ISSN 1883–1974
No. 33, April 2010
(This English language edition of NII Today corresponds to No.47 of the Japanese edition)

NII Today

National Institute of Informatics News

NII SPECIAL **New Trend of Software Engineering**

Encouraging Interactions Between Asian Researchers —Towards holding NII-based international seminars

Development of innovative bidirectional software

Constraint programming in an attempt to stretch the limits of computers

Study group for enhancing software dependability established

Zhenjiang Hu

Professor, Information Systems Architecture Research Division, NII

NII Interview: Zhenjiang Hu + Kayoko Yamamoto

Encouraging Interactions Between Asian Researchers

Towards holding NII-based international seminars

NII SPECIAL

NEW TREND OF SOFTWARE ENGINEERING

Yamamoto: I am informed that you regard researchers' communities as important for the international development of software research. What are researchers' communities like?

Hu: Communities where researchers interact are usually formed through academic societies and meetings. For example, some researchers from the West are already aware of the results of other researchers before the official announcement is made, in more than a few cases. Researchers from many countries often meet and gather, say, at international conferences held in Europe and America, and the information is conveyed upon these occasions. Researchers present their new findings when they gather in these locations, and are criticized, encouraged, or given the opportunity of working together. They gain a great deal through research and personal interactions. If this happens often, people feel attached to these places as their communities.

Needless to say, Asian researchers can also participate in these conferences. However, they meet less frequently, and it is not easy to become part of the communities.

Yamamoto: Although this is a borderless age, close proximity is important for forming communities...

Hu: It is easier to visit China and Korea from Tokyo than Europe, isn't it? Unlike in the past, the level of research conducted in Asia has risen to a point where discussions are now possible on a global level. I therefore believe that an Asian community should be established and a mechanism formulated from which the ongoing work and research results are communicated to the world. In 2009, the Asian Workshop on Foundations of Software (AWFS) was held at GRACE Center, NII's Center for Global Research in Advanced Software Science and Engineering. In addition, I was involved in the Seventh Asian Symposium on Programming Languages and Systems (APLAS 2009) held in Seoul, Korea, as the program committee chairman. A great number of researchers from the West also participated, and the Symposium was highly regarded worldwide.

Yamamoto: Please tell us about the new type of seminar you are planning.

Hu: It will be an Asian version of the famous Dagstuhl Seminar(＊) in Germany. The most notable feature is that the Seminar aims to encourage interactions between researchers who are active worldwide through discussions on important issues in each area of informatics. It is not of a structure where programs are prepared beforehand and researchers make presentations. I have participated in the Seminar several times. One of the organizers is German, and around 30 of the world's top researchers take part.

To encourage interactions, participants stay in a location slightly remoted from the city center for a week. On the first day, each of the participants presents the issues in the areas he/she is involved with and provides a brief introduction to the research he/she is

(＊)Dagstuhl Seminars refer to international meetings that have been held every week since the 1980s in Dagstuhl, Germany. The world's top informatics scholars gather at this residential seminars and discuss open problems.

Kayoko Yamamoto

Editorial staff writer, Science and Technology Division,
Editorial Department, The Nikkan Kogyo Shimbun, Ltd.

engaged in. Then all the participants vote to determine the program for the entire week of the Seminar. The participants dine together, and their seats are switched each time by lottery. Hiking and other events are planned. During the week of the seminar, the participating researchers become very close.

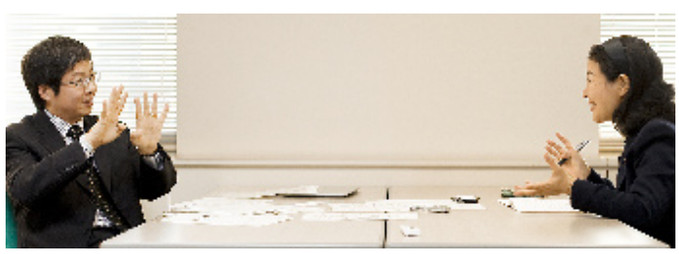
The Seminar is extremely popular, and one program is implemented for one week only, with another program commencing the following week. The weekly programs are arranged two and a half years in advance. Even if one expresses a wish to hold a seminar with a certain theme, only half the proposed topics are adopted. I definitely hope to hold an extremely popular seminar that is effective in both forming a community for researchers in Asia, and promoting research communication worldwide. Isn't it an appropriate project for NII to pursue as it aims to become a center of research activities in Asia?

Yamamoto: Would there be any difference from the German version?

Hu: The organizers should not be limited to researchers from NII or other universities or research organizations from Japan. I hope that researchers from universities that have concluded tie-up agreements with NII will also function as organizers, including Beijing University and Tsinghua University of China, Seoul National University of Korea, the National University of Singapore, and Hanoi University of Technology of Vietnam, etc. NII will provide full support for the clerical aspects of the seminar so that the researchers can concentrate on discussions. We are extremely fortunate that most of the clerical workers from NII are fluent in English.

Yamamoto: Perhaps synergistic effects can be expected with the international exchange programs of NII.

Hu: There are many ways to enhance the effects, such as coordinating the schedule with personal interactions for joint research at NII, and having university students participate in the seminar.



NII has initiated numerous international joint research projects. Each of them has been recognized by the global society, but we cannot say that the activities of NII are fully understood. International meetings held by NII once or twice a year are not sufficient, but if they were held regularly once or twice a month, more impact would be felt. Through sustained actions, we can make our activities more visible.

Yamamoto: As an inter-university research institute, does NII have a mindset that differs from that of universities?

Hu: We are responsible for leading society toward change by exercising leadership in revitalizing research activities in universities and other institutions, and by offering services that are commonly required by universities. We do hold meetings for purely academic purposes. At the same time, we set up opportunities for discussion between theoretical researchers and people working in enterprises regarding the issues associated with the application of new technologies in society, for the above purpose.

Although NII is not a university, it has a system for educating students in doctoral programs. This is possible because we participate in the Graduate University for Advanced Studies, where diverse inter-university research institutes are utilized for education. Although many foreign students and working people study at NII, it is not well known because of its special systems. If NII gains more presence in Asia and understanding is fostered, I expect that we will attract a greater number of students from other Asian countries.

A Word from the interviewer

Professor Hu from China enrolled at the University of Tokyo in 1992 as a government-sponsored foreign student, and transferred to NII in 2008 after serving as Associate Professor of the University. Being familiar with the differences between Japan and China and also those between universities and inter-university research institutes is an asset when attempting community development involving the gathering of many researchers. Although Professor Hu has worked at NII for only two years, he has maintained a positive attitude of proposing a new type of seminar that NII should pursue on an organizational level, which is one of his characteristics. Professor Hu is an expert in programming language and software engineering, and the scientific analysis of programming capabilities by humans is one of the areas he is working in. The policy of NII, according to which social issues are dealt with by focusing not only on technologies but by merging humanities and sciences, certainly appears to have circulated.

New Trend of Software Engineering

Development of innovative bidirectional software

Productivity enhancement has long been a major issue in software development. Conventionally, update was undertaken by tracing back to the upstream processes when any change was made in the process of software development. This requires a tremendous amount of labor. In reality, the development works progress without tracing back to the previous processes, and discrepancies between the design and the implemented systems are often seen. Accordingly, Professor Hu and his team have developed an epoch-making system to synchronize the updates made in each stage of the processes, which could dramatically change the software development process. We interviewed the three researchers regarding the study of linguistic infrastructure technologies for bidirectional model transformation.

Zhenjiang Hu
Professor, Information Systems Architecture Research Division, NII

Soichiro Hidaka
Assistant Professor, Information Systems Architecture Research Division, NII

From unidirectional to bidirectional software development

Hu: NII has addressed various difficult issues to be tackled by next-generation informatics in the Grand Challenge Project Program. Our Study of the Linguistic Infrastructure Technologies for Bidirectional Model Transformation is one such Project. It is an international project organized mainly by NII researchers, including myself, with the participation of researchers from diverse organizations including the University of Tokyo, the University of Electro-Communications, the Shibaura Institute of Technology, Peking University, and Shanghai Jiao Tong University.

The keyword for the project is "evolution." Instead of simply developing software, the research aims to develop software that can cope with a society that is undergoing changes every minute, and make the software evolve along with the changes. For this purpose, the core technology would be "bidirectional." In other words, we are aiming to develop technologies with feedback mechanism in the process of software development.

For example, if you prepared a report and submitted it to your boss, he/she would ask you to make corrections, such as enlarging the title or using different wording. You would then make the corrections and submit the report again. We wanted to do the same in the world of programming. Conventionally, software development proceeded in one direction, from upstream to downstream. In contrast, the bidirectional approach plans to form a loop. If a correction is made in a downstream process, automatic corrections are made in all the other processes, including the design stage in the upstream. We hope to evolve programs in this way.

In conventional programming, directives flow in one direction only. If you need to trace back upstream, it is necessary to write another program to return. However, it is very difficult to formulate two programs while maintaining consistency between the two directions, or to make changes by tracing back manually. We hope to build the infrastructure technologies for developing software that can evolve and apply them in various settings.

——What are the concepts behind the technology?

Hidaka: We use programming languages to develop software. Most of the conventional languages, like Java, are one-directional. We plan to create a new language enabling bidirectional computation. Essentially programming languages represent computations. Numerous small-scale computations are combined to create large-scale computations, and commands are given to computers. We decided that we could perhaps create a framework for bidirectional communications if each of the small-scale computation components were given bidirectional properties and then combined. For this purpose, it would be important to divide the programs neatly into small components, and provide easy-to-understand descriptions for them.

Kato: To be honest, the importance of bidirectional software transformation was highlighted in the 1980s. Trial and error was undertaken during this decade, with no specific methodologies

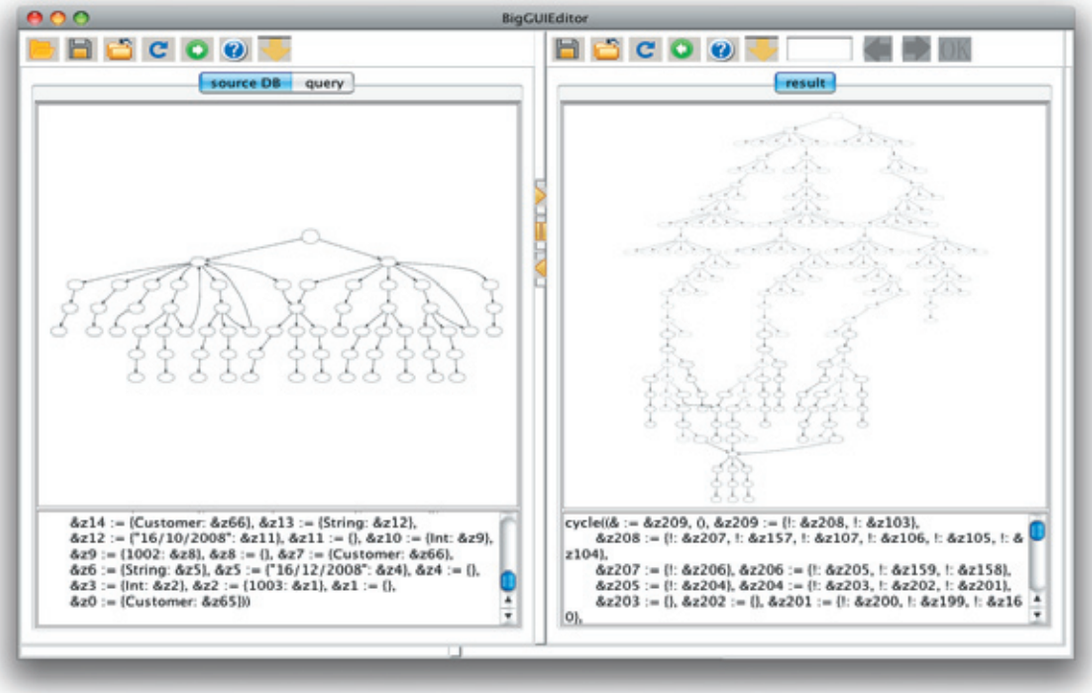


Fig.1: A personal computer screen showing examples of complicated graph transformations; the graph on the right shows the transformed version of the graph on the left.

able to be found. Among the results obtained during the 1980s was the development of a method called "View updating," which enables the cutting out of the portions to be modified and the provision of modification when making changes to a large database. Full-scale research into bidirectional technologies has been pursued since 2000. For instance, researchers from the University of Pennsylvania in the United States developed languages for synchronization, which enable, for example, the change made to a schedule book on the website to be reflected to data in different devices, such as personal computers, PDAs, and mobile phones.

Hu: I transferred to NII from the University of Tokyo two years ago, and the Takeichi Research Group, which I joined, was also trying to develop bidirectional languages for documents (specifications prepared when developing programs, etc.) It is a mechanism that can be used for website maintenance and other applications. If a website is required to be changed, it would be convenient if anyone could easily make the corrections without relying on expert service providers. Bidirectional transformation systems can achieve this too.

By the way, the majority of conventional programming languages deal with sequences and trees efficiently. The tree structure refers to a hierarchical structure without a loop. And most of the bidirectional transformation systems developed to date address computations on tree structures. Looking at the actual situation in society, however, we see mostly network structures, be it the relationship between a worker and his/her boss, or railway routes. As a result, we decided to address graphs. The graphs we are talking about comprise points (nodes) and lines (edges) that connect them, rather than the line charts or radar charts used with statistical data. Unlike tree structures, graphs sometimes share nodes or form loops. We are the first researchers to address graphs for bidirectional transformation mechanisms, which makes us pioneers in this area of research.

Development of transformation languages for achieving bidirectional mechanisms

——Specifically, what methods are used for driving the mechanisms?

Hu: In software development, executable instructions are developed step by step. The software artifact for each step (the software specifications/design drawings) is represented using a "model." The model refers to something that represents the essential nature of the object to be understood or manipulated according to a predetermined method. The Unified Modeling Language (UML) is used to represent models. Using UML, a model that uses graphical descriptions to express the abstract system is generated (Fig.1). In other words, the model we are talking about is essentially a graph.

Furthermore, transformation takes place from one model to the next, and so on. Ultimately, the codes (in the form of a graph) for executing the software can be generated. What we developed recently was the bidirectional transformation language UnQL+, which enables transformation in the reverse direction to provide bidirectional properties to the series of models.

Hiroyuki Kato
Assistant Professor, Digital Content and Media Sciences Research Division, NII

New Trend of Software Engineering

Kato: The language is an extended version of the existing query languages (languages for making queries on computer data). In other words, the language for extracting graph data has been extended to a language for transforming one graph into another. Grammatically, the syntax is very close to that of SQL (one of the query languages).

By using the bidirectional transformation language, automatic and consistent update would be possible by tracing the processes back to design or customer demand analysis, even when a change has been made in the implementation phase. This way, evolution of the software would be possible. The software design is separated from the actual architecture, because models are used. For this reason, the models can be reused even if the implementation technologies and architecture have been changed.

Hidaka: Let me now give a demonstration (Fig.2). You can see the models on the left- and right-hand sides of the screen. If you click the portion you want to update on the model on the right and make an update, the corresponding portions of the model on the left to be changed are displayed in red. You can see at a glance which portions have been changed and how. In actual programs, the size of models described is huge. It is therefore extremely significant that the portions that have been changed are displayed and made visible in synchrony. By the way, the text seen at the bottom is the representation of the diagram above in bidirectional transformation language. The transformation occurred instantly, and the speed is 100 times faster than the system we initially developed.

——Specifically, to what scenarios can the language be applied?

Kato: For example, when we develop software, it is tested before completion. If a defect is found, it needs to be corrected. If the correction is made in each process tracing back to the design drawings, the design (model) can be reused for other programs. It is very important to share information on the specifications of updated portions after testing by tracing back to the design phase, be it an automobile design or semiconductor design. Until now, there was no choice but to do this manually. Particularly in the case of the development of large-scale software, the design and implementation are undertaken by different people in many cases. As a result, a mechanism for automatic and bidirectional synchronization is needed to share information and reuse the design.

Hu: Speaking of bidirectional, there are many phases to it. For example, spotting the cause of a defect by tracing it back to the upstream processes, or the traceability mechanism, is one of them. Another is the mechanism that provides clues as to how to make a change when you wish to change some portions, and that makes the change by automatically achieving consistency and synchronicity. By using our mechanism, it would be possible to achieve bidirectional properties at various levels.

Hidaka: When we develop a program, it is rare for a genius programmer to begin writing it from scratch. Normally, programming proceeds from design to implementation phases in steps, according to a common method that everyone involved in the programming can understand. Since there are countless similar programs available in the world, it would be possible to reuse the software development processes if the consistency of the system were guaranteed by bidirectional transformation. We believe this would help enhance reliability and productivity.

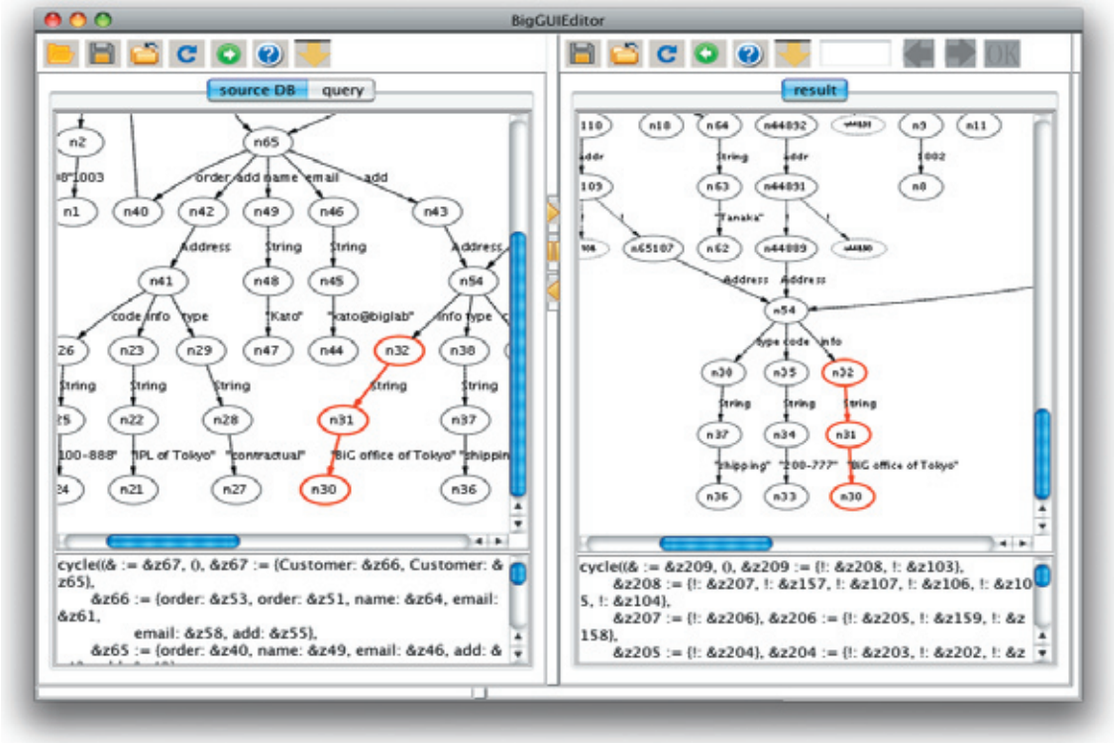


Fig.2: The node correspondence is shown in red; when an update is made on the right-hand screen (the red portion), the corresponding locations on the left-hand screen are displayed in red; the locations to be updated are identified at a glance.

Bidirectional transformation system under the spotlight

Hu: Our research has attracted a great deal of attention through the presentation made at the world's most authoritative academic society of software engineering [ICSE2009 (New Ideas and Emerging Results Track), FSE2009], and a paper chosen as the best paper at the Runtime Model Workshop, written by a student from Peking University who has worked at NII as an intern. In 2008, we organized the GRACE International Meeting on Bidirectional Transformations sponsored by the Center for Global Research in Advanced Software Science and Technology (GRACE) in NII, and held it at Shonan Village Center. The Meeting provided an opportunity for around 30 leading researchers on bidirectional transformation from numerous countries to gather and hold discussions. As can be seen, it is also very important that we have contributed to establish the new communities.

Moreover, to make our studies be widely recognized by many researchers, our project website (http://www.biglab.org) provides all papers we have published, the system and its source codes. In addition, we have on hand a demonstration system on which the bidirectional transformation can actually be performed, so that anyone can easily test the system. Recently, research institutes and enterprises have contacted us about the system. Discussions have commenced with an automotive parts manufacturer, as well as applications to sensor network research.

——What are the future prospects?

Hidaka: It was very helpful that each time we submitted and presented a paper at international conferences, numerous researchers expressed their opinions and provided criticism and advice, which led to improvements to the system. I hope to reason about the well-behavedness of the system in more detail and determine its properties.

Kato: At the same time as making improvements so that the system can be operated even more efficiently at greater speed, I think we should present what the system is suited for and also its drawbacks, rather than claiming that it can do everything. By doing so, I believe we can develop a truly useful system.

Hu: Software maintenance is a perpetual theme, and is the most costly aspect. Furthermore, it is extremely difficult to add new functions to software once it has been developed. Our research team has worked on the development and optimization of programming languages, along with theories on program development. It is built on a grand scheme that intends to develop the language itself from scratch and open up a new area of software engineering. It is extremely challenging, and that is the fun of it.

The members are having tough time staying up all night for many nights in a row, however, I would be very happy to see a greater number of application examples and persuade the world of the system's utility. (Written by Madoka Tainaka)

Constraint programming in an attempt to stretch the limits of computers

In response to progress in informatics and to train the younger generations, NII and Waseda University concluded an agreement for the effective exchange of human resources in 2005. How has the joint research on the leading-edge programming progressed under the agreement?

Kazunori Ueda
Professor, Faculty of Science and Engineering, Waseda University

Hiroshi Hosobe
Associate Professor, Information Systems Architecture Research Division,NII

When a ball is dropped from a height, it falls freely while gradually accelerating. When it hits the floor, it jumps upward by suddenly reversing its direction. After repeating this process several times, the ball loses momentum and stops bouncing. This is a physical phenomenon with which everyone is familiar. The computation process for the trajectory of the falling ball and the speed is described in high school physics textbooks. However, it is not easy to accurately simulate the overall motion of the ball using a computer.

The world we live in is more complicated than we think

"The freefall of a ball is relatively easy to understand. It is terribly difficult to use a computer to analyze the phenomenon of three or more billiard balls colliding at the same time," says Professor Kazunori Ueda of the Faculty of Science and Engineering, Waseda University. Although the movements of billiard balls and a free-falling ball are physical phenomena that we encounter often, they pose difficult issues for computers. Looking at these physical phenomena in detail, we find continuous elements, which continue certain motions, and discrete elements, where speed and direction change suddenly. Computers can handle both continuous (analogue) and discrete (digital) elements. They are rarely expected to handle them both at the same time, however, and are inferior at such tasks.

Just as automobiles equipped with two different motive powers, such as a gasoline engine and an electric motor, are called hybrid automobiles, the simultaneous handling of continuous and discrete elements is referred to as hybrid in informatics (Fig.1). Today, there is a rising worldwide movement to use computers to solve hybrid issues. At NII, Associate Professor Hiroshi Hosobe of the Information Systems Architecture Research Division, together with Professor Kazunori Ueda, has conducted research into this new area since around 2004.

Never say "I don't know"

There are many methods for addressing hybrid issues, and the two researchers plan to use constraint programming (Fig.2). The constraints describe the conditions for achieving something in a way that is easy for computers to handle. Computers are not good at handling uncertain values. However, only a few phenomena that occur in our world provide definite data.

For example, let us consider whether a ball thrown by a pitcher enters the strike zone. For this purpose, the ball's speed, the angle of the ball when it is thrown, the rotating force on the ball, and numerous other elements need to be taken into consideration. The speed of the ball when it is thrown varies infinitely, as it is sometimes slow and sometimes extremely fast. But simulation would not be possible if we said "we don't know" and gave up. A constraint is therefore set whereby the ball is thrown at a speed of between 140 and 150 km/hour. This way, the angle of the ball and its rotating force can be calculated, although within a limited range of speed. In reality, no pitcher can throw a ball at 200 km/hour, so there is no issue created by constraining the speed. Rather, it is more significant that solutions can be obtained. This concept of constraint programming has been applied in diverse areas of informatics.

Bond created by constraint programming

One of the reasons why computers can now be operated easily by anyone is the dissemination of the graphical user interface (GUI) during the 1990s. GUI refers to the operation screen upon which users can operate the computer as desired by simply clicking on icons and

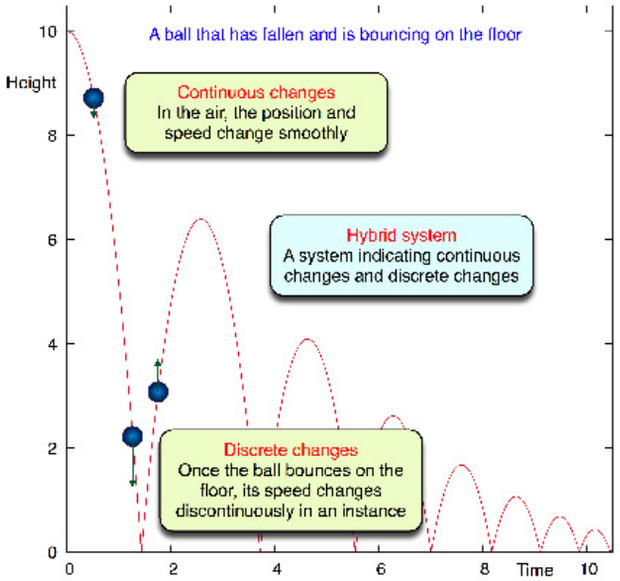


Fig.1: Here, hybrid refers to the handling of both continuous and discrete changes. The motion of a ball that falls and bounces on the floor is regarded as a hybrid issue.

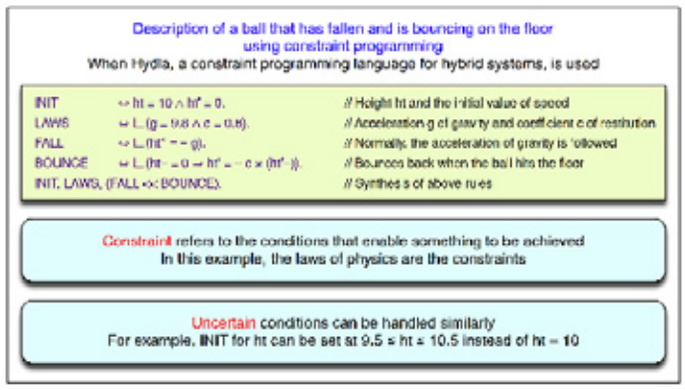


Fig.2: When handling a hybrid issue such as the motion of a ball that has fallen and is bouncing on the floor, one of the possible methods is to use constraint programming.

menus on the display using a mouse, etc. Today, this technology is taken for granted on the Windows and Mac operating systems. Associate Professor Hosobe has carried out research into the GUI since his undergraduate years. He chose constraint programming as his method of approaching the issue. By using constraint programming, diagrams and texts can be arranged on the operating screen without being overlapped or dislocated.

His efforts were recognized, and in 2002 he was invited by Professor Philippe Codognet of Paris 6 Université Pierre et Marie Curie and Professor Ken Satoh of the Principles of Informatics Research Division, NII, to join their project. Since then, Associate Professor Hosobe has undertaken joint research with French university researchers.

On the other hand, Professor Ueda is adept in the area to the point where he commented, “What I discovered through 25 years of research into constraint programming is that it is useful for neatly describing communications or exchanging information.” Associate Professor Hosobe says, “When we commenced the joint research with the researchers from the University of Nantes in France in 2004, I asked Professor Ueda to join to reinforce our team.” Since then, they have both worked on various types of research. Today, they are concentrating their efforts on hybrid.

Nurturing is important

“Programming requires manpower, and we need to nurture excellent students,” says Professor Ueda. NII and Waseda University concluded an agreement in 2005 to mutually strengthen their partnership. Within the framework, Associate Professor Hosobe became Visiting Associate Professor of Waseda University, and Professor Ueda became Visiting Professor of NII. This led to more frequent interactions between the two institutions, as Associate Professor Hosobe now participates in discussions at Waseda University at least once a week. It is important for the students of both NII and Waseda to relate to many faculties and seek their guidance. “Thanks to the hard work of a doctoral degree program student, our research has progressed significantly.” Associate Professor Hosobe is keenly aware that the students have made a great deal of progress.

Software that can guarantee accuracy

What becomes possible by applying constraint programming to hybrid? Software can be developed for creating animated picture shows with realistic motions, and for simulating automobile behavior and life phenomena. The two researchers are determined to achieve a level of software development that can guarantee accuracy. “Currently, there are no well-known software products that guarantee the accuracy of computation results. You may find it hard to believe, but guaranteeing accuracy is technically very difficult,” says Professor Ueda, expressing his dissatisfaction with today’s software. Furthermore, he says that by combining hybrid and constraint programming, it becomes possible to guarantee the accuracy of results, even under certain conditions. The reliability of the world of informatics is certain to improve dramatically. We cannot wait until the team of Professor Ueda, Associate Professor Hosobe and the students provides software that can be used with a sense of reliability. (Written by Akiko Ikeda)

Study group for enhancing software dependability established

In December 2009, NII established a study group called the Dependable Software Forum (DSF) along with Japan's five major IT companies: NTT Data, Fujitsu, NEC, Hitachi and Toshiba. DSF plans to continue with research and development work to achieve fault-free, dependable, safe software.

Shin Nakajima
Professor, Information Systems Architecture Research Division, NII

There are two approaches for developing dependable systems. One is to minutely control the work processes of the engineers . The other is to use a tool that automatically confirms the correctness of the design, for example the so-called formal methods(*). At DSF, a working group to study the methods for applying the formal methods to software development has been established. It plans to formulate guidelines, etc. by March 2012.

Verifying the correctness of software with formal methods

Formal methods are software development methods into which research commenced in Europe during the 1970s. These methods are backed up scientifically by mathematical logic, and can guarantee high reliability. In particular, correctness can be systematically verified by meticulously describing the specifications using a language based on mathematical logic for early stages of the software development processes, which serves as the cornerstone of large-scale software development.

The proposal document entitled "Toward Infrastructure Software Development for Achieving a Safe and Secure Society" published by the Industry-Government-Academia Cooperation Committee on Strategies for Infrastructure Software Technologies in March 2009 suggests that we should start with investigative studies for enhancing software reliability and productivity by applying the formal methods in the industries. In response to the proposal, Deputy Senior Executive Manager Tsuyoshi Kitani of the Research and Development Headquarters of NTT Data Corporation encouraged those in charge within each of the participating companies to establish a framework for cooperation, with NTT Data Corporation serving as the secretariat.

In the United States, the formal methods are mentioned in the 2007 report produced by Networking and Information Technology Research and Development (NITRD), a consultative body of the President that performs the core roles in IT strategies. In Europe, on the other hand, much of the software research investment is allocated to formal methods in the EU 7th Framework Programme for Research (FP7, 2007-2013). In view of these activities in the West, Japanese researchers cannot just wait and see if international competitiveness is to be maintained.

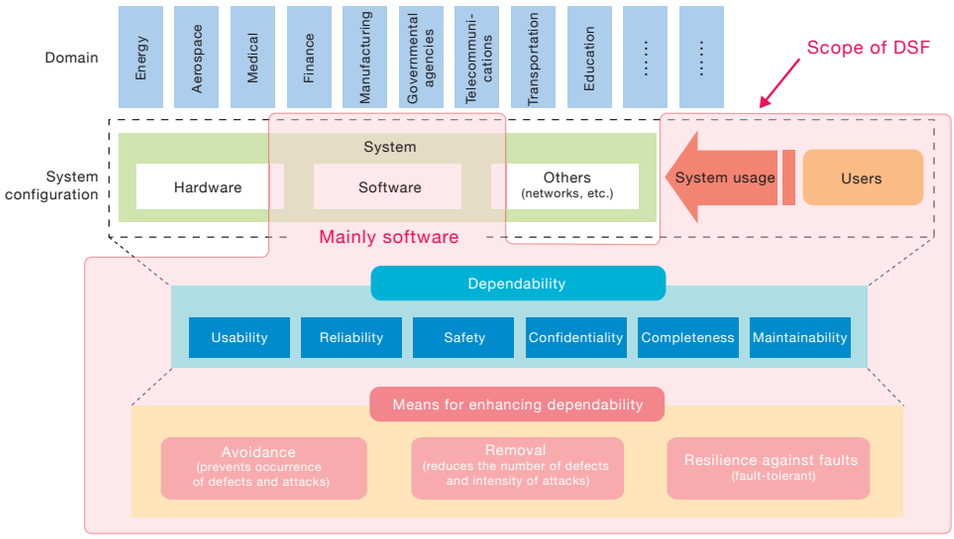
Professor Shin Nakajima of NII's Information Systems Architecture Research Division commented, "As a researcher, I have always been aware of the novelty and importance of the formal methods. However, it is meaningless if we direct society in how to use them. It is quite significant that NTT Data Corporation is assuming the leading role as an organizer so that the industries undertake studies into the formal methods on their own." Among the missions of NII is collaboration with universities and industries, and DSF intends to function well in setting the academic direction.

Rigid control of early stages of development reduces workloads and costs

There are several methods of description for the formal methods, which were all developed in Europe. They have grown in stability in recent years, and some of them support the Japanese language. To fully harness the descriptive methods, the user must learn the basics of set theory and mathematical logic, along with the descriptive texts themselves. Time for studying needs to be reserved before introducing the methods, and sometimes efficiency declines as many users find the approach difficult. For science and technology students, however, knowledge of mathematics taught in the freshman and sophomore years at university is sufficient. "Students put greater effort into studying when slightly difficult teaching materials are used, enabling them to deepen their understanding." (Professor Nakajima)

If any omission in the upstream design phase

(*)Formal methods: this is a generic term referring to methods that include technologies for describing precise specifications based on mathematical logic and for verification of the specifications. By precisely describing the specifications, defects in the software can be detected.



Scope of DSF: DSF conducts R&D studies on the scope within the red lines, mainly with regard to software. DSF encourages investigative research into methods for enhancing software dependability, beyond the framework of individual companies.

is entirely eliminated, the risk of problems in the downstream programming and testing phases can be reduced. Overall, a benefit is obtained in that the workloads and costs are significantly reduced. Consequently, the current development systems that rely on the large-scale mobilization of labor can be reformed. For the European researchers who developed the basic technologies, the universality of the technologies they developed can be confirmed if the formal methods work well in the Japanese styles of development systems. As such, the European researchers have also shown quite a lot of interest.

It has not yet been fully revealed what benefits are brought about by applying the formal methods to what type of actual issues, and how. In the case of large-scale systems, too much time would be required if they were developed utilizing the formal methods. As such, they would not be applied to all the software. However, it would be useful to spot the locations where the formal methods should be partially used. As a result, it was decided that, as a specific step, each participating company should take turns studying the library book reservation system they have all developed in the past, applying the formal methods this time. Using the example that the participants had previously designed using conventional methods, the differences compared with the cases applying the formal methods are readily seen. The findings are reported in the monthly meetings and shared. From April, they plan to commence demonstration tests that apply the formal methods a little more fully and to practical issues.

Overcoming competitive relationships to further the development of the software industry

Attempts by one company would not be sufficient to disseminate the formal methods. When major companies develop software, many software houses participate in addition to the ordering company, and it is necessary to ensure that all the developers acquire the formal methods in the future.

To disseminate the technology, it is desirable for the whole of Japan to work together. However, the study group has gathered without payment to date, and it is not supported by the national government, unlike in the West.

Deputy Senior Executive Manager Kitani noted, “Needless to say, we manufacturers compete against each other on a daily basis. But we stood up because we believe we should work together for the development of Japan’s software industry. If the benefits of the formal methods are shared more widely by studying one example after another, the industry will surely start using the methods enthusiastically.”

The number of tasks to be processed is projected to increase in the demonstration test phase, and the group is considering seeking additional members to distribute the load. The group could remain active even after March 2012 if required. Deputy Senior Executive Manager Kitani expressed his aspiration, “We hope to generate a trend whereby the formal methods are used whenever possible for verifying important systems that could affect human life or have a major impact on our daily lives.”

Professor Nakajima reaffirmed the responsibilities and expectations borne by DSF, saying, “If DSF, which organizes the major companies of Japan, failed to establish pathways for the development of formal methods through investigative research, these methods could never take root in Japan.”

(Written by: Asako Tsukasaki)

Tsuyoshi Kitani
Deputy Senior Executive Manager,
Research and Development Headquarters
Director of Center for Applied Software Engineering,
NTT Data Corporation

NII ESSAY

The Dizzy Borrow-the-Name Game

Shin Nakajima
Professor, Information Systems Architecture Research Division, NII

When you see the Japanese words *Kumo-no-su*, *Koshi*, and *Kumo*, you might feel puzzled. But if they are written in *katakana*, as foreign words usually are, these words refer to Web, Grid, and Cloud. You might then realize that they relate to computer software. As can be seen, it is customary in the world of software to borrow a term from somewhere else when a new concept arises. Perhaps people borrow words because they think these words make the new idea easy to understand. However, terrible misunderstandings sometime result, because the image evoked by a word differs for each person.

Fully absorbed in Keyword Engineering

We sometimes choose a term to attempt to provide a novel impression of technologies and concepts that are similar to conventional ones. We assign a new name or search for more attractive words, and it is like a first-come-first-served borrow-the-name game. People call it Keyword Engineering. On the other hand, the same word is occasionally used to denote different meanings, as with the word "model." It is forgivable if this is coincidental, but sometimes the choice of word appears intentional, or even an attempt to take advantage of people's misunderstanding. It could be called Dizzy Keyword Engineering. Members of the public are not the only ones who are confused. Software engineers are also mystified.

Keywords are replaced quickly in the world of software. An increasing number of working people have sought doctoral programs to work on, because they are interested in pursuing the basis of the technologies they have come to know in the practical business world. These are enthusiastic people who wish to know the fundamental technologies and contribute toward technological development that is useful in the engineering practice. However, some of them find themselves absorbed in Dizzy Keyword Engineering after commencing their studies, and feel bewildered.

Escape from the dizzy world at graduate school

Our daily lives are surrounded by invisible software. If this software failed, not only mobile phones and ATM terminals at banks would be seriously affected, but also the railway passenger gates we use for commuting. Technologies for developing software that functions and performs as expected are indispensable, and the importance of software engineering never ceases to grow. Methods for automatically confirming that there are no software bugs have been under investigation for quite some time, and a technology known as Model-checking has been in the spotlight. An increasing number of people wish to study this technology at graduate schools, which is wonderful.

The word "model" is the best performer in Dizzy Keyword Engineering, and people are often surprised to find that the word "model" in Model-checking differs completely from the meaning of the word we are familiar with. In our daily lives, the word "model" means a fashion model, an example to be learned, or a typical example. There are also analytic models represented by numerical formulae. Graphical symbols (icons) and an object that resembles an actual item are also called models. Furthermore, "model" plays an important role in "Model-based Development" in modern Software Engineering. Such confusion comes to end when we find out that the word "model" in Model-checking is a term from mathematical logic.

Last, let me bid farewell to the borrow-the-name game with a dizzying question: "Does Model-checking refer to fashion models having a health checkup?"

Weaving Information into Knowledge

NII

NII Today No.33, April 2010 (This English language edition of NII Today corresponds to No.47 of the Japanese edition)
Published by: National Institute of Informatics, Research Organization of Information and Systems
Address: National Center of Sciences 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430
Chief editor: Yoh'ichi Tohkura Cover illustration: Makoto Komori Photography: Shuichi Yuri Design: Kotaro Suzuki Production: Sci-Tech Communications Inc.
Contact: Publicity and Dissemination Team, Planning and Promotion Strategy Department TEL:+81-3-4212-2135 FAX:+81-3-4212-2150 e-mail: kouhou@nii.ac.jp
http://www.nii.ac.jp/