

読者の理解度によって難易度が変化する電子テキストの提案

内田 智史†

神奈川大学工学部‡

1. はじめに

筆者は、Pascal、C 言語、Java、アセンブラの書籍を 10 冊程度出版してきた。書籍の価格を抑えるために、そのうちの 6 冊の書籍の版組は自分で行い、Java の書籍に関しては簡易的な版組システムを構築して作成した。

これらの書籍を執筆していて、用紙に印刷する書籍の制限に大いに悩まされた。たとえば、動的に動くプログラムの流れを、固定的な図、プログラムリストと文章で表現しなければならない。C 言語や Java の for 文の動作の順番なども文章や図で表現しなければならない。if 文や switch 文、関数の流れなども同様である。また、プログラムの流れもその動作を動的に流れるアニメーションで表示した方が読者にとっての理解度はかなり向上する。関数やメソッドを呼び出した時の引数のコピーの様子なども同様である。

しかし、残念ながら、用紙に印刷する書籍の場合、上記のような構成を取ることは不可能であった。しかし、電子テキストの場合にはこのような構成を取ることはさほど難しくない。

2. 電子テキストの利点

紙に印刷された書籍とは異なり、電子テキストには、次のような利点がある。

- (a) ページ数の制限がない
- (b) フルカラーに対応している
- (c) アニメーションや動画を導入できる
- (d) サーバとリンクさせることが可能である

サーバとリンクさせることにより、リアルタイムな読者管理が可能となる。たとえば、(i) 試験の実施、(ii) オンラインの質問受付と FAQ などの公開、(iii)、読者の閲覧状況の確認、(iv) 誤植の修正、(v) 索引の追加、(vi) リンクによる相互参照、(vii) などが挙げられる。

3. アニメーションの積極的な導入

プログラミング、データ構造、電子回路図などの授業では、実際の処理の流れやデータの変化の状態を受講者に知らせることで教育的効果を上げることは、多くの教員が理解していると考えられる。もちろん、さまざまな分野の教育で同様のことが言える。

しかし、紙をベースにしたテキストでは、アニメーションが表示されないことでデータの変化の状態を読者に知らせることができない。あえて示すことでもないが、次のようなアニメーションが電子テキストには多く取り入れられるべきと考える。

- (1) プログラムリスト上にプログラムの流れを表示する
if 文や switch 文、for 文などの動作の流れ、関数やソッドの動作の流れと引数処理、自動変数の扱い、静的変数や自動変数、オブジェクトの生成と削除の流れ等
- (2) データ構造の処理の流れを表示する
リンクリスト、ツリー、ソート、検索の流れを表示
- (3) 電子回路図の流れを表示する
AND 回路、OR 回路、NOT 回路などの流れ、フリップフロップ、記憶装置、演算回路などの実際の動作状況をアニメーションで表示

この分野に非常に興味を持ち勉強家である読者にとっては、このようなアニメーションなどは不要で、読者がきちんと理解できると主張する方もいると思われる。しかし、読者や授業の受講者全員がそのような方ではない。教育用のテキストでは正確に間違いなく処理内容を伝える必要があると考える。

ただし、for 文の動作内容を完全に理解したからと言って、複雑なフーリエ級数のプログラムを完全に書けるように指導できたとは言えない。この場合には、for 文を用いたプログラミングのテキストが必要になる。

このようなアニメーションは現在でも技術的には問題なく表示できる。しかし、現状のソフ

A proposal of electronic text which changes the degree of difficulty depending on a reader's understanding
Satoshi Uchida†
Faculty of Engineering, Kanagawa University‡

トウェアでは、これらのアニメーションを作成するのにかなりの労力を要する。そこで、簡易的にこれらのアニメーションを作成し表示できるシステムの開発が必要である。固定的なアニメーションだけでなく、読者がデータを入力するとそれに伴って、動作が変化することに対応する必要があり、この処理が難しいと考えられる。

4. ページ数とカラー化

書籍を書く際に、書籍のページ数は重要な項目である。日本の出版社の場合、長いページは避けるようである。筆者も 900 ページほど書いた原稿を 700 ページに減らされた経験を持つ。ただし、長い書籍が有効かという読者側から見るとそうでもないようで、短く簡潔なものがよいという意見もよく聞く。しかし、筆者の意見としては、説明すべきところはたとえページ数が長くなったとしても説明すべきと考えている。そこで、電子書籍の利点を活かし、短く読みたい人向け、長く読みたい人向けに内容を変化するようにできないかと考えている。

また、書籍を書く場合、多色刷りの効果は非常に大きいですが、売れる可能性の高い本以外ではカラーで出版することは印刷代が高くなりほとんど無理である。しかし、電子書籍であれば、カラー化は問題にならない。カラー化だけでなく、必要に応じては 3D 対応にすることもできる。

プログラミング言語のテキストにおいてカラーの効果は大きい。例えば、関数名、クラス、メソッド、自動変数、静的変数などを、それぞれに固定の色でプログラムリスト上に表現できれば読者の理解度が向上すると思われる。

5. サーバとの連携

多くの e-Learning システムではすでに取り入れられているが、LMS のようなシステムが書籍と連動すると教育効果が高くなると考えられる。たとえば、現時点での読者数、読者の理解度状況、リアルタイムな読者からの質問の収集、テスト等による読者の理解度状況などが把握できる。

リアルタイムな読者からの質問の収集は非常に重要である。読者は執筆者が想定もしないような考え方で、内容を理解しようとし、その結果、非常に貴重な質問をすることがある。このような質問を収集し、読者が見やすいように加工し、リアルタイムに提示する。FAQ のようなものであるが、検索システムを充実させ、大方の質問をその場で読者が読めるようにする工夫が必要となる。

また、誤植の更新もリアルタイムに行うことができるが、その内容は整理して読者が読めるようにする必要がある。

6. 読者のレベルに合わせた内容の多層化

書籍の読者のレベルは多様である。初心者もいれば、中級者や上級者もいる。当然のことであるが、同一の書籍では、これらのすべての要望を満たすことはできない。一般的な書籍では、これらの要望を満たすために、初級編、中級編、上級編などと分けることがある。当然、上級編は、内容も上級であり、上級者用に書かれているのであるが、実際には、これらの説明項目は他の編の本とは独立している。

ここで主張したいのは、たとえば、if 文の説明内容についても、初級者用、中級者用、上級者用にして、読者が自由にこれらを選択し、読めるようにしたいということである。たとえば、初級者用では、単純な if 文の説明、中級者用では if 文のブロック内部での自由変数の定義と文法の説明、上級者用では、if 文の内部ブロックに goto 文でジャンプする場合、その中で定義された自動変数は初期化されないなどの説明を行う。

現在、上記のような構成を持つ文章をどのように作成したらよいか検討している。しかし、読者にとって読みやすい書籍にするためにはかなりの日本語作成能力が要求される。文章をブロックにまとめ、文章全体にプロパティを持たせ、それを管理するワープロのようなものが必要になるかもしれない。

7. 今後の予定

「内容の多層化」以外の点については、現状の技術で対応できる。問題は「内容の多層化」を、それほど苦勞しないでどのようにして実現するかという部分である。現在、さまざまな方法で記述実験を行っている。予定としては、本稿で述べた方針に従って、C 言語や Java、オブジェクト指向の電子テキストを作り、Web 上に公開し、読者の意見を仰ぎたい。

レベルが低くても本主張の裏付けをする実コンテンツを作成し、公開し、多くの方の意見を参考にその内容を更新する必要がある。

参考文献

[1] 永井 大圭成、佐々木 太良、内田 智史、会話性を持たせた電子書籍に関する研究、経営工学会 2014 年度秋季大会予稿集、F19