

## Lyee Internet Information

### 対談

「21世紀の新しいソフトウェア・メソドロジーについて」(前編)

カナダ オタワ大学 グレゴール・V. ブックマン教授

ソフトウェア生産技術研究所株式会社 根来文生社長



司会 : アイサム・A・ハミド教授(岩手県立大学)

#### 【グレゴール・V. ブックマン(Gregor V. Bochmann)教授 略歴】

- 1941年 西ドイツのニーブルに生まれる
- 1971年 カナダのモントリオール、マギール大学より理論物理学博士号取得。
- 1972年 モントリオール大学において準教授に就任。
- 1983年 モントリオール大学において教授に就任。
- 現在 オタワ大学情報技術・工学部教授。

< 主な研究分野 >

コミュニケーション・プロトコル、ソフトウェア・エンジニアリング、  
高速ネットワーク、分散アプリケーション、システム管理、  
UML ( Unified Modeling Language : 統一モデリング言語 )

< 主な著書 >

・ Concepts for Distributed Systems Design  
( 分散型システム設計コンセプト ) 他

## 【 新しいソフトウェア・メソドロジー / Lyee と UML 】

- ハミド教授      それではお二人に新しいソフトウェア・メソドロジーをテーマに  
お話をして頂きたいと思います。
- 根来社長      まずは Lyee のメソドロジーの背景にあるものについて話をさせて  
頂きたいと思います。例えば UML (ユニファイド・モデリング・  
ランゲージ) はどういった背景から作り出されたと思われるか、  
そして Lyee はどういった背景から作られているのかといったことですが、  
そういうレベルのディスカッションがこれまであまり行われて  
こなかったのではないかと思います。この事がソフトウェア・サイ  
エンスの中に哲学を持ち込む局面で更に深めなければならない問題も  
も阻害していると思います。いかにその方法論を利用していかかとい  
う事は哲学の問題ではないですが、いかに作られたかについては  
哲学でなければなりません。
- ブックマン教授      しかしながら私達は経験からいろいろな事を学びます。ソフトウェア  
では、いろいろな人が作ったものを見る事で、それが指針や参考と  
なります。他の方法ではどういった事をしているか、この機能をどう  
やって扱っているかという事を見まして、相対的な評価をし、問題の  
分析や適切な解決策の発見に役立てます。と同時に、それが実際に  
使えるという事を可能な限り証明しようとしています。
- ハミド教授      UML はオブジェクト・オリエンテッド分析とデザインにおける経験か  
ら作られ、80年代に出てきた OMT (オブジェクト・マネージメン  
ト・ツール) を修正した形で90年代に出てきました。OMT 自体は

残念ながらそれほど実際に使われませんでした。考え方は非常に新しいものを含んでいます。その後新しい考え方がいろいろ出てきましたけれど、それらは最新のソフトウェア・サイエンスのための革新的な研究課題を定義したという点で非常に重要ではありますが、一般にこうした革新的な概念はいずれも、標準化した方法がなかったり、また一定のレベルの開発志向型統合がなかったりして、実用するのは容易なことではありませんでした。

そこにひとつのソリューションとして CORBA が出てきて、今はその OMG (オブジェクト・マネージメント・グループ) の標準化がひとつの潮流になっています。現在、E コマースなどいろいろな事が行われていますが、結局共通に理解できる標準化がないとやっていけない事にみな気がつくようになりました。そういう中でグラディ・ブーチ (Grady Booch)、ジェームス・ランボー (Jim Rumbaugh)、イヴァル・ヤコブソン (Ivar Jacobson) が現在 UML と言われているものを作ったのだと思います。

私が強調したい事が1つあります。それは UML と Lyee にはある意味で、共通項があるという事です。根来さんは30年にわたり80以上のプロジェクトを指揮されたそうですが、その間には経験に基づいたいろいろなフィード・バックを受けたと思います。そのフィード・バックは、当然従来のソフトウェア開発プロセスに関する不備から出てくるものだったと思います。だからこそ根来さんは普遍的なソフトウェア開発方法に関する新しい考え方を構築したいと考えられたのだと思います。

UML というのは、オブジェクト指向の考えを基盤にしておりますし、その他、OMT などたくさんアプローチがあって、みんな何か新しいものを求めていました。その中で、ブッチらは革新的な考えをもってなんとかしたいと考え、過去のいろいろな方法論を分析して、悪い点を排除し、いい点だけを集めて方法を作ろうとしました。その結果、考え出されたのが UML だったのです。そういう点は Lyee と共通点していると思います。UML 自体はソフトウェア・デザインの代表的な表記法としては、簡単とは言えませんが、そういう発想の基に生まれたものなので、UML に関する評価やリサーチが今も行われています。また、みな、シンプルで、確実に、うまくいく普遍的なソフトウェアの開発方法を見出せないかと新しい考え方を模索しています。

**ブックマン教授**

また、UML はまた別の点で重要だと思います。UML はプロセスではなくて、ノーテーション (notation : 記法) を示すものです。

根来社長

ノーテーションはプロセスで使われるものですが、プロセスを表現する事は大変重要なことです。

また違うものを集めて、処理するという様な事もしますが、どのノーテーションをどの段階で使うかとか、ジェネレーションを使うとか、ツールをどうするかといった事も決めていきます。

それからもう1つ、ある傾向があります。一般に人々は多くの言語を学びたいとは思いません。それはあまりにも大変な事だからです。

私も実際フォーマル・メソッドをコミュニケーションの時に使いました、私自身は、プログラミング言語は得意ですけども、ITU が確立した標準化された言語を使いました。しかしそういった過程の中でも多くの方は、あまり多くの言語は使えないという事は経験的にわかっております。

さらに UML の良い点は、標準化されており、方言が少ないので、みんなが共通語を話している様な感じになるので学びやすいのです。

また、Lye に UML のノーテーションの要素を埋め込むことで、UML の考えを現実に Lye に適用していくことは可能だと思います。

私は過去にソースコードを分析するという事から始めました。私がその作業に着手したのは30年くらい前ですけども、私がある意味で恵まれていたのは、当時世の中に方法論があまりなかったのです。しかし、たくさんのソフトウェアが作られておまして、そういったメソドロジのバックグラウンドがないソースコードを私は徹底的に分析したのです。ですからもっとも人間の素直な気持ちが現れているソースコードを分析した訳です。もちろん世界標準もありませんでしたし、ソフトウェアを作る為のルールらしいルールは何もなかったのです。そういった時代のソースコードを分析できたのは、私にとって大変幸せでした。そういう意味で私はソースコードのなんたるかについては、大変詳しいのです。最近ではいろいろな方法論が出てきて、標準化のルールができてきまして、それを利用してソースコードが作られる様になりましたが、私のはっきり申し上げられる事は、そういった方法論で作られたソースコードも30年前のソースコードも、実態は全く同じだという事です。違いがあるとすれば、プログラム・ランゲージが違うという事です。しかしプログラム・ランゲージというものは、マシン・ランゲージに落とせば、本質的にはほとんど意味を持っていないものなのです。この指摘はとても重要だと思います。

こういう意味で、ソフトウェア・サイエンスはもっと効率的方法で

進化されなければならないと思います。実際、そういった認識の上に立つか、立たないかという事もソフトウェア・サイエンスに関する問題を理解するうえで、私は重要だと思っています。

もちろん UML のような考え方も素晴らしいとは思いますが、私は、今申し上げた視点から、ソフトウェアの実現・ソフトウェアそのものの表現に関し、もっと効率的な方法ないものかと今なお求めています。私が申し上げた物の見方は、今後更に検証されるべきだとは思いますが、ハードウェア・サイエンスに比べてソフトウェア・サイエンスはまだ確立されていないという点で同意が得られるなら、私の考えも納得されることでしょう。

ブックマン教授

確かにソフトウェアはハードウェアほどには効率性が上がったとは言えません。しかし、ソフトウェアの生産方法や、場合によってはソフトウェアの質については、進歩がありました。数年前にある国際学会で、ソフトウェア・サイエンス界で非常に有名なホーワ先生の話聞いたのですが、先生はソフトウェアの世界は進歩しているという講演をされました。それは私達が望むものが一気に出来上がる様な奇跡的なものではなく、段階をおった形で漸次進歩していくというような内容でした。私も講演を聴いていて、それはそうだなと思いました。また他にもフォーマル・メソッドについての講演を聴いたりしましたが、その方もやはり進歩していると話していました。

根来社長

その点に関して、アカデミアの人達と実プロジェクトを経験している人達との間には、見解に相違があるのではないかという感じがします。たとえば、1960年頃アメリカ人が書いたペーパーにそうした見解の違い、実務面とソフトウェア・サイエンスの研究に関するある種のギャップを生み出すということ書かれていました。そこには学会の数々のデータがありまして、研究や実用に関する調査が示されていました。また、ソフトウェア・サイエンスの研究と実用は別々に行われるべきではなく、相互に関係した一体のものであるとの結論が出されていました。そういったペーパーが40年前にアメリカで書かれたという事は大変驚くべきことだと思います。

しかし、1980年代にオブジェクト・オリエンテッド・プログラミングが現れましたが、これは今述べた試みのひとつの例だとは思いますが、まだ、研究と実用の間の違いが残っています。

一方、それより以前に、フランスのワーニングという数学者が提唱した方法がありましたが、言語ではなく開発のメソドロジーなので、そういう意味では本当に初期のメソドロジーでした。こういうメソド

ロジーが出ると、それを使って実開発をやってみるのですが、この方法は非常に優れておりました。この方法が何故、その後使われていないのか大変不思議に思っております。

ブックマン教授 この30年間に非常に面白い、優れた方法論がいろいろ出てきましたが、それだけ出てきながら進歩が乏しく、ほとんど変わっていないという事は、進歩そのものが大変難しいという事を意味していると思います。簡単にできてしまう事はやられてしまいますので。根来さんが今提唱してらっしゃいます方法というのは、過去にいろいろ出てきたさまざまな方法の中で、進歩の難しい状況の中から出てきたと思いますが、他の方法よりも、一段と優れているとおっしゃられる根拠と背景になっているものは一体何なのですか？

根来社長 私は Lyee が大変優れているという立場で議論をした事はありません。私は、Lyee はインテンションを表現する方法だと説明しているだけです。その為には、インテンションは何かと定義する必要がありますが、そのインテンションは少なくともコンベンショナルな世界の人達が言っているインテンションではありません。Lyee の世界観は従来のな世界観とは違っているという事をまず申し上げておきたいと思います。例えばオブジェクト指向にしても UML にしても、クラスという概念が必要です。そういう意味では集合論という概念を用いています。なぜ完全な公理論ができないかというのは、クラスと同じ概念の問題が作用していて、公理系を作るのが難しいことははっきりしています。論理学でも手におえない概念をソフトウェアの世界に持ち込むという事は、とてもおかしいと思います。オブジェクト指向のパラダイムにおけるクラスという抽象概念は、ソフトウェア技術の実現を非決定論的に行うものだと思います。こういう理由から、私はオブジェクト指向の技術に賛同しきれないのですね。

ブックマン教授 クラスという概念を用いる事が問題だとしている理由が理解できません。

根来社長 クラスというのは正に心の問題で、クラスを定義する事はできないからです。クラスの問題は人間1人1人皆違うものです。こうした違いは、クラスの実現がケース・バイ・ケースとなることに現れており、ソフトウェア開発が非決定論的となり、個別性が出てきてしまいます。ですから、私は事例によるクラスの特定制が異なるという事から、ソフトウェアが異なると言っているのです。

ブックマン教授 先程インテンションについてお話されましたが、インテンションは人によって違うのですから、クラスが人によって、また特定化後の

