

# 「モデル」はすべてのソフトウェア開発の基盤となりうるか

ソフトウェア・シンポジウム2009  
モデリング・ワークショップ  
ポジション・ペーパー

(有)メタボリックス 山田正樹  
masaki@metabolics.co.jp  
<http://www.metabolics.co.jp/>

## 1. 自己紹介

SRA, ソニーコンピュータサイエンス研究所を経て, 1995年にメタボリックスを設立. オブジェクト, モデリング, ソフトウェア・プロセスなどを技術をベースに, コンサルティング, プロジェクト支援, 教育, ソフトウェア開発, 執筆などを行っている.

ソフトウェア・シンポジウムでは, 2007新潟, 2008高松とモデリング・ワークショップを開催している.

自分と「モデル」との個人的な関わりは,

- \* 1960~70年代 - 挫折した模型趣味の子供
- \* 1980年代 - 世界模型を夢見るぼんやりした若者
- \* 1990年代 - オブジェクト指向なソフトウェアのモデリングで生計を立てる
- \* 2000年代 - 見るモデルから動くモデルへ, モデルから知識へ, と妄想に歯止めが掛からない

最近の「押し」はGrails. 単なるWebアプリケーション・フレームワークに見えるけど, 実はモデル駆動/ドメイン駆動開発のための, よくできた現実的な環境だから.

## 2. 立場

今回は探究者の立場からワークショップに参加します。

私の探究者としての「問い」は「モデルはすべてのソフトウェア開発の基盤となりうるか」です。

ここではあえて「モデルとは何か」は定義しません。モデルは多分みんながモデルと思っているもののぼんやりとした総体です。もちろんUMLとかだけには限定しません。

ソフトウェアには、さまざまな「知識」が詰め込まれているはずですが。例えば業務系ならば業務の知識、業界の知識、業務が扱う商品の知識、法に関する知識などなど。もちろん実現のための知識（レジスタ機械、有限状態機械、ネットワーク・プロトコルなどなど）も。しかもそれらの知識は「形式的に書かれており」「実際に動いている」という意味で、非常に大きな価値を持っているはずですが。でも残念ながら現状では、その知識は目に見えず、取り出すこともできず、伝達も検証も変更も再利用も進化（成長）も難しい有様です。

通常の手続き型のプログラミング言語はレジスタ機械を実現系として扱うのには適しているかもしれませんが、より高いレベルの知識はプログラム・コードの中に溶けてしまいます。自然言語は高いレベルの知識を扱うのには適していますが、自然言語で表した要求や設計などと最終的に動いているソフトウェアとのリンクは最初から存在しないか、いつの間にか切れてしまいます。

ソフトウェアを実行可能な知識とすることは、ユーザにとっても、我々開発者にとっても、社会にとっても大きな利益があると思います。文字や印刷術が「見ることのできる」「記録可能な」「時間を経て伝達可能な」「再現可能な」知識となることによって、世界を大きく変えたように、です（もちろん悪い面もあったと思います。文字は人間を馬鹿に、愚劣にした点は否めませんが、それは置いておくことにします）。

では、どうすればソフトウェアを実行可能な知識とすることができるのでしょうか。「モデル」はソフトウェアを実行可能な知識とするために必要不可欠な技術ではないかと、ここ十数年考えています。それが本当か、それを実現するためには

何が足りないのか、どうすればいいのかを探求したいと思います。具体的には

\* ソフトウェアを開発するプロセスのすべてを、モデルを中心としたプロセスにできるか？

大槻さん/飯泉さんの問題フレームの考え方はとても抽象度が高く、(実現ではなく) 問題寄りのモデリング手法です。桶谷さん/尾形さんもユーザからのモデリング視点を提示していただけるかも知れません。その一方で、多分太田さん、夏目さん、時本さんは実現よりのモデリング手法を提示していただけるでしょう。これらを組み合わせることにより、すべてのプロセスをカバーできるのか？ プロセスや手法の間はうまくつながるのか？ どうやればうまくつながるのか？ ということを知りたいと思っています。

\* モデルではどこまで何を表現できるか？ どうやって解釈するのか？

問題フレームの考え方は特に形式的な記述はありませんが、そのパターンには形式性があります。問題フレームは実はいったい何を表しているのでしょうか？ それをどう解釈すればいいのでしょうか？

夏目さんの手法はシナリオというある意味でとても人間的な「物語」をベースにした表現方法があります。しかもそのシナリオを実現系にマッピングする仕組みも提供してくれます。そこではシナリオは何を表現しているのでしょうか？

上はあくまでも自分が知っている範囲での例ですが、ソフトウェアが実行可能な知識であるためには、モデルはどの範囲の知識をどのような形で表現できなければならないのかを知りたいと思っています。

\* ソフトウェアが実行可能な知識であることは本当に幸せなことか？

これはまったく具体的な話ではありませんね。でも、皆さんとの議論の中でそのヒントも得ることができれば、と思います。

それから、ソフトウェアを実行可能な知識とするための活動の一環として、アジャイル・プロセス協議会内に「知働化研究会」を立ち上げました。関心のある方の参加をお待ちします。メールを下さい。アジャイル・プロセス協議会会員で

なくてももちろん参加できます.

どうぞよろしく願いいたします.