

## 「新世代ソフトウェア開発ツール」セッションテーマ紹介

松下 誠<sup>†</sup> 権藤 克彦<sup>††</sup>

本稿では、どのような考えから「新世代ソフトウェア開発ツール」セッションを企画するに至ったかについて述べ、予定される本セッションの討論テーマについて紹介する。

### Session Topics of “New Generation Software Development Tools”

MAKOTO MATSUSHITA <sup>†</sup> and KATSUHIKO GONDOW<sup>†</sup>

In this paper, we introduce why this session, “New Generation Software Development Tools,” is planned to open in this workshop. Themes to be discussed with is also described.

#### 1. はじめに

ウィンターワークショップでは、過去 3 回に渡って、ソフトウェアプロセスに関する議論を行ってきた。具体的には、2001 年の「プロセスと方法論」、2002 年の「ソフトウェアプロセス」、そして昨年となる 2003 年の「軽量ソフトウェアプロセス」セッション<sup>1)</sup>である。本セッションは、これまでのプロセスに関する議論の延長としつつも、ツールに特化した議論が必要ではないか、という古くて新しい問題意識に基づいて企画されたものである。

以下、昨年の議論を取りあげて過去の議論をふりかえった上で、今年どのような議論を行いたいと考えているかについて述べる。

#### 2. これまでの経過

##### 2.1 前回のワークショップ

昨年の議論は主に XP (eXtreme Programming) に関する物であった。XP の考え方は、多くの組織において広く普及しており、実際の取り組みも行われているが、昨年はその中で XP の良い点を生かすための方法や、実際に開発をした結果得られた結果についての評価などについて議論を行った。議論の経緯などについては、昨年のワークショップ報告を参考にしてい

ただきたい。

この議論の中で、ソースコードとドキュメントの相互関係をいかに明確にするか、という点が今後の課題としてあげられた。仕様とソースコードの対応づけ、という話題をとりあげるまでもなく、この相互関係を明確にすることは重要であることは論を待たないが、XP の場合は比較的開発の早い段階でソースコードを記述しはじめる傾向があるため、ソースコードの理解を助けるための可視化や、整合性を確保するための支援環境などが必要ではないか、という指摘が行われた。

##### 2.2 オブジェクト指向 2003 シンポジウム

そこで、この点についてより議論を行うための場として、2003 年 8 月に行われたオブジェクト指向シンポジウムの場を借りて、パネル討論の場を設けた<sup>2)</sup>。本パネルでは、ソースコードを形式的なドキュメント(具体的には仕様書)としてとらえた場合に何が問題なのか、また、いわゆる形式記述言語が用いられないのはなぜか、という点について議論を行った。

この時の議論の中で、例えば何かを行いたいと思った場合に、現状ですぐに使える物として何かがあるのか、その中で目的を達成するために何が必要なのかを判断した結果が現状の方法である、という意見がなされた。これは、現状が必ずしも満足の行く開発環境というわけではなく、例えばソフトウェアによる支援をはじめとした、開発者にとって有用なツールが必要であることの裏返しでもある。議論の場では、書かれたソースコードを何らかの形で形式的に検証したいと思ったとしても、既存の検証ツールが有効に使われていないのではないか、という意見もあった。

<sup>†</sup> 大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

<sup>††</sup> 東京工業大学情報理工学研究所

Graduate School of Information Science and Engineering

### 2.3 オープンソースソフトウェアの台頭

ここ数年、開発されたソースコードを公開する動きが盛んになっている。この動きは個人の趣味の範囲に留まらず、産業界はもちろんのこと、研究機関などにおいても、自らの成果をソースコードの形で公開し、自由に利用できるようにする動きが進んでいる。この動きはさらに「既存の公開されているソフトウェアを利用して別のソフトウェアを作成する」動きに繋がっている。

例えば、IBM によって公開された Eclipse は、本来 Java 向けの統合開発環境として公開されたものであるが、Eclipse の機能をプラグインと呼ばれる追加ソフトウェアによって拡張することができるため、Eclipse を前提としたソフトウェアが生まれてきている。例えば、Java のプログラム解析ツールを Eclipse のプラグインとしてで実装することにより、同等のツールを最初から作成するよりもより早く作成することができ、かつ他の Eclipse プラグインが提供する機能も利用することによってより高度な機能を提供することが可能となってきた。

一方、いわゆるオープンソースソフトウェアの開発環境の多くは既存のオープンソースソフトウェアを用いて構築されている。これらの環境で用いられているソフトウェアは、例えば CVS やメーリングリストなど、従来からある基本的なツールを運用するだけでありながら、大きな効果をあげている。

これらの事例は、単にソフトウェアツールがそれ自体単独で存在するものではなく、他のツールと有機的に結びつくことによって、全く新しい物として捉えることが可能であることを示しているといえる。従来の研究でも、例えば PCTE などに代表されるように、ツール間の共通プラットフォームをトップダウン的に定めた上で全体を構築する試みはあったが、近年の動きは、どちらかといえばボトムアップ的であり、可能な所から複数のツール間の共通化を行っていると言える。

## 3. 討論テーマ

これらの背景を踏まえ、今回のワークショップでは、ソフトウェア開発の際に用いるソフトウェアツールを題材とした議論を行いたいと考えている。具体的には、現状の把握に基づいて問題点を列挙し、将来進むべき方向性について議論を行いたい。具体的には、以下のような論点がある。

### 3.1 ツールインフラストラクチャ

産業分野、学術分野を問わず、日々何らかのソフトウェア開発ツールの作成がさまざまな目的で行われて

いる。一般的に、各ツールは、それ自身が本来持つべき機能を実装した部分と、他と共用可能な部分が存在している。

今、ソフトウェア開発のためのツールに限って考えた場合、「他と共用可能な部分」で既存のソフトウェアとしてどのような物があり、どのような分野に利用できるのか、また、不足している部分にどのようなものがあるか、既存のソフトウェアを利用する場合にどのような問題があるか、共用可能なソフトウェアはどのような特性を持つべきか、という問いを考える。しかし、この問いに対しておそらく明確な答はこれまで提示されていないのではないだろうか。

今回の議論を通じ、まず、ソフトウェア開発ツールにとってのインフラストラクチャのあるべき姿について、おぼろげながらもはっきりさせることができるのではないかと考えている。

### 3.2 ツールの相互利用

UNIX オペレーティングシステム的环境では、単純な機能を提供するツールをパイプやシェルなどによって結びつけることによって、より高度な処理を行うツールを新たに生み出すことが可能である。前節で述べたインフラストラクチャを仮にこのパイプやシェルに例えるならば、もう 1 つの論点として、さまざまな機能を実現するツールと、そのツールをいかに組みあわせるか、という点をあげることができる。

ある個人が、自らが必要とする機能を全て自らの手で実装することは、ソフトウェア自体がより複雑となってしまった現状では現実的ではない。また、ソフトウェア再利用に関する議論を取りあげるまでもなく、他人が作成したソフトウェアを利用する際には種々の困難な点があることも指摘されている。より良いソフトウェアを早く、確実に生み出すためには、各種ツールをどのように利用していくか、というこれまであまり省みられなかった問題にいま一度取り組まなければならないと考えている。今回のワークショップにおいて、この点について何らかの方向性が見いだせるのではないかと期待する。

## 参考文献

- 1) 中谷, 松田, 横山, 藤井, 松下, 平山, 青木, 大木, 佐伯, 綿引: “ウィンターワークショップ in 神戸報告”, 情報処理学会研究報告, 2003-SE-142, pp.45-53 (2003).
- 2) 松下: “パネルディスカッション: 形式仕様記述言語としてのプログラミング言語 - XP を例にして”, オブジェクト指向 2003 シンポジウム論文集, pp.295-296 (2003).