6ZD-07

EPUB 出版物のための EPUBCFI に基づく索引の提案

丸山 純 † 天笠 俊之 ‡ 北川 博之 ‡

† 筑波大学情報学群情報科学類 ‡ 筑波大学システム情報系情報工学域

1 はじめに

近年，電子書籍が広く普及している．電子書籍の形式の中でも EPUB*1は，仕様が公開されており，Web 標準規格を使用していることから今後広い普及が期待されている．電子書籍の普及に伴い，ソーシャルリーディングサービス（以下，SRS）が出現している．例えば，booklook*2や liblar*3といったサービスが出現している．さらに SRS を対象とした研究も進められている [1]．SRS において，ユーザーは電子書籍リーダー上でマーカーやコメントなどの書籍への注釈や，本のレビューを行うことができる．さらに，作成された注釈やレビューは利用者間で共有可能である．

ところで EPUB では，EPUBCFI*4 と呼ばれる，書籍内の任意の点を唯一に識別できる識別子を利用することができる．これにより，注釈やブックマークなどの情報を容易に記述することが可能である．例えば，片岡らの研究 [1] では，ユーザーが付与した注釈やブックマーク，ページ送り等の情報を EPUBCFI を用いて記述している．さらに，これらを大量に収集し，ユーザの読書行動を分析している．このとき，注釈やブックマークが，書籍内において，どの位置に存在するか（先頭から何文字目にあたるか）が重要な手がかりになる．ところが，EPUBCFI は，相互の順序関係は判別可能であるものの，コンテンツにおける位置を直接導出することはできない．このため片岡らは，EPUB ファイルから別途コンテンツを抽出し，これと各 EPUBCFI を突き合わせることで，書籍内の位置を導出しているが，これには多大なコストが必要である．任意の EPUBCFI に対して，その位置情報を即座に得ることができれば，分析処理のみならず，電子書籍リーダーや SRS サーバにおいても有益である．

An index structure for EPUB publications based on EPUBCFI

Jun Maruyama†, Toshiyuki Amagasa‡and Hiroyuki Kitagawa‡

†College of Information Science, University of Tsukuba

‡Faculty of Engineering, Information and Systems, University of Tsukuba

*1 http://idpf.org/epub

*2 http://booklook.jp/

*3 http://liblar.com/

*4 http://www.idpf.org/epub/linking/cfi/epub-cfi.html

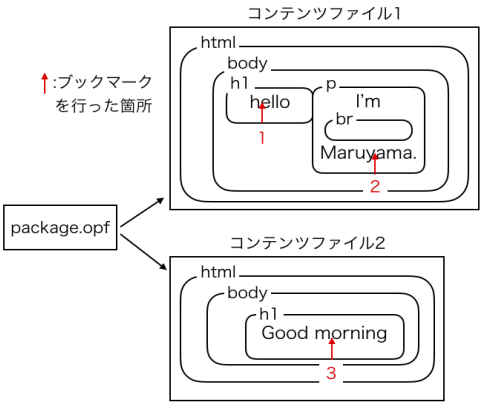

図 1 単純化した EPUB 書籍の例

そこで本研究では，与えられた EPUBCFI に対して，書籍の先頭からの位置を効率的に取得可能な索引手法を検討する．

2 基本事項

2.1 EPUB と EPUBCFI

EPUB は，IDPF が策定している電子書籍のためのフォーマットである．EPUB 出版物は，関連するファイルを ZIP で圧縮した形式になっている．説明のために単純化した例を図 1 に示す．`package.opf` には，各種メタデータに加えて，EPUB 書籍を構成するファイルと並び順を規定している．コンテンツは XHTML ファイルに記述される．

EPUBCFI は，EPUB 書籍内の任意の一点を指し示すことのできる識別子である．これはスラッシュ（/）で区切られた複数のステップから構成されるルートからのパス形式で記述される．各ステップは，XML 要素など中間ノードに対しては偶数，テキストノードなどのリーフノードに対しては奇数のインデックス値が割り当てられる．例えば，`epubcfi(/6/2!/2/2/1:2)` を例にとると，先頭の 6 はパッケージ文書内での 3 番目の要素を示しており（図には未表示），次の `2!`は，最初のコンテンツ（コンテンツファイル 1）を辿ることを示す．最後の 1 は，先頭のテキストノードに対応し，`:2` はテキストノード内の位置を示す．図 1 中には，合計三ヶ所にブックマークが付与されており，それを表にまとめたものが表 1 である．

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.

表1　ブックマークとそこから得られる EPUBCFI

ブックマーク ID	EPUBCFI
1	epubcfi(/6/2!/2/2/1:2)
2	epubcfi(/6/2!/2/4/3:5)
3	epubcfi(/6/4!/2/2/1:6)

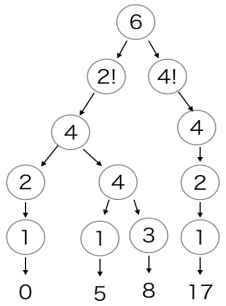

図2　提案する索引の例

3　EPUBCFI の書籍内位置導出

EPUBCFI では，コンテンツ中の任意の点を識別できるが，その位置が書籍の先頭から何文字目に位置するかは直接は分からない．このため，片岡ら [1] は，以下の手順で EPUBCFI の位置情報を導出している．

1. 分析対象の EPUBCFI をあらかじめ全て抽出しておき，文書順でソートする．
2. EPUB 書籍のコンテンツを先頭から，テキストノードの文字数をカウントしながら読み出すとともに，分析対象の EPUBCFI と照合する．

この手法では，複数の EPUBCFI を一括して処理しているだけでなく，分析の度に書籍全体をスキャンする必要が有り非効率である．このため，任意の EPUBCFI に対して，その書籍内の位置が即座に取得できると，さまざまな処理での応用が可能である．

4　提案手法

前述の目的のため，提案手法では，1) コンテンツ内の全てのテキストノードに対して EPUBCFI を導出するとともに，そのサイズを記録しておく，2) 得られた EPUBCFI とサイズの情報を元に，各テキストノードの先頭からの一度算出した上で Trie 木に格納する．

検索の際には，処理対象の EPUBCFI を元に，Trie 木を検索する．その結果，テキストノードの書籍の先頭からの位置が得られるので，EPUBCFI のオフセット情報を加えることで，書籍内での位置を即座に得ることができる．

5　評価実験

提案手法の有効性を評価するために，実験による性能比較を行った．Trie 木による索引の有効性を示すため，

表2　処理時間（ms）

	BDN	IIR
Trie 木	4.04	56.07
Trie 木 (Double Array)	3.80	52.47
PostgreSQL (B-tree)	36.90	110.10

EPUBCFI を既存の関係データベース PostgreSQL に文字列として格納した上で B 木の索引を付与し，文字検索を行うベースライン手法と，索引のサイズおよび処理時間を計測した．提案手法および比較手法対して，クエリとなる EPUBCFI をランダムに生成し，その書籍内位置を計算する処理を 10 回行い，その平均時間を比較する．実験に用いた書籍は Big Data Now[*5]（以下，BDN）と Introduction to Information Retrieval[*6]（以下，IIR）である．クスの組を取得し，PostgreSQL と Trie 木のそれぞれに格納する．Trie 木に関しては，通常の木構造を用いた場合と，Double Array による実装を用いた場合の 2 通りで実験した．また，取得された EPUBCFI は，BDN が 2,203 個，IIR が 53,392 個となった．

処理時間を表 2 に示す．処理時間は BDN，IIR 共に，Trie 木を用いた場合の方が短くなっている．また，データベースのサイズにおいては，IIR の場合，PostgreSQL の場合 6144KB に対して Trie 木および Double Array ではそれぞれ 4948KB，1328KB となった．

6　まとめ

本研究では，EPUBCFI から書籍内インデックスを効率的に取得するための手法として，Trie 木を用いた索引構造を提案した．また，実験により Trie 木による索引構造の有用性を示した．

謝辞

本研究の一部は JSPS 科研費 25330124 の助成を受けたものである．

参考文献

[1] Eri Kataoka, Toshiyuki Amagasa, Hiroyuki Kitagawa, A System for Social Reading based on EPUB3, iiWAS2013, 2013.

*5 http://shop.oreilly.com/product/0636920022640.do
*6 http://nlp.stanford.edu/IR-book/

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.