■中島震　アーキテクチャ科学研究系　教授

【タイトル】
ソフトウエアの信頼性を高めるための形式検証技術

【本文】
かつては、限られた専門知識を持つオペレーターのみが大型計算機を使いこなしていましたが、今やそこで動いていたのと同じような規模のソフトウエアが携帯電話にも搭載されるようになり、一般の人が意識せずに使うようになりました。家電などに用いられる、いわゆる組み込みソフトウエアは文字通り桁違いに普及しており、もし不具合があれば影響力も大きくなります。そこで、信頼性の高いソフトウエアを作りたいというのが、私の研究の目的です。

数理科学に裏付けられた客観的な開発手法

正しいソフトウエアを作るためには、開発にかかわる人々の作業進捗を管理して品質を高めるという方法もありますが、もう少し厳密かつ客観的に、数理科学を応用してソフトウエアを開発しようという「形式手法」について研究しています。

「形式手法」によれば、まず、設計仕様を記述する際に曖昧さを排することができるので、ソフトウエアに本当に必要な機能が明確になります。次に、論理的な誤りを系統的に検証して、バグが自動的に検出できるようになります。

「形式手法」のひとつに、アメリカのエドムンド・クラーク博士たちが 1980 年代に開発した「モデル検査法」という技術があります。当初はハードウエア関連に用いられていましたが、90 年頃からソフトウエアの検査にも応用が試みられるようになり、私はその実用化に力を注いできました。

モデル検査法では、ソフトウエアの初期状態から処理が進行するのにしたがって、それぞれの実行経路におけるエラーの有無を網羅的に解析していきます。無限には調べられないので、あらかじめ調べる範囲が決められていなくてはなりません。どういうソフトウエアであれば適用できるかを見極めるために、対象となるソフトウエアを“抽象化”あるいは“モデリング”することで、ツールを使えるような規模に変換していきます。この変換がうまくいけば、有限の手順でバグを検出できるようになります。

限界を知りつつ実用化を探っていく

設計工程で効率的な検査ができれば、不具合のまま出荷されるという事態が回避でき、開発コストも抑えられます。ただし、ソフトウエアは継続的に改良され、機能も拡張し続けているという面での難しさがあります。また、ソフトウエアが正しく動いてもハードウエアとの相性からシステムエラーが生じたり、例えば気象条件などハードウエアの置かれている状況からエラーが誘発されることもあります。このため、「形式手法」によっても 100%不具合がないと保証することは不可能であり、過大な期待は禁物です。しかし、限界を知りつつ、ここまでは確実と言い切れるようなところまで実用化するのは重要なことです。

新しい何かを作り出すわけではないので、地味で効果も見えにくい技術ですが、自動車や航空機、さらには交通や金融システムなど、あらゆるものにソフトウエアが組み込まれるようになっており、その信頼性確保は、社会の安全にとっても不可欠です。

（取材・構成　塚﨑朝子）