practical form of software development education that provides students with a more rounded set of practical skills and self-motivated qualities in software development. By developing teaching materials for a self-motivated practical education, centered on open-source software development and implementing an education program that incorporates these materials, our goal is to provide practical education using the UEC software repository as an autonomous practical educational resource in order to cultivate highly creative developers with excellent research and development skills.
2.1 Development of Teaching Materials

During the development of the teaching materials for the practical software development course, we plan to design courses on software engineering, infrastructure system architecture, and OSS development. Some of the materials have already been designed, and we are continuously developing the lecture plans.

2.2 Fundamentals of Practical Software Development

The "Fundamentals of Practical Software Development" module is concerned with the details of software engineering. This module is based on the results of the module "Introduction to Practical Software Development I" (Introduction to Software Engineering) which was conducted in the 2009-10 academic year. This module was offered in the 2010-11 academic year as the interdisciplinary specialist module "Fundamentals of Practical Software Development" at the graduate school. For the Introduction to Practical Software Development, in addition to an overview of software engineering, a larger proportion of time was set aside for students to engage in proactive tasks such as discussing a chosen theme or developing Web applications. This results in a course with greater practical content that more closely resembles the scene of actual software development. Since the course includes Java software development exercises, it is necessary for students to have previously gained the requisite knowledge of "Basic Java" relating to the basic details, such as the classes and inheritance, equivalent to an introductory course in Java. They should also have knowledge of "Applied Java" relating to the construction of Web applications using server-side Java.

2.3 Foundations of IT System Software

In the "Foundations of IT System Software" module, students will perform hands-on learning of infrastructure systems such as operating systems, networks, and databases (Linux, Tomcat, PostgreSQL etc.) in order to gain a thorough understanding of the workings of the UEC software repository. Students can also learn the working principle of infrastructure systems systematically, and they have to construct a server while considering the performance of its system.

2.4 Foundations of Open Source Software Development

In the "Foundations of Open Source Software Development" module, students will engage in PBL (Problem-Based Learning) to develop OSS for inclusion in the UEC software repository. We obtained several requests about development themes from our campus and neighboring school. Tools such as a matrix calculator, ledger sheet printer, and a supporting tool of color amblyopia, are developed by students. In development phases, students can develop systems with their users, therefore it is a valuable experience for students. All of the OSS produced in this module will be submitted to the repository.

3. The UEC Software Repository

3.1 Overall Concept

Our efforts are centered on the UEC software repository [3] and a course in practical software development. In the practical software development course, students receive lectures on software engineering, system infrastructure, and OSS development. Upon completion of these lectures, they are granted permission to make submissions to the UEC software repository. After receiving permission to submit software to the UEC software repository, the students will continue with their PhD or MSc studies, and publish their research results in the repository as open-source software. Figure 1 illustrates an outline of our education program for practical software development.

By making practical use of the system development knowledge gained in the practical software development course, students are able to experience the software development process for themselves in their research and development work, and by submitting the resulting software to the repository. By centering the education of graduate students around the UEC software repository and a practical software development course, we will instill our students with self-motivation and practical abilities.

At the University of Electro-Communications, a great deal of software is written every year through the course of student research and educational activities. Normally, most of this software is managed and stored on servers in the research laboratories, but since the software is generally lacking in documentation and is not developed with third-party use in mind, it tends to be used only by the original developers. Once the developers have graduated, there are very few opportunities for this software to be reused. As one
approach to addressing these problems, we are working on the construction of a UEC software repository. The UEC software repository is a database for the centralized in-house management of software developed at the university. Our goal is to make it possible for ordinary users to download and use software stored in this repository, by accessing it through site searches. The repository system has the following features:

1. Software registration, search and download functions
2. Research paper registration, search and download functions
3. Download ranking functions
4. User management functions
5. Development support functions

Figure 2 illustrates the concept of the UEC software repository.

Software developed by students and teachers is uploaded to and registered in the repository. When registering software in the repository, the person registering the software can choose to either retain the software copyright or to transfer it over to the university. Software published in the UEC software repository is not only available for download by students and teachers, but is also made publicly available for downloading by ordinary users, businesses, researchers, and the like.

3.2 User Registration

This repository supports three types of users with different privileges:

1. Software project leader
2. Software users
3. System administrators

Table 1 lists the features available to each type of user. Software project leaders are permitted to submit software to the repository. To qualify as a software project leader, a user must have studied for and completed a graduate course in the practical software development course, which is described above. Software project leaders are also able to search and download research papers.

Software users are permitted to search and download the software and research papers published in the repository. Software users need no particular qualifications, and only need to complete the software user registration process in the repository.

<table>
<thead>
<tr>
<th>Table 1. Features available to each user type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software project leader</td>
</tr>
<tr>
<td>Software users</td>
</tr>
<tr>
<td>System administrators</td>
</tr>
</tbody>
</table>

System administrators are permitted to check and approve software that has been submitted to the repository, and to perform user administration. They are also permitted to register and manage research papers such as technical reports.

As an example, Figure 3 illustrates the registration procedure for software users. To register as a software user, the user first accesses the repository and clicks the user registration link. The user is then shown an outline document that describes the purpose of the repository, and a simple copyright statement. When the user clicks to accept this information and move to the next page, a UEC software repository user agreement is displayed. At the bottom of this user license, the user is asked to mark a check box to confirm his or her acceptance of the items of particular importance in the user agreement, such as the member eligibility conditions and prohibited actions. Next, the user provides a login ID and password, together with details such as the user’s name and email address. When the user has finished registering his or her user information in the repository, the system sends out a confirmation email containing a link that the user must click to complete the registration process. A similar procedure is used for the registration of software project leaders.
3.3 Software Registration

The registration of software in the repository can only be performed by users that are eligible to act as software project leaders. When registering software in the repository, software project leaders are able to choose either of the following copyright conditions for software that they have developed:

1. Transfer copyright to the university
2. Keep copyright with project leader

It was originally envisaged that all copyrights on software registered in the repository would be transferred from the software project leader to the university at the time of registration, and that the university would then grant users the right to freely copy and adapt the software for their own purposes. However, when the teaching staff were surveyed to find software candidates for inclusion in the repository, we found many cases where they were unwilling to transfer copyright to the university. We therefore adopted a system where software project leaders can choose either to transfer their copyright to the university, or to retain their individual copyright without transferring it to the university.

![Figure 4. Software submission procedure](image)

Figure 4 illustrates the procedure for submitting software to the repository. First, the software project leader logs in to the repository. This brings up a software registration page where the project leader registers the details of the software. These details include the name of the software, its scope of use, its version number, a list of keywords, a summary, a description, and a list of co-authors. The project leader can then upload the software data and summary documentation, at which time the items relating to software registration in the user agreement are displayed once again, and the project leader must confirm these items by marking a checkbox before completing the upload process. When an upload is processed, the software specified in the registration procedure becomes provisionally registered in the repository.

After the software project leader has provisionally registered the software in the repository, the application details are checked by a repository system administrator. This includes checking the information submitted along with the software, and checking the software for viruses. If all is well, the work of registering the software is continued. The next step is the copyright transfer procedure, although this is unnecessary if the software copyright is being retained by the project leader. The system administrator then performs the software publication process to publish the software. When the software copyright is transferred to the university, a copyright transfer form is displayed in the page showing the details of this software in the software project leader’s user pages, from there it can be downloaded. The software project leader downloads the copyright transfer form from the repository, adds a signature and/or seal, and submits it to the university. Once the university has received the copyright transfer form from the software project leader, the system administrator issues a free use license agreement on the repository, and the software project leader will be able to download it from his or her
user pages. Finally, the system administrator performs the software publication process to publish the software.

The project leader is then given a project page of the software to collaborate with the coauthor. A project page has functions such as forums, and bug tracking. Project leaders can treat their project page as either public or private.

4. Practice and Evaluation of Foundations of Open Source Software Development

4.1 Objective of Practical Software Development Education Based on Open Source Software in the University

According to a report by Mitsubishi Research Institute, Inc [4], there were more than 5,000 OSS developers in Japan in 2004. They made up 2 percent of the total OSS developer numbers in the world. On the other hand, Shimizu et al. reports that "the ratio of OSS developers from the university is low in Japan, the university should construct a curriculum for programming education based on OSS" [5].

In order to increase the number of Japanese OSS developers (especially for Japanese university students), an OSS development education is needed in the university. Therefore, we designed a curriculum of OSS development education in the university. The goal of this method is to cultivate persons of talent who will be capable of developing and realizing their ideas while developing software. The best way for students to learn OSS development is to participate in the OSS development community. There are some difficulties with the participation in the activity of OSS development; (1) only a few students can take part in OSS community because of the student's development skills, (2) it is too short of a time period to treat actual OSS development in a subject, (3) it is hard to develop OSS from scratch in a subject.

For the above reasons, we developed materials for OSS development education. In particular, we prepared some themes, templates of software, and software specifications for each theme. These materials are simple, but reading source codes that were developed by other people is an important method to learn developing. Many software development educational programs in Japan treat PBL (Project Based Learning), while in our subject students develop their software independently.

Neighboring publicly run schools or public institutions have requested developing software for a purpose of educational support or office job support. However, it is often the case that they do not have much of a budget to develop such software, therefore efficient work supported by ICT (Information and Communication Technology) has not been realized yet. We focused attention on this factor and adopting demand by publicly run schools as one of the themes of software development in our subject area. Software developed in our subject will be published as OSS on the UEC software repository. By doing this, people involved in neighboring publicly run schools can download and use the software for free. In the future, we have a plan to develop or revise software on the repository by the students from the upper processes (i.e. from the hearing and requirement definition) based on OSS development.

4.2 Details of Foundations of Open Source Software Development

This subject targets learning the software development process from the upper process to the lower process based on open source software development. Students develop their software based on the waterfall model. The subject includes 15 classes in one semester as shown in Table 2.

<table>
<thead>
<tr>
<th>No.</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Guidance, introduction of development theme</td>
</tr>
<tr>
<td>2</td>
<td>Decision on software development theme, requirements analysis brief</td>
</tr>
<tr>
<td>3</td>
<td>Discussion for requirements analysis of each student, external design brief</td>
</tr>
<tr>
<td>4~6</td>
<td>Discussion for external design of each student</td>
</tr>
<tr>
<td>7</td>
<td>Interim report (presentation), internal design brief</td>
</tr>
<tr>
<td>8</td>
<td>Discussion for internal design of each student</td>
</tr>
<tr>
<td>9~12</td>
<td>Implementation (including discussion)</td>
</tr>
<tr>
<td>13</td>
<td>Testing</td>
</tr>
<tr>
<td>14</td>
<td>Accomplishment report</td>
</tr>
<tr>
<td>15</td>
<td>Documentation</td>
</tr>
</tbody>
</table>

In each lecture, a teacher shows an outline of the software development process, and then students report their progress from the preceding week to this week with their software documents. In the progress reports, teachers and students discuss the reports in order to refine the design and outputs (documents) of the students’ software.

4.3 Evaluations of Foundations of Open Source Software Development

In the final lecture of this course, we carried out a questionnaire survey to evaluate and improve our OSS development subject.

The Questionnaire includes the following four questions; (1) Answer the estimated numbers of weeks and actual achievement weeks for each development process (requirements analysis, external design, internal design, Implementation, Testing), (2) Explain the process that has the most difference between the estimated weeks and actual weeks, and consider solutions to this matter, (3) Choose one software development phase that is the most valuable for you, and explain the reason in detail, (4) Choose one software development phase that is the most difficult phase
for you, and explain the reason in detail. Table 3 shows one result of question (1).

In the requirements analysis and external design phases, there are no students who get delays. It is thought to be the aftermath of the identification of the requester and developer (same student). In this year’s case, three students, of the total eight students, chose their original development theme. The other students chose prepared material, however there was not enough time to have a hearing of software requirements from the real user. Therefore, the rest of the students chose prepared development themes. For this reason, each student plays both the role of requester and developer against their own software. This subject focuses on learning the whole development process among the individuals. It is allowed to play both the part of the requester and the developer. In the internal design phase, almost all students finished each design according to the plan.

In response to question (2), five students answered “implementation,” and three students answered "testing." Students who get delayed were observed in relatively large numbers in the implementation and testing phases because of the students’ implementation and testing skills.

When asked question (3), five students answered "requirements analysis" (Table 4). We presented samples of specifications and prototypes for some development themes developed in collaboration with industries and universities. These were effective for almost all students who had never experienced software design or writing documentation, thus far.

Four students answered "implementation" for question (4) (Table 5). The reasons for this answer include programming skills (for example, how to write codes in order to realize desired functions) or difficulty in source code reading.

Table 3. Answers for question (1)

<table>
<thead>
<tr>
<th>phase</th>
<th>result</th>
<th>Student A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements analysis</td>
<td>estimate</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>result</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>difference</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>External design</td>
<td>estimate</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>result</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>difference</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Internal design</td>
<td>estimate</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>result</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>difference</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Implementation</td>
<td>estimate</td>
<td>5</td>
<td>5.2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>result</td>
<td>7</td>
<td>4.5</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>difference</td>
<td>+2</td>
<td>-0.7</td>
<td>0</td>
<td>+2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>+4</td>
</tr>
<tr>
<td>Testing</td>
<td>estimate</td>
<td>1</td>
<td>5.4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>result</td>
<td>2</td>
<td>7.4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>difference</td>
<td>+1</td>
<td>+2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

(Number of weeks)

Table 4. Answers for question (3)

<table>
<thead>
<tr>
<th>Process</th>
<th>Count</th>
<th>Comment</th>
</tr>
</thead>
</table>
| Requirements analysis | 5     | ✓ Understanding the difference between the request specifications and requirement definition was important ★
|                     |       | ✓ Experience of considering software behavior and writing specifications was good ★
|                     |       | ✓ Learning how to write the requirement definition was beneficial ★
| Implementation      | 1     | ✓ This subject gave me an opportunity to try to use new program language and unknown server side programming ★
| External design     | 1     | ✓ Understanding the difference between external design and internal design was important ★
| Internal design     | 1     | ✓ Learning how to write the internal design specification was beneficial ★
Table 5. Answers for question (4)

<table>
<thead>
<tr>
<th>Process</th>
<th>Count</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>4</td>
<td>✓ Understanding the prototype system was difficult</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Java GUI programming was difficult because of no experience</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ It was hard to construct the development environment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Considering of bit operation was difficult</td>
</tr>
<tr>
<td>Internal design</td>
<td>2</td>
<td>✓ Precise describing of the internal design specification was difficult</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Analyzing the internal design of the prototype system was difficult</td>
</tr>
<tr>
<td>External design</td>
<td>1</td>
<td>✓ Understanding the difference between external design and internal design was difficult</td>
</tr>
<tr>
<td>Requirement analysis</td>
<td>1</td>
<td>✓ Took a long time to understand the difference between request specifications and requirement definition</td>
</tr>
</tbody>
</table>

Students can choose arbitrary program language. The presented prototype was developed by Java or Ruby on Rails. Therefore, some students developed their software using Java or Ruby for the first time. The experience of development using various program languages is important for the students, therefore it is a good opportunity for students. Moreover, students who selected presented themes had to read source codes developed by third persons. Source code reading is one good way of learning software development, but the students of standard level have less opportunity to read source code. This subject also provides opportunities for source code reading.

In the subject "Foundations of Open Source Software Development," we could provide an opportunity to have the experience of a software development process based on Open Source development. We could accomplish certain results by the students’ recognition of the importance of the design phase. However, effective support for development and testing are needed for more consideration. Developed software is published on the UEC software repository. However, the repository and software (including documents) are described in Japanese only.

4.4 Published Software on the UEC Software Repository Developed in the Open Source Software Development Subject

Currently the UEC software repository published software developed in the Open Source Software Development Subject shown in Table 6. Software from No.1 to No.4 was developed based on the students’ ideas. Software No.1 (NehanProxy) was developed by changing an existing OSS. Software from No.2 to No.4 was developed by a full scratch build. Software from No.4 to No.6 was developed by changing prepared prototype systems.

There is still not much software on the repository and moreover each software program on the repository should be improved for more usability. The repository does not play enough of a role in the contribution to society. We will continue revising our material pedagogical method in order to realize our purpose of contribution to society.

Table 6. List of software developed at the Open Source Software Development Subject in 2011

<table>
<thead>
<tr>
<th>No</th>
<th>Title</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NehanProxy</td>
<td>Program for displaying Japanese characters vertically on browsers</td>
</tr>
<tr>
<td>2</td>
<td>Pagvp</td>
<td>Program for describing statistical graphs</td>
</tr>
<tr>
<td>3</td>
<td>GUIIP</td>
<td>Image conversion program (brightness, grayscale, feathering, sharpness)</td>
</tr>
<tr>
<td>4</td>
<td>ViewSim</td>
<td>Image conversion program (brightness, monochrome, sepia)</td>
</tr>
<tr>
<td>5</td>
<td>Authentication function on a Slide Management System</td>
<td>Implementation of authentication for a slide management system</td>
</tr>
<tr>
<td>6</td>
<td>Visual assistance on Slide Management System</td>
<td>Implementation of visual assistance (zoom) for a slide management system</td>
</tr>
</tbody>
</table>

5. Related Works

There are some attempts for practical software engineering or OSS development in different subjects [6] - [13].

Gnatz et al. [10] reports a project based software development education with a corporation. The goal of this subject is to learn development style in an actual company. Students develop software within three months using the relevant company’s requirement. The company is involved
in testing, and decides whether or not to accommodate the request after the final presentation from the students.

Buchta et al. [11] presented a project based OSS development program in a course. The course addresses revising some of the OSS such as JAdvisor, JCDSee, and WinMerge along with the requirement from. The goal of this subject is to have the experience of an actual OSS development style in the university.

Borner et al. [12] reports an education of software engineering using the software repository. In this paper, they make good use of the software repository in their lectures and research, especially for designing novel information visualization using material such as "Description," "Applications," and "How to Use."

As for software development education using open source software, the effort of Jaccheri et al. [13] is an example of it, describing a case in which students took part in the actual open source project, "NetBeans."

We focus primarily on the development of OSS based on the student's idea, and providing practical software development opportunities. Realizing the student's idea by developing software is especially important for the student's software development skills and design skills. Therefore, each student develops their software by themselves.

These days there are so many smart phone applications, and this kind of software can impact users. We intend to construct a software development method based on the OSS development style, and software engineering knowledge in a small size while developing meaningful applications. Therefore, we have not adopted a plan to use an existing open source project. It is important to develop software in consideration of its evaluation by third parties. Students release their software developed in the lecture or research through the UEC software repository.

5. Conclusion

In this paper, we presented an outline and described the current state of the UEC software repository. We also reported the details of our practical software development education based on Open Source Software. The subject "Open Source Software Development" was started only one year ago. After this fiscal year we will publish some additional software on the repository.

At the present stage, we intend to provide a subject where students can learn the software development process based on their ideas as OSS. We will enhance our OSS development education method by adding a method of the sophistication of the ideas of students. Moreover we will revise our material so that students can gain the knowledge of software engineering topics through development.

References


Takaaki Goto graduated with a Doctor of Engineering degree from Toyo University in 2009. He is a Project Assistant Professor at the Graduate School of Informatics and Engineering at The University of Electro-Communications. His main research interests are the applications of open source software development, software development education, graph grammars, visual languages, and software development environments. He is a member of IEICE Japan, IPSJ, ACIS, and IEEE.

Kensei Tsuchida received M.S. and D.S. degrees in mathematics from Waseda University in 1984 and 1994, respectively. He was a member of the Software Engineering Development Laboratory, NEC Corporation in 1984-1990. From 1990 to 1992, he was a Research Associate of the Department of Industrial Engineering and Management at Kanagawa University. In 1992 he joined Toyo University, where he was an Instructor until 1995, and an associate professor from 1995 to 2002, and a Professor from 2002 to 2009 at the Department of Information and Computer Sciences, and since 2009 he has been a Professor of Faculty of Information Sciences and Arts. He was a Visiting Associate Professor of the Department of Computer Science at Oregon State University from 1997 to 1998. His research interests include software visualization, human interface, graph languages, and graph algorithms. He is a member of IPSJ, IEICE Japan and IEEE Computer Society.

Tetsuro Nishino graduated from department of Mathematics, Waseda University, and continued his research and obtained a D.Sc. in Mathematics in 1991. He was a researcher at Tokyo Research Laboratory, IBM Japan during 1984-1987. He was an Associate Professor at School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku during 1992-1994. In 1994, he joined at Department of Communications and Systems Engineering, The University of Electro-Communications as an Associate Professor and in 2006 he became a Professor. He received The Funai Information Technology Prize (2003) and IBM Faculty Award (2008).
Information for Authors

Scope

The aim of the International Journal of Computer & Information Science is to present information on well defined theoretical results and empirical studies, to describe current and prospective practices and techniques, to provide critical analysis of the use of CASE tools and to present broad issues in the fields of Computer Science and Engineering, Software Engineering and Information Science and Engineering. A primary aim of the journal will be to make the research accessible and useful to practitioners through carefully written methodology section and discussions of research results in ways that relates to practitioners.

Submission Policy

1. The IJCIS does not accept manuscripts or works that have been published in a journal, but accepts substantially revised and expanded versions of papers that were previously presented at conferences. In this case, the author should inform the Editor-in-Chief of the previous publication when the paper is submitted.

2. Submissions, referencing and all the correspondence will be conducted by e-mail.

3. Manuscripts should be in English and are subject to peer review before selection.

Procedure for Submission for Review

1. Send your complete text of manuscript to the Editor-in-Chief (e-mail: lee@cps.cmich.edu) via e-mail as an attachment document either in Word, PDF or PostScript format.

2. It may be useful for non-English speakers to have a native English speaker read through the article before submission.

3. Your e-mail message (to the Editor-in-Chief) should include the title of your paper, author(s) information (department/section, university/company, phone/fax numbers, e-mail address), the corresponding author’s complete address (mailing address), and area(s) of the paper.

Style of Manuscript to be Submitted for Review

1. The text should be double-spaced, two columned, single-sided in Times 12-point type on 8.5-inch x 11-inch paper with margins of 1 inch on all four sides. It should not exceed 30 double-spaced pages. All pages must be numbered sequentially.

2. Provide a 200-250 word abstract, 5-7 keywords. Your abstract should briefly summarize the essential contents.

3. All figures/tables must be captioned, numbered and referenced in the text.

4. An acknowledgement section may be presented after the conclusion section.

5. References should appear at the end of the paper with items referred to by numbers in square brackets.

Journal Papers: author’s first initials, followed by last name, title in quotation marks, journal (in Italics), volume and issue number, inclusive pages, month and year. For e.g.:


Papers of Proceedings: author’s first initials, followed by last name, title in quotation marks, proceedings / symposium / workshop name (in Italics), location, month, and year, and inclusive page numbers. For e.g.:


Books: author’s first initials, followed by last name, title (in Italics), edition, location: publisher, year. For e.g.:


Accepted Manuscript Submission

All accepted papers are edited electronically. You are required to supply an electronic version of the accepted paper. The detailed guidelines for preparing your final manuscript will be sent upon article acceptance. It is also available on the ACIS website at http://acisinternational.org.

Page Charges

After a manuscript has been accepted for publication, the author will be invoiced a page charge of $400.00 USD for up to 10 pages if he/she is not a member of ACIS and thereafter an additional $30 per printed page to cover part of publication after the receipt of your final manuscript (camera ready).