



Interactive Essay . . .



XMLという道具

佐藤 洋介・高野 浩章／風林火山(WFFM)

本稿では、我々がゲームへXMLを使用したこれまでの成果や経験を踏まえ、XMLを取り巻く現状や、システムでXMLを使用する意義について述べる。

はじめに

昨今、世間一般の間ではXMLの熱が冷めつつあるらしい。これを裏付けるデータとして、XML書籍の売上不振が挙げられる。原因はいろいろ考えられるが、1つの理由はIT不況により新技術へ投資する企業が減ったことであろう。もう1つ考えられる理由としては、実際のプロジェクトへXMLを適用したが、期待したほどの効果がなく失敗してしまっただけの事例が増えてきてしまい、XMLの評判が下がりつつあることである。

XMLを前面に押し出し、失敗したプロジェクトでは、「XML」を使う根拠／理由が薄弱なケースが多い。オープンなデータフォーマットを積極的に使おう、という政治的な判断は否定しないが、「XML」を使う根拠／理由をプロジェクトへ適用する前に一考するべきである。

風林火山とXML－月山の場合

Java研究サイトである我々「風林火山(WFFM)」¹⁾では、XMLとJavaを活用した次世代ゲーム開発を行っている。これまでの成果の中で代表的なものが『月山実行環境』(以下月山)である。これはXMLで記述されたシナリオを実行する、デジタルノベルタイプのJavaゲームアプリケーションである。

他のノベルゲームアプリケーションで多く採用されているBASIC風のスクリプト言語は、シナリオが大規模になるとスパゲティコードと呼ばれるパターンが現れる。これは、BASICのGOTOステートメントのように自由自在にコード中をジャンプすることにより、少々規模が大きくなるだけですぐにシナリオを追いかけることが困難になり、メンテナンス不可能になってしまう状態である。月山では、シナリオを構造化することにより以上の問題の解決を目指し、構造化シナリオを最も楽に扱えることから、XMLを「構造化されたドキュメントを記述法するための一技法」として捉え、使用した。XMLドキュメントの構造はDOMツリーとして取り扱えることから分かるように、ツリー構造を表記するには「うってつけ」であった。

結果的に、ゲームシステムからシナリオが完全に切り離

され、他システムへの展開が容易となり、複数のソフトウェアでの再生を実現している。システムアーキテクチャは、XMLにより構造化されたシナリオをModelとして捉えた、MVC(Model-View-Controller)フレームワークによって構成されている。これにより、ModelとViewの分離がなされ、高いViewの柔軟性を確保した²⁾。

風林火山とXML－今後のXMLとの付き合い方

我々は現在、ノベルゲームシステムに特化した月山に対し、より抽象度を引き上げた、ゲームシステムが作れないか模索中である。ゲームを解くときの基本的なアクティビティ(作業)に着目し、法則・現象を記述し、それらの順序関係、状態変化の時系列を記述することによって、五目並べのような簡単なゲームを作ることができるゲームシステムを考えている。このシステムもご多分に漏れずXMLで開発する予定である。このシステムがXMLを使用する理由の1つは、月山の経験則からXMLをデータフォーマットとして使用した場合、他のオープンソースコミュニティがシステムを拡張しやすいことである。また、XMLはプレーンテキストデータであるためCVSでのリビジョン管理も可能であり、他のオープンソースコミュニティのシステムやライブラリとの連携が容易でもある。

以上より、我々は今後ともしばらくの間、リビジョン管理や連携性の面でXMLを使用していく予定である。

XMLは道具の1つ

これらが、我々がXMLを活用している理由である。ITのキーワードの1つとしてXMLが上げられるなか、SVGやSOAP等、次々とXMLは活躍の場を広げている。そのため、従来のドキュメントに取って代わる万能なもののように謳われている。しかし、冷静な目で見れば、どれも過去の実績や経験を積み重ねた結果、これはXMLにすべきだと厳選されたものと分かる。流行に流されたり、面白半分や勢いに任せてXMLにしたものではないのである。最近では、雑誌や書籍等の煽りによって、あまりにも急速にXML熱が高まってしまった。そのお陰で、なんでもかんでもXMLを適用しようとして失敗、そして「XMLは使えない」となって



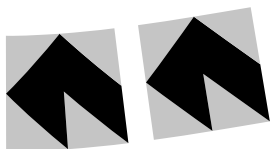
しまい、業界全体にXMLバブル崩壊的な臭いが広がりつつあるように思える（同じような現象として、最近のデザインパターンブームも挙げられる）。先ほども申し上げたとおり、我々風林火山にとって、XMLは単なる構造化されたドキュメントであって、それ以上でもそれ以下でもない。XMLに過度な期待をしないで、もっと肩の力を抜いて、問

題解決の道具の1つとして使用すればよいのではないだろうか？

参考文献

- 1) 「風林火山 (WFFM)」 <http://www.wffm.org/>
- 2) 佐藤洋介: オブジェクト指向開発プロセス, XML Press Vol.1, 技術評論社 (2000).

(2001.11.21)



XMLという道具の有用性

横山 至治 / (株)デュオシステムズR&Dセンター

まず、エッセイの主旨である「XMLを使う根拠／理由をプロジェクトへ適用する前に一考すべき」という指摘には全面的に賛成である。「XMLは汎用性が高い」ということを大げさに「XMLは万能」のように書き立てる風潮はまあ見受けられるが、それを開発者が真に受けてしまうのはいかがなものだろうか。XMLの仕様を見れば、XMLにも他のすべてのものと同じように得手不得手があって、その長ずるところをうまく活用しないとかえって非効率になってしまうのは明らかだと思うのだが...

さて、XMLを道具としてうまく活用した例として挙げられた風林火山の月山プロジェクトは非常に興味深い。シナリオの構造化を実現するためにXMLを使用し、さらにXMLと親和性が高く、プラットフォームに依存しないJavaを実行環境として採用したのはまさに適材適所である。また、優れたゲームのシナリオライターが必ずしもXMLを書けるとは限らない（恐らく書ける人の占める割合はごく僅かであろう）という観点から、直感的にシナリオファイルを作成できる開発環境「敵島」も並行して開発されているのはユーザーに優しいといえる。

ただ、ひとつ気になるのは上で書いた「XML + Javaの組合せは適材適所である」ことが、実は現状ではほとんど必然なのではないかということである。つまり、他にノベルゲーム開発環境を作ろうという人がいたら、「一考」の結果同じようにXML + Javaの組合せを採用することにしたとしても何も不思議はない。その人が月山の存在を知らず、独立してプロジェクトを発展させていったとすると、XML + Javaという組合せは同じでも、スクリプト言語の規格に互換性がないことになってしまう。もしそのようなことになるとお互いのユーザーにとって不幸なことだし、競争の結果どちらかの規格で統一されることになると、片方の規格で書かれたシナリオ資産が無駄になってしまう。

このようなことを避けるために、ノベルゲーム開発者のコミュニティ（XML開発者のコミュニティではなく）の中で月山プロジェクトを積極的に周知する必要があると考え

るが、そのような取り組みをされているかお伺いしたい。このような周知活動のなかで、もし似たようなプロジェクトを進めている人たちが見つければ、それぞれのスクリプト言語に互換性を持たせたり、お互いの長所を取り入れ合ったりすることができて、XMLをシナリオ記述のための道具として使用する場合の有用性をさらに高めることができると思うが、いかがであろうか。

最後に、今後の活動計画として「より抽象度を引き上げたゲームシステムの構築」が挙げられているが、これも興味深い。正直なところ、ノベルゲームはゲームという広大な分野のほんの一角に過ぎないので、より広い対象に対応することは好ましいと考える。ただ、エッセイの記述では具体的にはどのようなゲームが対象になっているのか判然としない。推測するにノベルゲームのシナリオではなくゲームのシステムそのものを記述する言語および開発環境を開発することを目的とし、主な（当面の）対象としては五目並べのようなパズル性の高いゲームになるのだろうか。このあたりについてより詳しい解説をいただければ幸いです。

いずれにしろ、対象とするゲームの範囲を広げれば、上で述べたような開発者のコミュニティも広がるのが容易に想像できる。ゲームシステムの構築と並行して、開発者のコミュニティとの連携を密にし、汎用性や互換性の高いゲーム記述言語を構築できればXMLという道具の有用性を最大限に活用できたといえるのではないだろうか。

(2001.11.24)



風林火山に期待する

浅海 智晴☆

XML&Javaの世界において風林火山の活躍はつとに有名である。

ホームページ上の美少女キャラを見て引いてしまわれた方も多と思われるが、この美少女キャラを仮想空間におけるアバタ、美少女キャラの登場するノベルゲームをMVCフレームワークに基づく対話型eラーニングシステムと考えると、その本質が見えてくる。

風林火山の活動は最先端のXML利用技術なのである。ホームページのファーストインプレッションによる誤解で、XML業界のバブルがさらに弾けるとまずいので補足しておきます。

さて、本エッセイによる風林火山の主張は、簡単にいうと「XMLは使えないっていう雑音も大きくなっているけど、便利なんだから使えるところで気楽に使ってみたら」ということだと思われる。ただし「はじめに」にある『XMLを使う根拠/理由をプロジェクトへ適用する前に一考すべきである』という文言によって「使えるかどうかはちゃんと一考してから決めてね」という予防線も張ってある。

困るのである。もっととんがったことを言ってくれないと。適当にポケてくれないと、ツッコミようがないのである。とても優等生的主張であるだけに、コミュニケーションのポイントがとりづらい。

一点、かろうじて引っかかるところを見つけるとすると、XMLバブルが弾けた理由であろうか。

風林火山の主張によれば、XMLバブルが崩壊した原因は、「実際のプロジェクトへXMLを適用したが、期待したほどの効果がなく失敗してしまった」となっているが、ボクはそう思わない。XMLの有効性そのものの主張はたくさんあったが、結局「何に使うと便利なのか分からない」ということではないかと思う。導入して痛い目にあったというより、そもそも導入の仕方が分からなかった、あるいは導入するメリットが分からなかった事例の方が圧倒的に多かったのではないかと思うのである。

新しい技術が登場するとき、必ず2つの方向からのアプローチが行われる。

1つはビジネスピープル主導による、演繹的、トップダウン的なアプローチである。このアプローチは技術分野にお金を持ってくる効果はあるが、これだけではイノベーションは起こせない。トップダウン的なアプローチは、結論

ありきの方法論であり、先進企業による成果をキャッチアップするには有効であるが、それ以上の成果は出せない。仕様書や論文をそのまま翻案しただけの技術解説もこのカテゴリーに入るだろう。

もう1つはエンジニア主導の、帰納的、ボトムアップ的なアプローチである。新しい技術を自らの手で具体的に実践しながら、実用という観点からその有効性を検証していく。

このアプローチはとても地味であるが、新しい技術が実用性というとても高度な性質を獲得するためには、必要不可欠な作業である。

そして、後者のアプローチを、実践している集団が、ほかならぬ風林火山だと思うのだ。

しかし、実践主義だけでは、限界がある。個々人の自由な発想による実践の成果を積み上げていくことは重要であるが、その次の段階において、それらの成果に対して、帰納的な推論を行い、その結果抽出された技術体系に対して演繹的な検証を施していく、という作業が必要となるであろう。

あるいは、SECI (Socialization, Externalization, Combination, Internalization) モデルによる知識創造プロセス¹⁾に沿っていえば、実践による暗黙知から、いかにして形式知を取り出し、形式知として加工した上で、さらなる知識創造の核となる暗黙知に内面化させていくのか。そして、この一連の流れを循環させることで、ポジティブフィードバックを現出させることができるのか、という点が問われているともいえるだろう。

このためには、このような知識創造プロセスを循環させるための「対話の場」の確立が重要となるが、それにもまして、対話の核となるべき実践による暗黙知が不可欠である。

この核ともなるべき暗黙知の蓄積を持つ実践者としての風林火山。このパワフルな集団が今後どのような活動を行っていくのか、ボクはとても注目している。

仮にXMLバブルが弾けたのだとすれば、これは実践の伴わない、翻案指向の脆弱な技術基盤がその真の原因であろう。「何に使うとどう便利なのか」というノウハウは実践の中からしか生まれてこないのである。

『XMLは単なる構造化されたドキュメントであって、それ以上でもそれ以下でもない』と醒めたことは言わないで、『問題解決の道具の1つ』として風林火山がどのようにXML

☆ <http://www.relaxer.org>

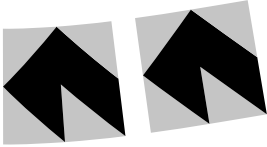


を使いこなしてきたのかという、実践による暗黙知の蓄積をXMLコミュニティに還元する活動を期待するのである。

参考文献

1) 野中郁次郎: 組織的知識創造の新展開, ダイヤモンド・ハーバード・ビジネス, Vol. 24, No. 5 (Sep. 1999) .

(2001.11.24)



システマティックなコンテンツ制作と風林火山

徳原 徹 / 慶應義塾大学政策メディア研究科

私は現在東京工科大学片柳研究所クリエイティブ・ラボで次世代映像制作システム「C-PALS」の一環として「ジオラマエンジン」という3次元絵コンテシステムを開発している。同じ「コンテンツ」ドメインでシステムを開発しているものとして意見を述べたい。

風林火山プロジェクトには以前から興味があった。柔軟なアーキテクチャと拡張性の高いJAVA + XML ベースのノベルゲームシステム。その場しのぎな単発的ゲームシステムが多いなか、将来性を見据えた風林火山のコンセプトは非常に新鮮で魅力的に感じた。

ゲームシステムとして開発されたものであるが、「ゲーム」から「コンテンツ」へ少しコンテキストを広げて述べてみたい。

コンテンツ制作の問題点

日本が世界に発信できるコンテンツとしてゲームとアニメがあげられる。共にエンターテインメントとして多くの人に親しまれ、経済価値も高い。しかし、制作現場では効率的とはいえない方法でコンテンツが制作されている。

ブロードバンドインターネット時代の到来により、コンテンツの重要性が増してきた。ゲームや動画アニメはキラ

ーコンテンツになり得るものとして注目を集めているが、そのためには、クオリティの高いコンテンツを継続的に制作できるシステムが不可欠である。

システマティックなコンテンツ制作

システマティックなコンテンツ制作を実現するには多くのハードルを乗り越える必要がある。その中でも「コンピュータ」に適しているのは「素材の再利用」であろう。現状では、素材はそれを生成したシステム内ではしか利用できないことが多いからだ。

風林火山はXMLを利用することによりシステムからシナリオデータを完全に切り離れた。このコンセプトを発展させ、コンテンツ制作の場で「素材データを他システムで再利用」可能にすれば非常に有用である。

風林火山の手法はゲームシステムとしてだけでなく、コンテンツ制作システムとしてみても大きな一歩であると感じる。

参考文献

1) 金子 満: 慶應義塾大学EENプロジェクト: メディアコンテンツの制作, (財) 画像情報教育振興 (CG-ARTS) 協会, 東京 (1998) .

(2001.10.29)



XMLプロジェクトのアンチパターンを

佐藤 洋介 / 風林火山 (WFFM)

浅海さんより「文章が綺麗すぎ! からめない! ポケなさい!」とのご指摘がありました。文章では去る第5回XML開発者の日のように歌って踊れないのが難点じゃば〜。WFFMサイトのファーストインプレッションが原因でXMLにトドメを刺すことになれば、これはこれで凄いアンチパターンな気がします、名付けて風林火山パターン。

XML規格カタログ

横山さんのおっしゃる通り、同じ目的のために複数の規格が乱立することは、ユーザにとって不幸、このうえない事態である。同じような“車輪の再発明”的状況は、現在のソフトウェア再利用部品にも見られる。過去から現在まで、全世界で同じようなソフトウェア部品が何度“車輪の再発明”されていることか。一向にソフトウェア部品の再

利用が現実味を帯びない原因の1つである。XMLの場合，“車輪の再発明”防止策として規格カタログの整備が望まれる。2年前の月山開発時は、同じような目的でXMLの規格を作っている人がいるはずであると考えた。そこで、既存の規格の存在を探索しようと試みた。しかし、そもそもどこを探せばよいか分からず、仕方がないのでWWW検索エンジンを利用するにとどまった。これは2年前も今も変わらぬ現状かと思う。

現在、XML規格カタログを目指した「XMLフォーマットのイエローページ」¹⁾という動きが(社)電子情報技術産業協会(JEITA)に存在する。残念ながら、いささか情報が古いということもあり、機能しているとはいええない状況である。事実、月山のシナリオフォーマットであるGSML(Gusan Scenario Markup Language)も現行バージョンは1.1であるが、そのページに掲載されているのは1.0のままである。更新通知を送っているのに反応がない...しょんぼり。やはり、営利目的ではないサイトを継続的に維持・管理していくということは、とても根気がいる作業なのである。

ちなみに今後の活動の予定の件だが、「LEGO」²⁾のコンピュータゲーム版の開発を目指す予定である(LEGO MIND-STORMSにはなはだしく感動したため)。理想は、あらかじめ用意されている部品化されたルールを組み合わせるだけで、簡単に新しいゲームの制作ができるようになることである。創作の楽しみや、完成の喜びを感じていただけるようなシステムを目指している(まだ具体性がなくてすみません)。

アンチパターン集

浅海さんから、「風林火山はエンジニア主導の、帰納的、ボトムアップ的なアプローチである。その暗黙知の蓄積をコミュニティへ還元しないとね。」とのご指摘を受けた。おっしゃる通りであり、お世話になった分、恩返しをしな

ければならないと考えている。

さて、暗黙知から形式知を取り出す手法であるが、ソフトウェアパターンが良いお手本であろう。オブジェクト業界では、現場で培われた暗黙知である「あるコンテキストで繰り返し遭遇する課題に対する解決策」を、ソフトウェアパターンと名付けて形式化している。本当に現場からの暗黙知を吸い上げることができているかいないかは別として、PLoP(日本ではJPLoP)等に代表されるソフトウェアパターンコミュニティは、その目的を担っている団体である。XML業界もこれを見習い、プロジェクトにXMLを適用したときの成功事例、失敗事例(アンチパターン)集を容易に参照できるようになればと思う。成功事例よりも失敗事例の方が重要度は高いのであるが、日本では「クサイものには蓋をしろ」という風潮があり、失敗事例を隠したがる傾向が強い。いかに失敗事例を現場から吸い上げるかがポイントになる。失敗事例が出揃ってくれば、自ずとXMLの長所、短所が浮き彫りになる形となり、プロジェクトへの採用の基準も明確になる。したがって、XMLによるプロジェクト失敗事例も減少すると考える。

参考文献

- 1) (社)電子情報技術産業協会(JEITA): XMLフォーマットイエローページ,
<http://it.jeita.or.jp/jhistory/japanese/committee/SOL/XML/index.html>
- 2) LEGO Group: マインドストーム公式サイト,
<http://mindstorms.lego.com/japan/>

(2001.10.29)



議論の続きは、次のURLをご覧ください。 <http://www.ipsj.or.jp/magazine/interessay.html>