

NII Today

National Institute of Informatics News

特集 ソフトウェア工学の新しい潮流

アジアの研究者の交流を推進

—NIIを拠点に国際的セミナー開催も

革新的な「双方向」ソフトウェアの開発

コンピュータの限界に挑む制約プログラミング

ソフトウェアの信頼性向上を目指す研究会が発足





こ・しんこう
国立情報学研究所
アーキテクチャ科学研究系教授

NEW TREND OF SOFTWARE ENGINEERING

アジアの研究者の交流を推進 ——NIIを拠点に国際的セミナー開催も

山本 胡先生はソフトウェア研究の国際化における、研究者コミュニティを重視していると聞きました。研究者コミュニティとは、どのようなイメージでしょうか。

胡 研究者が交流する場であるコミュニティは、学会や会議を通じて作られることが一般的です。例えば欧米の研究者は、別の研究者の優れた成果を、公式発表前にすでに知っていることが少なくありません。欧州で開かれる国際会議などに、各国の研究者仲間が頻繁に集まり、そこですでに情報が伝わっているからです。集まった場で新しいものを発表し、批判をされたり、励まされたり、一緒にやろうと声をかけられたりし、研究や人材交流で多くのものを得ます。それが頻繁になれば、そこは自分たちのコミュニティなのだと愛着が湧くでしょう。もちろん欧州の会議にアジアの研究者も参加できますが、顔を合わせる頻度が低いですし、そのコミュニティの一員になるのが簡単ではないのです。

山本 ボーダーレスといわれる時代ですが、仲間になるのにはやはり、近場であることが重要だと…。

胡 東京からでも、欧州より中国や韓国の方が頻繁に訪問できるでしょう。アジアの研究レベルは以前と異なり、世界的な議論ができるところまで高まってきました。ですから自分たちの居場所

として、アジアのコミュニティを確立し、そこから成果や主張を世界に発信する仕組みを、確立すべきだと考えています。2009年にはNIIの先端ソフトウェア工学・国際研究センターで、ソフトウェア基盤のアジアワークショップ(AWFS 2009)を開きました。もう1つ、韓国・ソウルでのプログラミング言語・システムのアジアシンポジウム(APLAS 2009)にもプログラム委員長としてかかわりました。これは欧米の研究者も大勢参加して、世界的な評価を得ています。

山本 計画している新しいタイプのセミナーについて、教えてください。

胡 有名なドイツの「ダグスツールセミナー(*)」のアジア版です。一番の特徴は、情報学の各分野で重要な課題について議論し、世界で活躍されている研究者同士が交流するのが目的だということです。事前にプログラムを用意して研究発表を行う形ではありません。私も何回か出席しました。オーガナイザーはドイツ人なのですが、参加者は世界の一流の研究者30人ほどです。

交流を促進するため、都会から少し離れた場所での1週間の合宿です。初日に各自が考えるその分野の課題や自分の発表したい研究内容を簡単に紹介し、皆の投票で1週間のセミナー全体のプログラムを決めます。食事と一緒に着席順は毎回、くじで入れ替えます

(*) ダグスツールセミナーとは、1980年代からドイツのDagstuhlで毎週に行われる国際集会である。世界トップクラスの情報学者が参加し、Open Problem(未解決問題)について議論する合宿形式のセミナーである。

NII Interview Zhenjiang Hu+Kayoko Yamamoto



やまもと・かよこ
日刊工業新聞社編集局科学技術
部 編集委員

山本佳世子

し、ハイキングなどの企画もあり、研究者同士はとても親しくなります。

このセミナーは非常に人気が高く、1つのプログラムは1週間限りで翌週は別のテーマで行われますが、その毎週の予定が2年先まで埋まっているほどです。あるテーマで開催を希望しても、半分以上しか採択されないのです

よ。このように人気が高く、コミュニティ形成効果が抜群のセミナーを、アジアの研究者向けに開催したい。アジアの研究拠点を

目指すNIIが継続的に手がけるには、ぴったりではないでしょうか。

山本 ドイツ版との違いはありますか？

胡 オーガナイザーはNIIの研究者や日本の他の大学・研究機関の研究者に限らないことにします。NIIが連携協定を結ぶアジアの中国・北京大学、清華大学、韓国・ソウル大学、シンガポール国立大学、ベトナムのハノイ工科大学などの研究者にも、オーガナイザーになってほしいですね。セミナーの事務的なサポートはNIIが全面的に行い、研究者には議論に専念してもらいます。幸い、NIIの事務職員は、ほとんどが英語を使えるので心強いですよ。

山本 NIIの各種の国際交流プログラムとの相乗効果も期待できそうですね。

胡 共同研究で人が行き来するのと合

わせたり、学生を参加させてみたり、工夫ができると思います。NIIは国際共同研究を多数、手がけており、その1つ1つは国際的に認められているのですが、ではNIIのアクティビティが世界に十分、理解されているかと問われると難しい。NIIの国際集會も毎年1回、



半年に1回では不十分ですが、定期的に月1〜2回となるとインパクトがあります。継続により、アクティビティに見える形に変えられると考えています。

山本 NIIは大学共同利用機関として、大学とも違う意識があるのでしょうか。

胡 私たちには、大学などの研究を活性化する仕組みについてリーダーシップを取ったり、大学が共通で必要とするサービスを提供したり、そのような形で社会の変化をリードする責任があると思っています。純粋な研究討論会も開きますが、企業人と理論研究者を同じ場に集めて、新しい技術を実社会に活用するための課題を議論する場を設定するのも、そのような意識があるからです。

NIIは大学ではありませんが、博士課程の学生の教育ができる仕組みを整えています。さまざまな大学共同利用機関を教育の場に活用する、総合研究大学院大学に参加しているため、それが可能です。留学生や社会人学生も多いのですが、仕組みが特殊なこと

もあってあまり知られていない面があります。アジアでのNIIの存在感が高まり、NIIに対する理解も進めば、アジアからの留学生も増えるのではと期待しています。

◆ インタビュアーの一言

中国人の胡教授は国費留学生として1992年に東京大学に入り、准教授まで務めて2008年にNIIへ移ってきた。日・中に加え、大学・大学共同利用機関の違いも経験していることは、さまざまな研究者が集まるコミュニティづくりにプラスになるだろう。NIIに来てまだ2年といいながら、組織的に取り組むべき新タイプのセミナーを提案する積極性も、持ち味の1つだ。専門はプログラミング言語とソフトウェア工学で、ヒトのプログラミング能力の科学的解析もテーマの1つだとか。技術だけでなく、文理融合で社会的な課題解決を意識するNIIらしさは、しっかり浸透しつつあるようだ。

NEW TREND OF SOFTWARE ENGINEERING

特集

「ソフトウェア工学の新しい潮流」

革新的な「双方向」ソフトウェアの開発

長年、ソフトウェア開発において、大きな課題となってきたのが生産性の向上である。従来は、ソフトウェア開発の工程で何か変更が生じた場合、上流まで遡って修正しなければならず大変な労力を要する。実際には、そのような遡及が行われないまま進行してしまうため、設計と実装が乖離するケースが多くみられた。そこで、胡教授らは、各所の修正を全工程に同期させる画期的な手法を開発、ソフトウェア開発を大きく変えようとしている。この「双方向モデル変換の言語的基盤技術に関する研究」とはどのようなものか、話を聞いた。



胡 振江
Zhenjiang Hu
アーキテクチャ科学研究系教授

ソフトウェア開発を単方向から双方向へ

胡 NIIでは、「グランドチャレンジプロジェクト」といって、次世代の情報学が立ち向かうべきさまざまな難問に取り組んでいます。私たちの「双方向モデル変換の言語的基盤技術に関する研究」は、その中のプロジェクトの1つです。これは、私をはじめNIIの研究者が中心となり、東大、電通大、芝浦工大、北京大学、上海交通大学などさまざまな機関の研究者が参画する、国際的なプロジェクトです。

この研究のキーワードは「進化」。ソフトウェアをただつくって終わりというのではなく、刻々と変化する社会に対応できるようなソフトウェアをつくり、変化に伴って進化させようというのが、この研究の狙いです。そのために必要な技術の核となるのが「双方向」です。すなわち、ソフトウェア開発において、フィードバック機能をもつ技術の開発を目指しています。

たとえば、何かレポートを作成して上司に提出すると、もっとタイトルを大きくとか、言い回しを変えてとか、赤字が入ってきますよね。それを修正して、提出し直すわけですが、それと同じことをプログラムの世界でもできないか、と考えました。これまで、ソフトウェア開発は上流から下流まで単方向で進められてきましたが、ここでは「双方向」ということで、下流の工程で修正を加えると、上流の設計のところまで、自動的に修正されるように、ループさせることでプログラムを進化させたいと考えています。



日高 宗一郎
Soichiro Hidaka
アーキテクチャ科学研究系助教

ところで、従来のプログラミングでは、指令の流れは一方向で、上流に遡ろうとすると、復路用にプログラムをもう1つ書く必要がありました。しかし、往路と整合性をもたせながら2つのプログラムを書いたり、人手で遡って変更したりすることは非常に困難です。そこで私たちは、進化するソフトウェアを開発するための基盤技術を構築し、さまざまな場面に応用したいと考えているのです。

——どのような考え方によってつくられたのでしょうか？

日高 ソフトウェアのプログラミングには言語が必要です。ところが従来のJavaなどの言語は、ほとんどが単方向のものでした。そこで、双方向に計算できるような言語を新たにつくろうと考えました。そもそもプログラミング言語というのは、計算を表すものです。そして、小さい計算をいくつも組み合わせ、大きな計算に組み立て、コンピュータに命令を与えます。そこで私たちは、この小さな計算の部品それぞれに双方向性をもたせることができれば、それを組み合わせて双方向のやり取りができる枠組みをつくれるのではないかと考えたのです。そのためには、いかにきれいに小さな部品に切り分けて、わかりやすく記述するか、というところがポイントになってきます。

加藤 実はソフトウェアの双方向変換については、1980年代から、重要性が指摘されてきました。しかし、それをどのように開発すればいいのか、具体

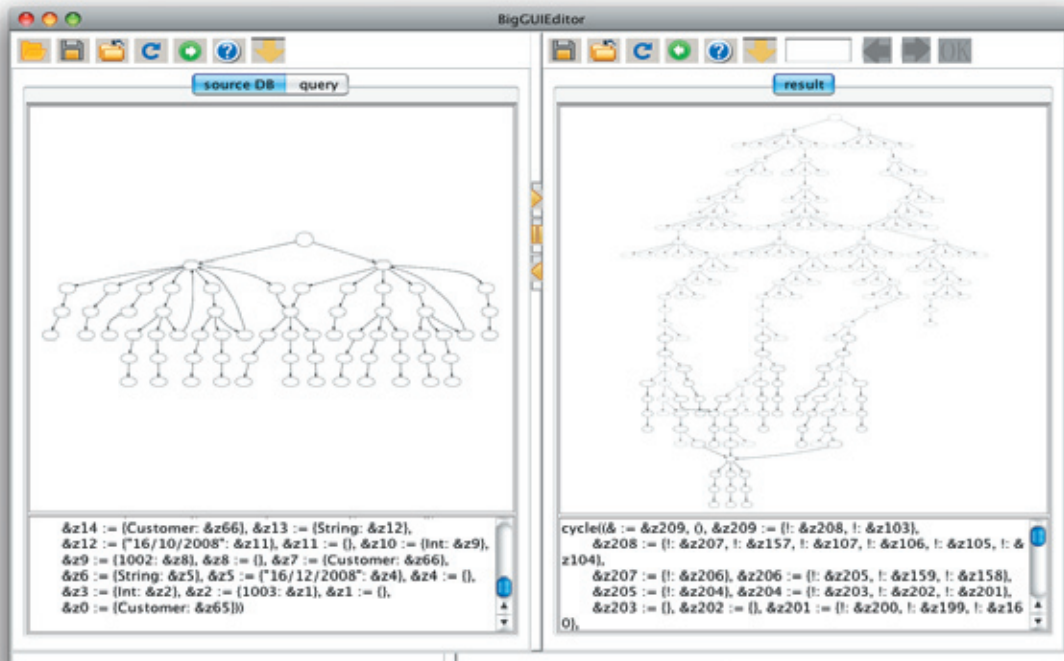


図1 複雑なグラフの変換例のパソコン画面。左のグラフを変換したものが、右のグラフとして表示される。

的な方法論を見出せないまま試行錯誤してきた歴史があります。80年代の成果としては、大きなデータベースに変更を加える際、変更したい部分だけを切り出して修正することができる“View updating”という手法の開発を挙げることができます。そして、2000年以降、双方向の研究が本格的になり、たとえば米国ペンシルバニア大学では、Web上のスケジュール帳に変更を加えると、パソコンやPDA、携帯電話など、違うデバイスでアクセスしても同一の情報が得られるという、同期のための言語を開発しています。

胡 私は2年前に東大からNIIに移ったのですが、私が所属していた武市研究グループでも、ドキュメント(プログラム開発の際につくる仕様書など)のための双方向言語の開発に取り組んでいました。これは、ホームページ(HP)のメンテナンスなどに利用できる仕組みです。HPに変更を加えたい場合、いちいち専門業者に依頼するのではなく、誰でもHP上で簡単に修正を加えることができれば便利ですね。これを実現するのも双方向の変換システムです。

ちなみに、従来のプログラム言語はほとんどが、数列やデータベース、あるいは「木構造」を対象としてきました。木構造とは、ループをもたない階層構造のことです。そして、これまで開発されてきた双方向変換システムも、木構造上での計算を対象とするものがほとんどでした。ところが実際の社会をみると、上司と部下の関係にしても、電車の路線に

しても、双方向にやりとりするネットワーク構造になっています。そこで、私たちは「グラフ」を対象にすることにしました。ここで言うグラフというのは、統計などで使う折れ線グラフやレーダーチャートといったものではなく、点(ノード)とそれを結ぶ線(エッジ)で構成されるものです。木構造と違って、ノードを共有していたり、ループを形成していたりするのがグラフの特徴で、グラフを対象とする双方向変換機構に取り組んだのは、私たちが初めてで、この研究分野の先駆けといっているでしょう。

双方向性を実現する変換言語の開発

——具体的にはどのような手法で駆動させるのですか？

胡 ソフトウェアを開発する際、段階を踏んで実行可能な命令を表現していくのですが、その段階ごとの中間成果物(ソフトウェアの仕様/設計図)を、「モデル」を使って表現します。モデルというのは、ある対象を理解したり、操作したいときに、その対象の本質を定められた方法で表現するものをさします。そのモデル化には、モデリング言語UML(Unified Modeling Language: 統一モデル言語)などを使います。この言語から、グラフィカルな記述で抽象化したシステムのモデルを生成します(図1)。つまりここで言うモデルとは、グラフを意味しているのです。

さらに、モデルから次のモデルへ、また次のモデ



加藤 弘之
Hiroyuki Kato
コンテンツ科学研究系助教

NEW TREND OF SOFTWARE ENGINEERING



ルへと変換していき、最終的にはソフトウェアを実行できるコード(グラフの形)を生成することができます。そして今回、私たちが開発したのが、このモデルの連なりに双方向性をもたせるため、逆方向の変換を可能にする双方向変換言語

「UnQL+」です。

加藤 この言語は、既存の問い合わせ言語(コンピュータのデータに対して問い合わせをするための言語)を拡張したものです。つまり、グラフの情報を取り出すための言語を、グラフを別のグラフに変換するための言語へ拡張したわけです。ちなみに、その文法はSQL(問い合わせ言語の1つ)の構文に非常に近いものになっています。

この双方向変換言語を用いれば、実装の段階で変更が生じて、設計や顧客の要求分析まで遡って自動的に一貫性をもたせて修正することができ、ソフトウェアを進化させることができますようになります。モデルを使っているので、ソフトウェアの設計は実際のアーキテクチャと分離されているため、実装技術やアーキテクチャが変化しても、モデルを再利用することができるというわけです。

日高 では、実際にデモをお見せしましょう(図2)。今、画面の左右にモデルが出ていますが、このように、右のモデルで修正したい個所をクリックして、修正を加えると、対応する左のモデルの変更個所が赤く表示されます。どこをどう変更したかが、一目でわかるかと思います。実際のプログラムでは書かれるモデルは膨大ですので、このように一瞬で変更個所が同期して示され、目で見てわかるというのはとても重要なことなんです。ちなみに、下に見えている文字列が、上の図を双方向変換言語で表現したものです。今、一瞬で変換できましたが、実は最初に開発したものに比べ

ると、そのスピードは100倍以上速くなっています。

——具体的にどんな場面に当てはめて考えることができるのでしょうか？

加藤 たとえば、ソフトウェアを開発する際に、最終的にテストをしますね。そこで不具合が見つかった場合、修正するわけですが、その修正を設計書まで遡って修正しておけば、その設計(モデル)をほかのプログラムに再利用できるようになります。たとえば自動車の設計にしる半導体の設計にしる、テストを経て修正した部分の仕様について、設計まで遡って情報を共有することはとても重要ですが、これまでは人手でやるしかありませんでした。特に、大規模なソフトウェア開発の場合は、設計と実装を手がける人が違っていることが多いので、情報を共有する上でも、後で設計を再利用する際にも、こうした自動的な双方向の同期の仕組みが必要になってくるのです。

胡 一言で双方向と言ってもいろいろな段階があり、たとえば、何か不具合が起こったときに、その原因がどこにあるかを上流まで遡って原因の個所を示す、すなわちトレーサビリティ機能もその1つですし、あるいは、どこかを変更しようと思った際に、どう変更したらいいのかヒントを与えたり、自動的に一貫性をもたせて同期させて変更をするといった機能もそうです。私たちの仕組みを使えば、さまざまなレベルで双方向性を実現することが可能です。

日高 プログラムを書くときに、誰か天才的なプログラマーがいきなり書き始めるということは稀で、通常はプログラミングにかかわる全ての人が理解できる共通の方法で、順を追って設計から実装まで進められていくわけです。また、世の中には似たようなプログラムが無数にあるので、双方向変換によりシステムの一貫性を保証することができれば、ソフトウェア開発のプロセスを再利用することができ、高信頼化、生産性向上に貢献できるようになると思います。

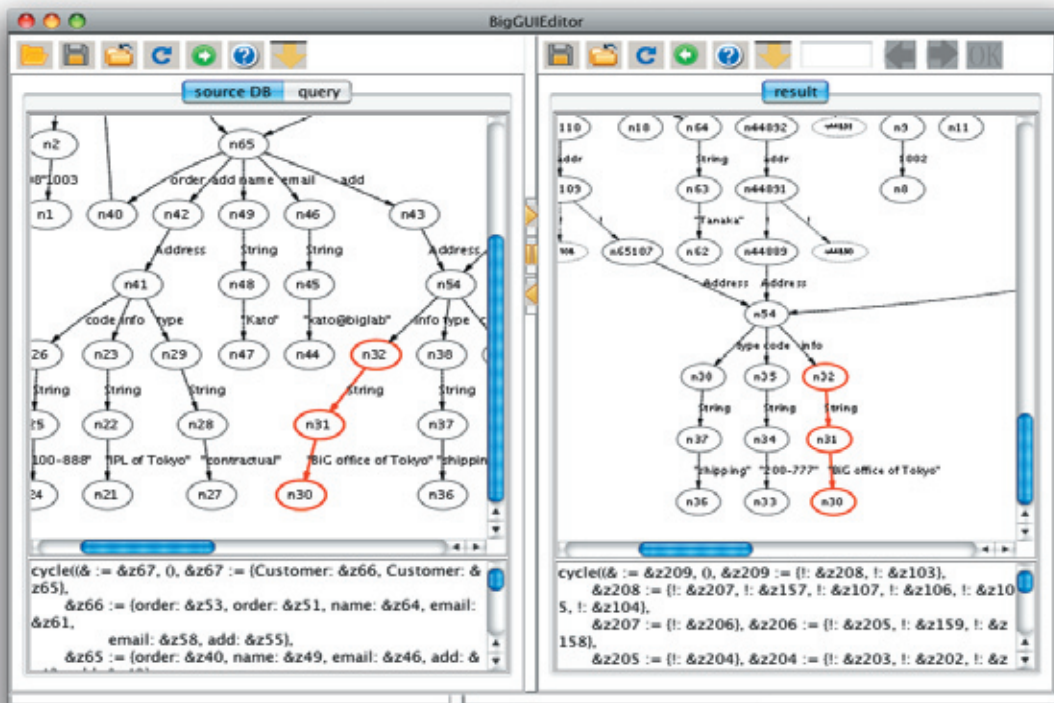


図2 ノードの対応関係を赤で表示したもの。右の画面で修正すると(赤い部分)、左の画面の対応箇所が赤く表示されるため、一目で修正箇所が特定できる。

注目を集める双方向変換システム

胡 私たちのこの研究は、ソフトウェア工学では世界で最も権威のある学会 [ICSE2009 (New Ideas and Emerging Results Track)、FSE2009] で発表されたり、北京大学からNIIにインターンとして来ている学生の論文がRuntime Model Workshopでベストペーパーに選ばれたり、注目を集めています。2008年には、NII内のソフトウェア工学に関する研究センター「先端ソフトウェア工学・国際研究センター (GRACE)」主催の、「双方向変換に関するGRACE国際集会」が湘南国際村センターで開催され、世界各国から、双方向変換で活躍している研究者が30名ほど集まり議論をする場を提供しました。このように、新しいコミュニティの形成において、私たちがリーダーシップをとっているということも、非常に重要なことです。

さらに、私たちの研究をより多くの研究者の方に知ってもらおうと、プロジェクトのホームページ (<http://www.biglab.org>) 上に、これまでの論文やシステム、ソースコードを全て公開しているだけでなく、先ほどお見せしたように、実際にこの双方向変換システムを動かせるデモを用意しており、誰でも簡単にテストすることができます。最近では、研究機関や企業からの接触もあり、自動車の部品メーカーとディスカッションが始まったり、センサーネットワークの研究への応用が始まったりしています。

——今後の展望

日高 国際会議へ投稿したり、発表したりするたびに、さまざまな研究者から意見や批判、助言などをいただき、改良を加えることができたのは、大変よかったと思います。今後はさらに、このシステムの振る舞いの良さを詰めて、どういう性質をもっているのか、見極めていけたらと思っています。

加藤 私は、この仕組みをより効率よく、より速く動かすための改良をしていくと同時に、何でもできるシステムというのではなく、こういうことに向いているよ、と弱点も含めて示していきたいですね。そうすることで、本当に使いやすいものができるのではないかと思います。

胡 ソフトウェアのメンテナンスというのは、永遠のテーマであり、もっともコストがかかる部分なんです。さらに一度つくったソフトウェアに新しい機能を付け加えるというのは、たいへん困難です。従来、私たちの研究チームは、プログラミング言語の開発や最適化、プログラムのつくり方についての理論を研究してきました。この研究は言語自体を一から開発し、ソフトウェア工学の新しい分野を切り拓くという壮大な構想であり、とてもチャレンジングな研究です。だからこそ面白い。メンバーは徹夜続きで大変ですが、今後、応用事例を増やしていつか、世の中にこのシステムの有用性を示していけたらとても嬉しいですね。

(取材・構成 田井中麻都佳)

コンピュータの限界に挑む制約プログラミング

情報学の進展に対応しさらに後進を育成するために2005年、NIIと早稲田大学は互いの人材が有効に交流できる協定を結んだ。この協定のもとで、最先端のプログラミングの共同研究はどのように進められているのだろう。



上田和紀
Kazunori Ueda
早稲田大学理工学術院教授

高いところからボールを落とすと、ボールは徐々に速度を増しながら自由落下する。そして、床にぶつくと急に向きを変えて跳び上がる。それが何回か繰り返されると、いつしかボールは勢いを失って止まる。これは、誰もが知っている物理現象だ。落下するボールが描く軌道や速度計算は、高校の物理の教科書にも紹介されている。しかし、ボールの動き全体をコンピュータで正確にシミュレーションするのは簡単なことではない。

世の中は意外に複雑なもの

「ボールの自由落下はまだ分かりやすい方です。ビリヤードで3つ以上の球が同時に衝突する現象を、コンピュータで解析するのは恐ろしく難しいのです」と話すのは早稲田大学理工学術院の上田和紀教授。ボールの落下やビリヤードの球の運動は、身近でよく目にする物理現象だが、コンピュータにとっては難問だ。このような物理現象を詳細に観察すると、一定の運動を続ける“連続的”な要素と速度や向きが急に変わる“離散的”な要素が含まれている。コンピュータは“連続(アナログ)”も“離散(デジタル)”も扱うことができる。しかし、コンピュータはこの両方を同時に扱うことは、これまでほとんどなく苦手としている。

ガソリンエンジンと電気モーターのように異なる2つの動力を搭載する自動車を“ハイブリッド車”というように、情報学では“連続”と“離散”を同時に扱うことを「ハイブリッド」と呼んでいる(図1)。今、世界中でコンピュータを使ってハイブリッドの問題を解こうという動きが起きている。NIIでも2004年頃から、アーキテクチャ科学研究系の細部博史准教授が、

上田教授とともにこの新しい分野の研究に取り組んでいる。

「分からない」といわない

ハイブリッドの問題に取り組む手法はいろいろあるが、2人は「制約プログラミング」を使おうと考えている(図2)。「制約」とは、物事が実現可能であるための“条件”をコンピュータが扱いやすいように記述したものだ。コンピュータはあいまいな値を扱うのは得意ではない。しかし、世の中の現象についてはむしろはっきりした情報の方が少ないかもしれない。

一例として、野球のピッチャーが投げたボールがストライクゾーンに入るかどうかを考えてみよう。それには、ボールの速度、投げ出される角度、ボールにかかった回転など多くの要素を考えなければならない。投げ出される速度に関しては、遅い場合もあれば非常に速い場合もあり無限の可能性が考えられる。だからといって「分からない」と諦めてしまうのでは、シミュレーションは成り立たない。そこで、時速140~150kmの範囲で投げるという制約を設ける。そうすれば、限られた速度の範囲ではあるが、ストライクゾーンに入るために投げ出される角度やボールの回転が計算できる。現実でもピッチャーが時速200kmでボールを投げることはないの、速度に条件を設けることに問題はない。むしろ、答えが得られる意味の方が大きい。このような制約プログラミングの考え方は、情報学のさまざまな分野で応用されている。

「制約プログラミング」がつかないで縁

コンピュータが誰にでも簡単に使えるようになった理由の1つに、1990年代のグラフィカル・



細部博史
Hiroshi Hosobe
アーキテクチャ科学研究系准教授

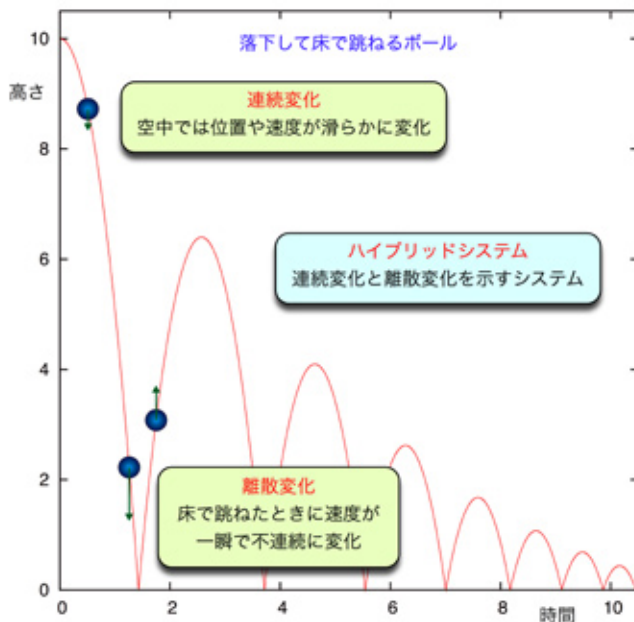


図1 ここでの「ハイブリッド」とは、連続変化と離散変化の両方の要素を扱うもので、落下して床で跳ねるボールの動きなどは「ハイブリッド」問題といわれる。

「落下して床で跳ねるボール」の制約プログラミングによる記述 ハイブリッドシステム向け制約プログラミング言語 HydLa を用いた場合		
INIT	$\Leftrightarrow ht = 10 \wedge ht' = 0.$	// 高さ ht と速度の初期値
LAWS	$\Leftrightarrow \square (g = 9.8 \wedge c = 0.8).$	// 重力加速度 g と反発係数 c
FALL	$\Leftrightarrow \square (ht'' = -g).$	// 通常は重力加速度に従う
BOUNCE	$\Leftrightarrow \square (ht = 0 \Rightarrow ht' = -c \times (ht' -)).$	// 床に当たったら反発
INIT, LAWS, (FALL << BOUNCE).		
// 上記ルール合成		
制約とは物事が実現可能であるための条件 この例では物理法則が制約となる		
曖昧な条件も同様に扱える 例えば INIT で $ht = 10$ の代わりに $9.5 \leq ht \leq 10.5$ としてもよい		

図2 落下して床で跳ねるボールの動きのようなハイブリッド問題を扱う場合、「制約プログラミング」の使用が1つの方法である。

優秀な学生を育てなければなりません」と上田教授。互いの連携強化のためにNIIと早稲田大学は2005年に協定を結んだ。その枠組

ユーザー・インターフェース(GUI)の普及がある。GUIとは、表示されたアイコンやメニューをマウスなどでクリックするだけで、思い通りにコンピュータを動かすことができる操作画面のことだ。現在、WindowsやMac OSでは当たり前になっている。細部准教授は、学生時代からGUIの研究をしてきた。アプローチ方法として選んだのが制約プログラミングだった。制約プログラミングを使うと、操作画面上に図や文字などを重なったりずれたりせずに配置させることができる。

この経歴が認められて、2002年にはフランスのバリ第6大学のフィリップ・コドニエ教授とNII情報学プリンシプル研究系の佐藤健教授が進めていたプロジェクトに誘われた。これをきっかけに、現在でもフランスの大学との共同研究を積極的に進めている。

一方の上田教授は、「25年の制約プログラミング研究でわかったことは、通信、つまり情報のやりとりをきれいに記述するのに役立つということです」と話すほど、この分野に精通している。「2004年にフランスのナント大学の研究者と共同研究を始めるときに、チームの強化のために上田教授に加わってもらいました」と細部准教授。以来、2人はさまざまな研究に取り組んできた。そして現在は、「ハイブリッド」の研究に力を入れている。

「育てること」を大切に

「プログラミングにはマンパワーが必要で、

みの中で細部准教授は早稲田大学の客員准教授に、上田教授はNIIの客員教授になった。おかげで、細部准教授が週1回は必ず早稲田大学でディスカッションに参加するほど頻繁な交流が行われるようになった。NIIの学生にとっても、早稲田の学生にとっても多くの教員と関わり指導を受けることは重要で、「博士課程の学生が頑張ってくれたから、研究が進みました」と細部准教授は学生が育っているのを実感している。

正しいことを保証できるソフトウェア

「ハイブリッド」に制約プログラミングを応用すると何ができるようになるのか。リアルに動くアニメーション、自動車の動作や生命現象のシミュレーションを行うソフトウェアが作れるだろう。中でも、2人が目指しているのは「正しいことを保証できるソフトウェア開発」だ。「現在、有名なソフトウェア製品で計算結果を保証しているものはありません。信じられないかもしれませんが、それは技術的に難しいことなのです」と上田教授はソフトウェアのあり方に不満をもっている。そして、「ハイブリッド」と制約プログラミングを組み合わせることで、条件付きではあるが結果を保証できるようになるという。情報学の世界の「信頼度」は格段に高くなるだろう。上田教授、細部准教授そして学生たちによって安心して使えるソフトウェアが提供される日が来るのを期待したい。

(取材・構成 池田亜希子)

ソフトウェアの信頼性向上を目指す研究会が発足

NIIは2009年12月、NTTデータ、富士通、NEC、日立製作所、東芝の日本を代表するIT関連5社とともに、「ディペンダブル・ソフトウェア・フォーラム(DSF)」という研究会を発足させた。DSFは、障害を起こさない、信頼性と安全性の高いソフトウェアの実現を目指して研究開発を進めていく。



中島 震
Shin Nakajima
アーキテクチャ科学研究系教授

ディペンダブル(信頼できる)なシステムを作るためには、アプローチが2つある。1つが、作り手である人間の作業プロセスを綿密に管理しようというもの。もう1つが、設計の正しさを自動的に確認するツールを用いるもので、その代表が「形式手法(*)」と言われるものだ。DSFでは、この形式手法をソフトウェア開発に適用する方法を検討するワーキンググループを立ち上げており、2012年3月までにガイドラインの策定などを行う予定である。

形式手法でソフトの正しさを検証

形式手法は、1970年代にヨーロッパで研究が始まったソフトウェアの開発手法である。数理論理学に基づいた科学的な裏付けをもって、高い信頼性を保証することができる。特に大型ソフトウェア開発の要となる上流工程の仕様を、数学理論を用いた言語によって厳密に記述することにより、系統的に正しさが検証できるようになる。

2009年3月、基盤ソフトウェア技術戦略に関する産官学協力委員会から出された『安心・安全な社会を実現する基盤ソフトウェア作りを目指して』という提言書には、形式手法を産業界で用いてソフトウェアの信頼性と生産性向上を図るために、まず調査研究から始めようという趣旨が盛り込まれている。それを受けた形で、NTTデータが事務局となり、技術開発本部副本部長の木谷強氏が中心になって各社に直接働きかけ、今回の連携の枠組みが出来上がった。

アメリカでは、大統領の諮問機関でIT戦略の中核となるNITRD(Networking and

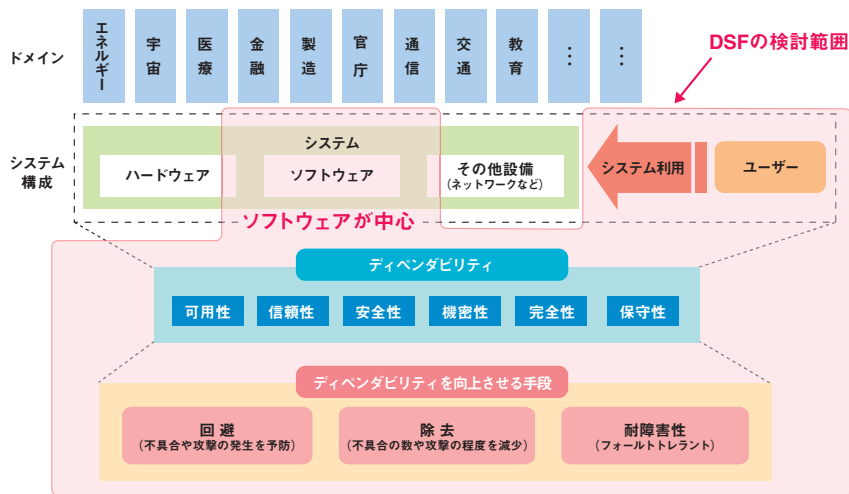
Information Technology Research and Development)のまとめた2007年のレポートで形式手法に言及している。一方、ヨーロッパでは、EUが策定した第7次研究枠組み計画(FP7、2007~13年)の中で、ソフトウェア研究投資の多くが形式手法に振り向けられている。こうした欧米の動きに対して、日本も国際競争力を維持するために、手をこまねいてはられない状況にある。

NIIアーキテクチャ科学研究系の中島震教授は、「研究者として、形式手法の面白さや重要性を感じていたが、私たちがこう使えばいいと言うのでは意味がない。今回、NTTデータが先導して求心力を発揮し、産業界が自分たちで形式手法に取り組むことの意義は大きいのです」と語る。NIIのミッションには、大学や産業界とのコラボレーションも掲げられており、DSF内では学術的な方向性をまとめるために力を発揮したいという。

上流工程の厳密管理で作業量、コストも削減

形式手法にはいくつかの記述法があり、いずれもヨーロッパで開発された。近年は安定度を増しており、中には日本語対応のものもある。ただ、これを使いこなすには前提として、集合論や数理論理学などの基礎を学び、記述そのものも勉強しなければならない。導入前の勉強時間が必要で、最初は取っつきにくさもあるため効率が落ちる場合もある。しかし、理科系の学生なら、大学1~2年で学ぶ程度の数学の素養があれば十分で、「少し難しそうな教材を使うと学ぶ人は努力するので、より理解

(*)形式手法(Formal Methods): 数学を基盤とした正確な仕様の記述と仕様の検証などの技術を含む手法の総称である。正確に仕様を記述することにより、ソフトウェアの不具合に気づくことができる。



DSFの検討範囲 DSFでは赤枠で囲まれた範囲についてソフトウェアを中心に検討する。DSFではディペンダビリティを向上させる手法の調査研究を企業の枠を越えて進めていく。

が深まります」(中島教授)という。

また、上流の設計段階で漏れを確実になくしておけば、下流のプログラミングや試験段階での問題発生が抑えられ、トータルでは作業量もコストも大幅に削減できるというメリットがある。その結果、人海戦術に頼っている現在の開発体制を改めることができる。基本的な技術開発を行ったヨーロッパにとっても、日本の開発体制の中でこの形式手法がうまく適用できれば、自分たちが作った技術の普遍性の裏打ちができることになり、関心が寄せられているという。

実際にどのような課題に対してどのように形式手法を使うと、どんなメリットがあるかということは、まだ十分に解明されていない。大型システムであれば、形式手法を用いた開発には時間がかり過ぎるため、ソフトウェア全体に適用することはないが、部分的にここは使うべきだという個所を突き止めることは有用である。そこで、具体的な手順として、2010年3月までは各社持ち回りで、過去に開発経験のある図書館の書籍予約システムについて形式手法を用いて検証をしている。従来のやり方で設計した経験のある例題であれば、形式手法を使った場合の違いが分かりやすい。得られた知見は、月1回の例会で報告し共有している。4月以降は、もう少し本格的で実践的な課題に適用する実証実験を行う予定という。

競争関係を越えて ソフト業界の発展を期す

形式手法を広めていくには、1社だけの取り

組みでは限界がある。また、大手の会社のソフトウェア開発には自社だけでなく、多くのソフトウェアハウスが参加しており、将来はその開発者のすべてに形式手法を身につけてもらう必要がある。その裾野を広げるには、“オールジャパン体制”に期待がかかるが、研究会は現在まだ手弁当の集まりであり、欧米のように国家が後押ししているわけではないという課題とも向き合っている。

木谷氏は、「私たちメーカーはもちろん互いに競争相手でもあるが、皆で知恵を出し合って日本のソフトウェア業界を発展させていかななくてはならないと考えて立ち上がりました。事例を重ねて、形式手法のメリットが共有されてくれば、産業界にも積極的にこれを使おうという気運が広がるはずですよ」と語る。

実証実験の段階では、さらに取り組むべき作業が増えてくると考えられており、その負担を分担するために、参加メンバーを増やすことも視野に入れている。2012年3月以降、ニーズがあれば組織をさらに継続していく可能性もある。木谷氏は「成果を世に問いながら、人命にかかわったり、生活に大きな影響を及ぼすような重要なシステムについては、できるだけ形式手法を使って検証するという流れを作りたいのですね」と抱負を語る。

中島教授は、「日本を代表する企業が集まったDSFで、形式手法を調査研究して道筋を定めることができなければ、たぶんもう日本では、形式手法が根付くことはないかもしれない」と、その責任と期待の大きさをかみしめる。

(取材・構成 塚崎朝子)



木谷 強
Tsuyoshi Kitani
NTT データ技術開発本部副本部長
ソフトウェア工学推進センター長

トンデモ借り物競走

中島 震（国立情報学研究所アーキテクチャ科学研究系教授）

蜘蛛くもの巣、格子、雲、と書いていくと何のことかさっぱりわからないが、カタカナでウエブ、グリッド、クラウドと書くとうまくソフトウェアのことだと気がつくかもしれない。このように、ソフトウェアの分野では、新しい考え方が出てくると用語をどこからか借りてくる習慣がある。わかりやすいと思って言葉をお借りするのもいいが、思い描くイメージは人それぞれなので、とんでもない誤解が生じることもある。

キーワード工学にとつぷり

似た技術や概念なのに、新しいという印象を与えようとして言葉を選ぶ。新しい言葉を当てはめたり、より魅力的な言葉を探してきたり、早いもの勝ちの借り物競走のようだ。人呼んで「キーワード工学」という。逆に、モデルというように同じ言葉が異なる意味で使われることもある。偶然の場合はまだしも、故意というか、思い違いの利用という企みさえ感じられることがあって、トンデモ・キーワード工学かもしれない。混乱するのは一般の方ばかりではない。

ソフトウェア技術者も煙に巻かれる。

ソフトウェアの世界はキーワードの移り変わりが早い。実務で接した技術の正体を知りたいと博士課程の扉を叩く社会人の方が増えている。普遍的な基本技術に立ち返りたい、それをベースに実務で役立つ技術発展に貢献したい、と意欲的な人が多い。



入学して、トンデモ・キーワード工学に浸っているのに愕然がくぜんとすることさえある。

トンデモ世界からの脱出は大学院で

日常生活は目に見えないソフトウェアで囲まれて

いる。ケータイや銀行のATM端末はもとより、通勤時に使う鉄道の改札まで、ソフトウェアの不具合で大変なことになる。期待通りの機能、性能を持つソフトウェアの開発技術は不可欠で、ソフトウェア工学の重要性は高まるばかり。ソフトウェアに不具合がないことを自動的に調べる方法が長い間研究されていて、モデル検査という技術に関心が集まっている。これを大学院で学びたいという人が増えていて、それは素晴らしい。

モデルという言葉はトンデモ世界の代表で、モデル検査のモデルが慣れ親しんだ意味と異なることを知ってびっくりする。日常の言葉だと、ファッションモデルとか、お手本や代表例のことをモデルという。数式で表現した解析モデル、図形記号（アイコン）や本物に似せた作り物もモデルと呼ぶ。ソフトウェア工学ではモデルベース開発という言葉づかいもある。モデル検査のモデルが、論理学の用語だとわかると、ようやく気分がすっきりする。

で、最後に、トンデモついでに、「モデル検査って、モデルさんの健康診断かい？」

情報から知を紡ぎだす。

NII

国立情報学研究所 ニュース (NII Today) 第47号 平成22年3月

発行：大学共同利用機関法人 情報・システム研究機構 国立情報学研究所 <http://www.nii.ac.jp/>

〒101-8430 東京都千代田区一ツ橋2丁目1番2号 学術総合センター

編集長：東倉洋一 表紙画：小森 誠 写真撮影：由利修一 デザイン：鈴木光太郎 制作：サイテック・コミュニケーションズ

本誌についてのお問合せ：企画推進本部広報普及チーム TEL:03-4212-2135 FAX:03-4212-2150 e-mail: kouhou@nii.ac.jp