

コンテンツ開発技術とソフトウェア工学

井 上 克 郎†

ソフトウェア工学は、対象として、ソフトウェアやそれに付随する仕様書、設計書などのドキュメントをいかに効率よくかつ高品質で作ることを目的としてきた。一方、最近では、Web ページやそれに伴うソフトウェアの開発の需要が、急速に拡大してきており、そのような Web システムを高品質で効率よく作成する技術が、必要になってきている。

本稿では、そのようなシステムを作るための技術と、ソフトウェア工学の技術と対比しながら、構築のための方法論や技術について議論する。

Contents Development and Software Engineering

KATSURO INOUE†

Software Engineering is collection of technologies and methodologies for developing software and related products effectively and efficiently. In these days, development of contents associated with Web systems is an emerging and required issue.

In this paper, we discuss contents development technologies in comparison with software engineering technologies.

1. はじめに

ソフトウェア工学は、対象として、ソフトウェアやそれに付随する仕様書、設計書などのドキュメントをいかに効率よくかつ高品質で作ることを目的としてきた。一方、最近では、Web ページやそれに伴うソフトウェアの開発の需要が、急速に拡大してきており、そのような Web システムを高品質で効率よく作成する技術が、必要になってきている。

ソフトウェア工学は、その対象は、純粋なソフトウェアである（ここでは P と書く）。一方、Web システムでユーザーが目にする文章や図、音、アニメーションなどの個々の表現物をコンテンツ C と書く。Web システムを開発するためには、 P と C が必要である。さらに、HTML のタグなど、 C を構造化して P へつなぐための構造化要素 T 、さらに、構造化要素と親和性が高く記述能力の高いスクリプト S が用いられる。 S は P の一部とも考えられることが多いが、開発や保守には違った考え方が必要と思われるので、ここでは区別している。 C 、 T 、 S 、 P 全部を合わせてコンテンツプログラムと呼ぶことにする。ここでは、コンテンツプログラムを効率よく、高品質に開発するための

技術について考えてみる。

ソフトウェア工学では、 P の開発のため、要求獲得、仕様記述、設計、コーディング、テスト、保守など、開発を工程に分割して、それぞれをうまくやるために方法やツールが提案され開発されてきている。またこのサイクルを開発の部分ごとに繰り返し適用するスパイラルの開発も、近年のオブジェクト指向技術の発展とともに用いられるようになってきている。ここでは、この種の技術を E_P (Engineering for P) と表す。

一方、高い品質の C を効率よく開発するための方法はあまり知られていないが、読みやすい文章を書くための工夫、見やすい画面を設計するためのグラフィックデザインのガイドラインなどがそれに相当する。これらを E_C と呼ぶ。

2. コンテンツのための技術

T や S に対して、 E_T や E_S は、どのようなものであるか、ここで例をあげて考えることにしよう。

- (1) 読みやすい形で矛盾のないタグ付けができていのかどうかを、通常の文章を読んで点検する（これは E_C の一つ）のと同じように、HTML 文を読む。
- (2) 出来上がった Web ページが、使いやすいかどうかについて、 E_C と同様な手法で評価する。すなわち、ユーザーへのアンケート調査し、ユーザ

† 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University

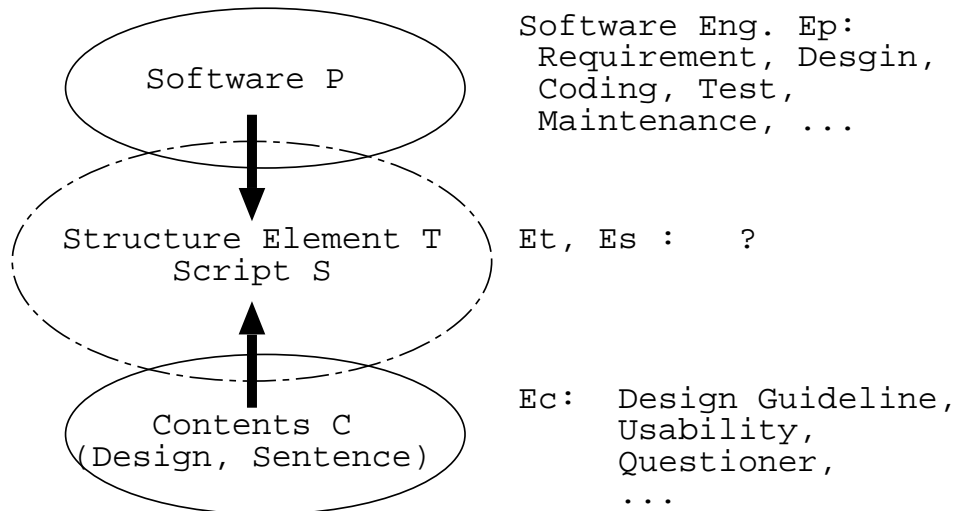


図 1 ソフトウェア工学とコンテンツ開発の関係

ビリティの実験をする(操作に要する時間、間違いの数、など)。また、実際に運用されているならば、その評判をアクセス数やリンク数で調べる。ユーザビリティでの評価とアクセス数、リンク数との相関も興味深い。

- (3) T や S に対して、 E_p で行うような、 C_0, C_1 などの網羅テストを行い、その品質を調べる。 S に関しては、 P と同様なアプローチが考えられる。 T に関しては、そのいろいろな意味を持つタグのそれぞれを網羅して試してみることが必要で、自動的にできる部分もあろうが、人手に頼る部分もある。他のシステムへの URL リンクなどは、どこまで、対象手法の範囲であるかをあらかじめきっちり決めておく、など、効率よくするための種類の工夫が必要である。
- (4) T や S に対する保守技術の一つとして、影響波及分析、結合度メトリクス、リンクの依存解析など、 E_p と同様な解析が必要である。 S は、その意味が P と同様な方法で定義されているので、解析や意味的なメトリクスを P と同様に定義することができる。しかし、通常、 S は、 S 外部とのインターフェースが重要な意味を持つことが多い。したがって、外部情報をどう定義して、内部の解析やメトリクスにつなげるかを考える必要がある。 T の解析・メトリクスは、シンタクスに関しては、容易に行うことができ、Web ページの統計的分析をするのに有効である。しかし、その意味的な解析については、定義が明確でないせいもあり、まだ、これから考

えていく必要がある。

- (5) P, S, T, C の全部をまとめて効率よく作成し、デバッグするためのプロセスや開発方法論、そしてその支援環境の技術が必須である。現在でも個別の開発システムとして商用で売られているものも多数あるが、その考え方や方法論的な裏づけが弱い。それらの補強が必須である。今までのソフトウェア開発モデルが適用できるのか、新たに作る必要があるのか、興味深い。

3. む す び

現状では、上記のような技術的な課題より、急速に拡大するニーズをどう処理するか、また、開発保守技術以外の技術的要素(例えばネットワークレスポンスをどう確保するかなど)のみが注目されてきている。しかし、今後のコンテンツプログラムの急速な発展を考えると、今まで抜けてきたこれらの技術を真剣に開発する必要がある。上述の例にもあるように、 T や S で抜けている技術に対しては、 E_c や E_p の技術をそのまま、あるいは一部変更して、適用することができるものが多数ある。今後、その対応を整備し、実際に使えるようにするための支援ツールの開発が重要である。

問題は、そのような対応が直接なく、類推がうまく働かないような場合である。特に、 C, T, S, P の連絡、連帯に関する技術、特に(5)で述べた開発方法論やプロセスに関しては、新たな努力が必要である。