

札幌モデリング WG の風景



北海道札幌のソフトウェアシンポジウム2009のモデリングWG、無事終了しました。
6月18日まるまる1日＋宴会付き（サッポロビール園）、19日半日の長時間会合でした。進め方は山田さんの司会進行でまず探求者から各自が抱えている課題とか開発したいものを述べ、その後、宣教者から各自の推奨手法について披露してもらうというもの。濱さんが19日は参加できないとのことで、仲間内（つまり私とか（汗））の話は後にしてよということで、問題フレームの話は後になりました。

エンジニアの開発技術関連の会合に出席するのはひさびさなのですが、それなりに得るものはあったように思います。私はメモは採らない主義なので、あまり個別の詳細内容は記憶に残っていませんが、思い返してみても、独断でまとめました。（各自のポジションペーパーも参考にしています。）

■山田さんのイントロ

モデリングとは何かというのはさておき、大局として要求（問題世界？）と、ソフトウェアとの間のギャップを埋めるということが課題という話がありました。ずいぶんと手短かでしたが、まっ、いいか（汗）。

ということで、まずは探求者の一連の発表がありました。

■私（大槻）の予測マシン（見積り手法支援システム）の話

これは、いつもの話？なので、省略。

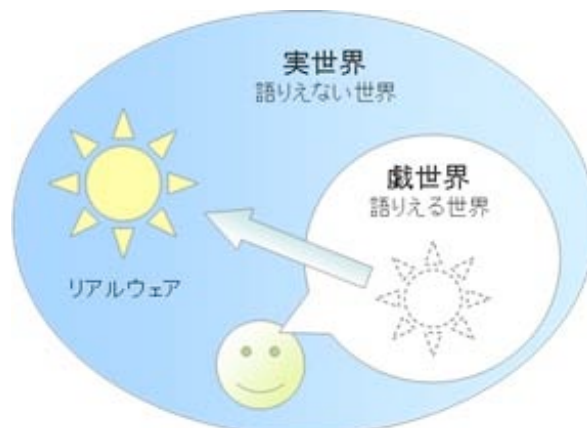
「ソフトウェア経済学」とか、「価値と解のマトリクス」の話とか、キレイな絵満載で紹介。

（ブログ「問題は「問題」にある＋予測マシン」（2009年6月18日木曜日）参照）

■濱さんの「リアルウェア」の話

濱さんが「リアルウェア」のアイデアを閃いたのは、昨年(2008年)の夏のことと記憶しています。仏教の勉強や禅の修行をしながら、ソフトウェア企業の実践者としての直感から導き出した考え方です。

「リアルウェアとは、実世界にある戯世界（人間の私的世界）の写像である」つまり、絵で描くと以下のような感じだとのこと。



実世界から戯世界への写像ではない！ところがミソです。神が作った客観世界というのがあって、それを下々（しもじも）の人間が認識していくというデカルト主義的な世界観からの脱却です。佐々木閑『犀の角たち』の書籍で言う「人間化」です。こういったパラダイムシフトは、ヴィトゲンシュタイン（Ludwig J. Wittgenstein）の前

期哲学の『論考』（論理的・哲学的論考, 1921）から、後期哲学の『探求』（哲学的探求, 1949）へのシフトと同じです。『探求』では、言語によるコミュニケーションで「言語ゲーム」という考え方が提唱されていて、「家族的類似性」という社会的規範による共通理解が言語の使用と理解の基本であると述べています。

元来、ソフトウェアに関わる活動というのは、言語による記述（分析モデル）活動ですから、上記のような世界観は必然的な流れだと思っています。

濱さんの話は、哲学的な話からの前振りで、あとは、アジャイル・ソフトウェアセル生産やメタコンポーネントの話がありました。もちろん、メタコンポーネント（プロダクト、機能、様相？、プロセスのモジュール化）はリアルウェアです♪。

ソフトウェアの世界における開発者の呪縛（例えば「人月の神話」ね）を解き放つことができるインパクトのあるアプローチと言えるでしょう。

■本橋さんの「ゆるい」のお話

本橋さんが「ゆるい」ソフトウェアを思いついたのは、2008年12月28日祐天寺の喫茶店です。証人は濱さんと私（笑）。今回のWGでも、あれやこれやと「ゆるい」について考えていることを披露しました。

大局としては、仕様をあらかじめ決めて、超計画駆動で進めても、ビジネスも変わり、人々の認識も変わり、技術も進歩して、どんどん変化していってしまうので「ゆるく」行きましょうということです。ログ解析、リコメンデーション等、それなりにゆるくなくてはやっていけない領域も出現しているのも確かです。

バリー・ベーム（Barry Boehm）が『アジャイルと規律』で計画駆動とアジャイルプロセスとのバランス（スイートスポット）があるよという話をしていますが、この話をもう少しおし進めて、進化とかサイバネティクスといったものも視野に入れていこうというものです。たぶん。

「ゆるい」は多義です。以前、本橋さんとメールでやりとりした時の私の整理では、以下のようなものがあります。（ゆるいの研究は厳密に進めるとよいな♪）

ゆるい1：抽象的

これは、問題を解くために、不要な情報を捨象することです。
抽象的であっても、厳密性は保持することができます。
用例：仕様がゆるいと実現の選択肢が多くってたいへんだね。

ゆるい2：不確実

確率事象として捉えるということです。非決定的な定式化や計算をしなくてはならないということです。
用例：ゆるい要求にはアジャイルな対応しかないでしょ。

ゆるい3：曖昧（未認識・未定義）

概念や言葉が明確に定義されていなかったり、認識が完全でない状態のことをいいます。
用例：「抽象的」という言葉のゆるさが多くの誤解を生んでいるんだね。

ゆるい4：相対的（言語ゲーム的）

概念や言葉が用法によって定義される。社会的コード（規範）の中で日常言語を定義していくこと。
用例：ゆるいってしまりがいいことってとりあえず定義しとこ。

ゆるい5：進化的

（抽象）機械がメタな機構を具備していて、状況対応、環境適合をして変貌していくこと。
用例：ゆるいですが、だんだんしっくりくるようになりますよ。

WGでは、「阿吽」の話（阿＝世界の始まり、吽＝終わり）とかのつかみもあって面白かったですね。



法隆寺中門金剛力士
（阿形）



法隆寺中門金剛力士
（吽形）

■尾形さんの「ユーザエクスペリエンス」の演習

「現状の人の振る舞いがどうなっているか」ということを、居酒屋のサービスを例にミニ・ワークショップを行い、製品企画の体験をさせてもらいました。具体的な利用場面（人物、状況）を想定して、シナリオを記述することによって、UIのインタフェースをモデリングしていくという手法です。（アジャイルプロセス協議会／見積・契約WGの地方合宿でやっている「掃除機のフィーチャ抽出」の雰囲気に近いですね。こういった体験型の企画は、啓蒙活動ではとても有効です。参考になりました♪）

演習例題が居酒屋ということもあって、「サービス」（SaaS？）のデザインには使えそうに思えました。製品企画の本質とは、「人の振る舞いを、有効で、効率的で、満足できる状態に変えるための仕組み」であるという主張は、それなりの視点を与えています。

UIに現れない世界の分析と合わせて活用していくとよいと思います。ビジネス戦略、競合他社との比較分析、ブランディング、シリーズ設計、ライフサイクル、価格、顧客心理や満足度、ビジネスモデル（いつどのようにお金をとるか、契約の方法等）、もろもろの本質的困難（規模、同調、変化、不可視性）への対処等の観点をどのように組み入れて統合化していくかも課題です。

サービス（サイエンス）に限って言えば、サービスの本質である、同時性、無形性、異質性、消滅性に照らし合わせてみるに、利用者／提供者との同時性が、ここで言う「インタフェース」のモデリングに関係しています。ジャクソン流に言えば、利用者と提供者との間の共有現象を、ダイナミックなシナリオを記述していくことにより、実世界（ユーザ）と製品（サービス提供）との接続方法がデザインできるということになります。

--

以下は宣教者という位置づけの発表です。

■桶谷さんの「ユースケースモデリング」のお話

小樽商科大学ビジネス創造センター所属のユーザエクスペリエンス部門の研究員をされているとのことで、今までの実務経験の基づき、ソフトウェア開発プロセス、特に「要求定義」を研究されているそうです。

要求定義は対象の事象（人も含んで）から情報を引き出して整理し、システムが実現できることに変換していくプロセスであるという立場で、「ユースケース導入事例ガイド」（Daryl Kulak and Eamonn Guiney）のユースケース駆動モデリングの紹介をしていただきました。（ユーザは要求を知っているかどうかは別として・・・（汗））

実践的な立場でのユースケース活用法です。目的指向（ゴール）の考え方を導入しており、要求の場面抽出にも有効かもしれません。ユースケースのようなダイナミック（時間に沿った）記述は、どこかでやらなくてはならなりません。シナリオ型のゆるい記述がよいのか、CSP(Communicating Sequential Process)のようなもう少し厳密なプロセスモデルがよいのかは議論が分かれるところだと思います。

また、「要求」を記述するのか、要求の前提となっている対象領域の定義や認識（あるがままに法則や原理を分析）を記述するのかといった区別も必要に思えます。実装（実現）との独立性の問題もでてくるでしょう。大きなシステムの要求を、サブシステムに分解して分析していくといったとたんに、実装概念が入ってしまうところも心配なところです。

■太田さんの「MSモデル駆動開発」のお話

以前は富士ゼロックスにおられた方で、2006年からマイクロソフトで、エンベデッドエバンジェリストを担当。マイクロソフトの組込み系製品/技術の普及啓発に携わり今に至るとのこと。エバンジェリストさんですから、つつい宣伝が入ってしまうのをぐっと堪えて、「より早く、より簡単に、望みのシステムを作り上げる方法はあるか」のお話。

まず、モデルとは「アイデアやコンセプトを可視化し、一定の形式に従って、形式的に表現したもの」と、WGへの気配りが前振りです。まあモデルとは「記述」ということね♪。ソフトウェアは目的別に、知識体系、実現環境、製品開発に分けられるという分類をしていて、MSは「実現環境」を提供することをミッションとしているとのこと。（まあその裏にはプラットフォームベンダとして収益を上げることが本音というのは飲み込むことにしましょう（汗））

これに引き続き、MDA(Model Driven Approach)、プラットフォーム独立、同依存部分の話とか、Visual Studio（ビューポイントの話とか）、ソフトウェアファクトリー（ズ）、DSL(Domain Specification Language)とか話題満載でした。次世代のモデル

駆動型開発環境として現在開発を進めているOsloの話もあったような、なかったような・・・（汗）。

例題としてETロボコンのレースのモデリングの話がありました。プラレールみたいにコースを直線部分、カーブ部分といった形にモジュール化して、それぞれの詳細化をして実装につなげるというもの。

会合時だったか忘れましたが、山田さんが、よくもまあMSさんは次から次へと新しいもの出して来て、ユーザ（開発ベンダも）は困らないかなと鋭いつっこみがありました。ソフトウェアのライフサイクルと開発環境のライフサイクル（ついでに、企業のライフサイクル）とかの関係は、ちゃんと整理しておかないと火傷しそうな領域ではあります。

■時本さんの「TM（T字形ER手法）」のお話

佐藤正美氏が考案したTMの紹介をしていただきました。私も、かつて（四半世紀くらい前（汗））、設計支援システムなるものをERモデルベースで開発していたこともあって、お話の内容も共感できるものでした。H社系は、データ中心設計（堀内一氏）とか、TH(椿・穂高)モデルとかの派閥？だったので、TMの話をちゃんと聞いたのは初めてです。

モデルはコミュニケーションのためのツール、または表現方法の一つという考え方を採っています。つまり、モデル（記述）は、単純に自分勝手に描くというより、表現されたものが正しいかどうか、書かれた論理を正確に読み取らなければならないとのこと。

コンセプトとしては、仕様（インタフェース）から実装までをハイブリッドな一つの言語の中で一貫して扱ってしまおうというもの。ERモデルは静的なモデルですから、実行やプログラムまで包括的に扱うためには、何らかの動的（ダイナミック）なモデルを導入する必要があります。一つの解決方法は、独立したダイナミックなモデル（状態遷移、並行プロセス、ペトリネット、プログラム制御等）を導入する方法ですが、TMはあえてハイブリッドな一つの言語にこだわっています。

モデルとコードの一貫性を保つことが、実践的な観点では重要ですし、多くのシステム開発は、既存のコード（母体）がある場合がほとんどですから、ハイブリッド方式の方がよいかもしれません。

無論、ERモデルを中心に扱える範囲というのは、適用領域としてはかなり限定したものになるでしょうし、新たに適用領域の記述体系を開発しなくてはならない場合にはTMは、ハイブリッドであるが故に、かなりの難局に直面すると思われます。そもそも問題領域（実世界）の構造と、計算機の世界とは縁もゆかりもないはずですから、その意味では、適用範囲は限定されます。一方で、中小規模の情報システム（データベースによる蓄積と問合せ）の場合には、素朴なユーザ層でも対応が可能なものにすることができるかもしれません。

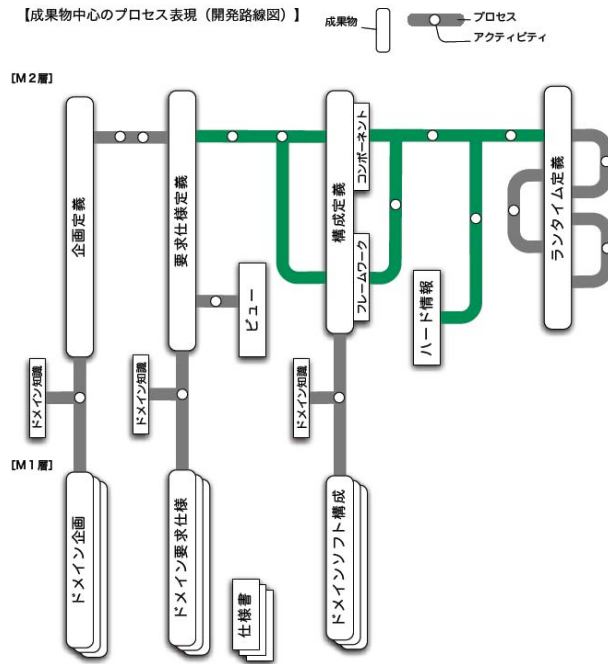
蛇足ですが、データベースの世界で概念、論理、物理モデルという言い方をしますが、これが最大の失敗と言われています（これはジャクソン先生の書籍に書かれていました）。これ等を実世界、インタフェース（共有現象）、実装モデルという言い方をしていれば、もう少々我々にも馴染み深いものになったと思います。

■夏目さんの「メタモデルによる成果物定義と開発プロセス」のお話

デンソーで、走行系システムを中心に組込みソフトウェア開発に従事してこられた方です。夏目さんの理想と、会社の現実との間で悩みつつ、開発プロセスや支援環境について、志高く、いろいろ試みているようです。

今回は、開発プロセスそのもののモデルについてお話がありました。V字モデル（SLCPとかも同じ）を規定しただけでは、組織や部署の派生が生まれてしまい混乱してしまうという問題があります。そこで、中間成果物を駅に見立てた路線図で表現するアイディアを提唱しています。この考え方は秀逸。企業間（例えばSI企業とベンダ）の調整に適用できると思いました。

メタモデル（M2層）で開発プロセスを定義し、これにドメイン知識を加えてインスタンス化することで、ドメインモデル（M1層）を導出すれば、どのドメインであってもM2層で定義した手順が共通に利用できるという所が狙いとのこと。



夏目さんの問題意識は、上記のようなゆるいプロセスモデルを定義しただけではだめで、無秩序な努力を重ねるのではなく、部分を担当する人ががんばっても、全体としてバランスが保てるようにする組織デザインが必要とのことです。TOC(Theory of Constraint)とか何らかの全体と部分との関係を分析する手法が有効かもしれません。

後半の話は、仕様化（要求定義）から実装、ランタイム環境までの支援環境です。ご本人はいやがっているようですが、とにかくキレイです。FlashとFrameMakerとを使って個人的にコツコツ、かつ、センス良く作り込んでこられたようです。ベンディングマシンのシミュレーションデモを見せていただきました。

UMLを使わないとか、関数型ではないとか、情報隠ぺいしないとか、いわゆる常識を破りつつ、実践的にバランスのよいものを作りあげています。仕様から実装に至るトレーサビリティに随所で工夫（マトリクス記述）があるのがスゴいなという印象を持ちました。

■飯泉さんの「問題は「問題」にある」のお話

最後は、時間もぎりぎりになってしまったのですが、飯泉さんの問題フレームの紹介です。本来の予定では、話の後半で予測マシンの問題フレーム適用例を解説する予定でしたが・・・時間切れ。まっ、いいか。

（ブログ「問題は「問題」にある＋予測マシン」（2009年6月18日木曜日）参照）

■総括

1 日半みっちりとお話の話をきかせていただいて、新しい時代の方向性についても確認することができました。各自のメッセージ性を私の偏見で列挙すると以下の通りです。

・山田さん：	機能	から	様相	へ
・大槻：	人月	から	価値	へ
・濱さん：	ソフトウェア	から	リアルウェア	へ
・本橋さん：	かたい	から	ゆるい	へ
・尾形さん：	機能	から	ユーザエクスペリエンス	へ
・桶谷さん：	実装	から	要求定義	へ
・太田さん：	プログラミング言語	から	ドメイン記述言語	へ
・時本さん：	スペシャルモデル	から	ハイブリッドモデル	へ
・夏目さん：	V字プロセスモデル	から	メタプロセスモデル	へ
・飯泉さん：	要求	から	問題	へ

新しい方向性を強引に総括するとしたら、《知働化》でしょうかね。人働説から知働説へのパラダイムシフトといったところでしょう。社会や経済におけるソフトウェアの役割、また、ソフトウェア開発の社会的役割分担、ビジネスモデルも時代とともに変わってきています。サービス指向、クラウドコンピューティングといったキーワードも世間を騒がせていますが、時代の潮流を見極め、本質的なアプローチをしていく必要があると思います。

■謝辞

札幌のシンポジウムは、モデリングWGのみの参加でしたが、私としても学ぶところも多く、大変楽しめました。ここにご参加、ご討論いただいた方々に感謝いたします。そして、山田さん、お疲れさまでした。

今後の皆様の、ますますのご発展をお祈りしております。

