

Jak używać rzeczy niezgodnie z przeznaczeniem, czyli interaktywny PDF składany w T_EXu

Andrzej Odyniec

14 lutego 2023

Spis treści

1	Wstęp: książka interaktywna	2
2	Format PDF	2
3	Akordy nad tekstem	2
3.1	Transpozycja tonacji	3
3.2	Jak to zostało zrobione?	4
3.2.1	Warstwy OCG	4
3.2.2	Ograniczenia rozwiązania	4
4	Zakładki na życzenie	5
4.1	Rodzaje zakładek	5
4.1.1	Hiperłącza	5
4.1.2	Zakładki pliku PDF	5
4.1.3	Zakładki aplikacji	5
4.2	I to czwarte... niezgodnie z przeznaczeniem	6
4.2.1	Realizacja zakładek w formularzu	6
4.2.2	Ograniczenia i niespodzianki	6
5	Podsumowanie	7

Streszczenie

Celem tego wystąpienia jest pokazanie dwóch niezależnych rozwiązań, czyniących interaktywną publikację składaną w T_EXu do formatu PDF. Pierwsze pokazuje możliwość dynamicznej zmiany tonacji akordów złożonych nad tekstem pieśni przy pomocy mechanizmu warstw OCG. Drugie demonstruje realizację dynamicznych, interaktywnych zakładek, obsługiwanych przez warstwę formularza i skrypty w języku JavaScript. Oba te mechanizmy: zarówno OCG jak i Form, przeznaczone są do nieco innych zastosowań.

1 Wstęp: książka interaktywna

Przed kilkadziesiąt laty, jeszcze jako student, miałem okazję podziwiać wystawę książki alternatywnej prezentowaną w Auli Głównej Politechniki Warszawskiej. Prezentowane „książki” ze szmatek i guzików, z elementami wysuwanymi i rozkładanymi, były dla mnie zaskoczeniem. Wydawało mi się wtedy, że książka jest do czytania, a nie do zabawy i własności interaktywne są w niej zupełnie zbędne.

Ćwierć wieku później „książeczki” tego rodzaju stały się powszechne, a ostatnio zaczynają mieć w sobie coraz więcej gadającej, śpiewającej i grającej, a nawet świecącej elektroniki. Jak widać, życie potrafi zaskakiwać nas na każdym kroku.

Książka dzisiaj to nie tylko treść, nie tylko rozbudowane indeksy tematyczne, nie tylko zaawansowana matematyka. Nawet nie tylko powszechne już dzisiaj hiperłącza. Od książki oczekuje się coraz częściej, aby dynamicznie reagowała na potrzeby czytelnika, dostosowując się do jego potrzeb, a nawet więcej: zastępując go w wielu czynnościach, które do niedawna były przypisane wyłącznie człowiekowi.

Oczywiście książka wydrukowana na papierze z celulozy nie jest w stanie robić tego, do czego jest zdolna ogólnie pojęta elektronika. Ale dzisiejsza książka od samego początku jest elektroniczna. Najpierw poprzez elektroniczny skład, potem poprzez elektroniczną dystrybucję, aż do elektronicznej prezentacji na różnych formach e-papieru albo komputerowo sterowanych wyświetlaczy.

2 Format PDF

Od dziesięcioleci podstawowym formatem komputerowym dla książki jest PDF¹. Alternatywą dlań jest format HTML², przeznaczony do specyfikacji treści prezentowanych na stronach internetowych. Obydwa formaty obsługują dzisiaj pewne zachowania dynamiczne, realizowane przez komputer wyświetlacza, w szczególności interpretacyjny język programowania JavaScript³. Fundamentalną różnicą w tych podejściach jest nie tylko moc specyfikacyjna formatu (PDF umożliwia specyfikację dużo ambitniejszej typografii niż HTML) ale także moc opisowa języka interpretacyjnego, dynamicznie wpływająca na prezentowaną treść (JavaScript w HTML może znacznie więcej niż w PDF, co nie pozostaje bez wpływu na bezpieczeństwo rozwiązania).

3 Akordy nad tekstem

Wydaje się, że autor publikacji ma wybór. Treść zdecydowanie interaktywną może prezentować w formacie HTML, natomiast treść ambitną typograficznie prezentować w formacie PDF. Co zrobić, jeśli dla prezentowanej treści istotne są oba aspekty i z żadnego z nich nie można zrezygnować? Takim przypadkiem okazał się śpiewnik liturgiczny, w którym nad tekstem trzeba w miarę precyzyjnie umieścić symbole akordów gitarowych, określające zmieniającą się harmonię melodii. Należy to zrobić tak, aby niezależnie od metryki fontu akord był zawsze umieszczony precyzyjnie w tym samym miejscu tekstu, zazwyczaj na początku konkretnej sylaby, której nuta melodii dotyczy.

¹Format PDF (ang. *Portable Document Format*, przenośny format dokumentu) — format plików służący do prezentacji, przenoszenia i drukowania treści tekstowo-graficznych, stworzony przez firmę Adobe Systems. Język PDF jest okrojoną wersją języka programowania PostScript wzbogaconą o elementy hipertekstowe (łącza). Okrojenie sprowadza się do zmiany języka opisu strony z języka programowania na język specyfikacji. 2 lipca 2008 Międzynarodowa Organizacja Normalizacyjna ogłosiła PDF 1.7 za obowiązujący standard ISO 32000-1:2008. W lipcu 2017 roku został opublikowany standard ISO 32000-2:2017, później zrewidowany jako *ISO 32000-2:2020 Document management — Portable document format — Part 2: PDF 2.0*. Ten ostatni standard w klauzuli 8.11 opisuje warstwy OCG.

²Format HTML (ang. *HyperText Markup Language*, hipertekstowy język znaczników) — język znaczników stosowany do tworzenia dokumentów hipertekstowych.

³JavaScript — skryptowy (interpretacyjny) oraz wieloparadygmatowy język programowania, stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych. Twórcą JavaScriptu jest Brendan Eich.

Takie rzeczy umożliwia format PDF i system \TeX , który panuje nad linią bazową składanego tekstu oraz umożliwia wysuwanie materiału ponad tekst, niemalże nie naruszając duktu pisma (z dokładnością do podcięć). Jeśli akord nie jest typowy, sam symbol może nie wystarczyć i wtedy układ palców na chwytni gitary można zaprezentować w postaci obrazka złożonego \LaTeX owym środowiskiem `picture`. Fragment takiego składu prezentuje Ilustracja 1. Sposób pisania komend składających akordy jest tutaj rzeczą drugorzędną i zainteresowanych można odesłać do źródeł⁴.

W obliczu aniołów (Ps138)

<p> C E d₇⁹ </p> <p> W W obliczu aniołów pragnę Cię uwielbiać, G E w obliczu aniołów chcę Ci psalmy śpiewać; d₇⁹ sławić będę Twoje imię F E za Twą wierną miłość do mnie. </p>	<p> C E d₇⁹ </p> <p> K Pan jest Najwyższy, widzi pokornego, G E poznaje pysznego, patrzy nań z daleka; d₇⁹ ale gdy ja chodzę pośród utrapienia F E wyciągasz rękę, Twoja prawica mnie zbawia. </p>
--	--

Ilustracja 1: Tekst pieśni z akordami i diagramami chwytów (fragment)

3.1 Transpozycja tonacji

Akordy w muzyce są związane z tonacją i dla każdej tonacji będą inne. Oczywiście akordy można transponować z jednej tonacji do drugiej i sprowadza się to do prostej arytmetyki opartej na *kole kwintowym*⁵. Taką operację można bez większych problemów przeprowadzić mechanizmami arytmetyki wbudowanymi w \TeX a podczas składu, uzyskując każdorazowo w wynikowym pliku PDF akordy w wymaganej tonacji, ale będzie to rozwiązanie statyczne, realizowane w czasie składu, a nie użytkownika.

Zdarzają się jednak sytuacje, gdy trzeba zmienić tonację *ad hoc*, bez możliwości przeskładania dokumentu, podczas prowadzenia śpiewu, np. z powodu chrypki wykonawcy. Narzuca się pomysł, aby posłużyć się kodem JavaScript, który — aktywowany przez dotknięcie przycisku na stronie — przeliczy i podmieni akordy na odpowiednie w danej sytuacji. Każdy, kto próbował JavaScriptu w środowisku HTML, wie, że to jest zupełnie naturalne podejście. Ale niestety! Specyfikacja PDF, a konkretniej *JavaScript for Acrobat API Reference*, nie przewiduje interfejsu do modyfikacji treści składu z poziomu języka JavaScript. Skład stron gotowego pliku PDF jest niemodyfikowalny, przynajmniej w naturalny sposób. Wpływ na treść dokumentu PDF mamy niewielki i jedynie w dwóch sytuacjach:

1. przebudowy zaawansowanym oprogramowaniem redakcyjnym, czyli w rzeczywistości przeskładania fragmentów dokumentu od nowa;
 - oprogramowanie posiadające takie funkcjonalności jest płatne (np. Adobe Acrobat Pro),
 - efekt redakcji pliku PDF z gotowym składem jest zwykle bardzo mizerny, obciążony brakami w fontach oraz brakami w wiedzy o całości składu, jak np. wiedzy o łamaniu akapitów i zależnościach między obiektami typograficznymi,
 - nadal takie oprogramowanie nie daje możliwości wspomaganie algorytmami automatyzującymi np. w celu transpozycji tonacji akordów;
2. wypełniania pól formularza, czyli specjalnej konstrukcji „kładzonej” na wierzchu składu i przeznaczonych do wprowadzania informacji, która następnie ma być przesłana albo z całym plikiem PDF albo tylko jako dane wprowadzone do późniejszej obróbki na serwerze zbierającym dane;
 - modyfikację zawartości pól formularza powinno dać się wspomagać programami w języku JavaScript,


⁴Kompletny kod źródłowy dostępny jest na stronie <http://spiewnik.andrzej.odyniec.info>.

⁵Koło kwintowe to schemat przedstawiający gamy (tonacje) uszeregowane według zmieniającej się liczby znaków przykluczowych. Nazwa wywodzi się od kwintowego pokrewieństwa między sąsiadującymi tonacjami, rozmieszczonymi na kole według zmieniającej się liczby znaków przykluczowych. W jednym kierunku w każdej następnej gamie przybywa dodatkowo jeden krzyżyk, przy czym tonika gamy (nadająca jednocześnie jej nazwę) jest dominantą gamy wyjściowej (poprzedzającej). W kierunku przeciwnym zwiększa się liczbę bemoli, a tonika jest subdominantą gamy wyjściowej.

- zawartość wypełnianych pól formularza służy wyłącznie przesłaniu informacji i z natury jest bardzo uboga typograficznie, jeśli w ogóle można tutaj mówić o typografii.

Żadna z powyższych własności formatu PDF nie spełnia oczekiwań dotyczących automatycznej transpozycji tonacji i to nie tylko ze względu na brak możliwości sterowania programowego albo ubogiej typografii formularza, ale także absolutnej niemożliwości transpozycji diagramów nietypowych chwytów akordowych, których przeliczyć się już nie da i należałoby je ponownie przeszkłać.

3.2 Jak to zostało zrobione?

Podczas prezentacji pliku PDF w wersji elektronicznej możemy zobaczyć niemożliwe. Nad tytułem pieśni w wersji książki obsługującej transpozycje tonacji *ad hoc* zostały złożone prostokąty z różnymi tonacjami docelowymi. Dotknięcie (kliknięcie) takiego obszaru skutkuje zmianą wszystkich akordów nad tekstem pieśni oraz pogrubienie aktywnej ramki. Dotknięcie ramki z symbolem  przywraca oryginalną tonację akordów. Efekty te można obejrzeć w złożonych plikach śpiewników w katalogu <http://spiewnik.andrzej.odyniec.info>, ze słowem Tonacje w nazwie pliku. Należy zdecydowanie zalecić tutaj przeglądarkę Adobe Acrobat Reader DC instalowaną ze strony producenta.

3.2.1 Warstwy OCG

Zgodnie z obietnicą w tytule, efekt transpozycji uzyskujemy poprzez używanie rzeczy niezgodnie z przeznaczeniem. Tą rzeczą jest mechanizm warstw zwany przez Adobe skrótem OCG (ang. *Optional Content Group*). Skoro PDF nadaje się do opisu zarówno tekstu, jak i grafiki, jest idealny do składu rysunków nie tylko technicznych, a dotyczących różnych projektów. Tymczasem rysunki tego typu grupują obiekty na różnych warstwach tak, aby widoczność poszczególne warstw można było włączać lub wyłączać. Był to prawdopodobnie główny powód wyposażenia formatu PDF zarówno w słownik warstw (grup) z informacją o ich nazwie i widoczności (drukowalności) oraz grupowania specyfikacji obiektów identyfikatorem wybranej warstwy. W ten sposób będzie możliwym np. wyłączenie warstwy z projektem instalacji elektrycznej gdy zajmujemy się hydrauliką i odwrotnie.

Zastosowanie warstw OCG jako narzędzia transpozycji tonacji akordów sprowadza się do wielokrotnego składania akordów w tym samym miejscu, za każdym razem w innej tonacji oznakowanej identyfikatorem warstwy OCG zbudowanym z identyfikatora pieśni i numeru kolejnego tonacji.

Jest dostępnych szereg pakietów \LaTeX owych obsługujących warstwy. Do obsługi warstw z tonacjami akordów używam pakietu `ocg-p`, który stworzył Werner Moshhammer. Przełączania widoczności warstw w stylu *radio button* (jedna włączona, pozostałe wyłączone) obsługuję własnym makrem \TeX owym, chociaż są też pakiety, które mają taką funkcjonalność wbudowaną.

3.2.2 Ograniczenia rozwiązania

Różnych tonacji na kole kwintowym jest dwanaście. Składane są tylko tonacje wcześniej zadeklarowane. Jest to pewne ograniczenie, ale nie do wszystkich tonacji transpozycje mają sens, więc ograniczenie możliwych transpozycji do kilku sensownych nie jest czymś tragicznym.

Wprawdzie warstwy OCG są w standardzie PDF od wielu lat, jednak do standardu ISO zostały wpisane dopiero w roku 2020 i tym samym nie są obsługiwane przez wiele przeglądarek. W szczególności pod systemem Android, jeśli są nawet obsługiwane, to ewidentnie nieprawidłowo.

Typowym objawem jest ignorowanie operacji `/SetOCGState`, która włącza i wyłącza widoczność warstw. Wtedy przełączanie tonacji na alternatywne zwyczajnie nie działa.

Jeszcze bardziej przykrym objawem jest ignorowanie przez przeglądarkę słownika warstw określającego stan początkowy widoczności warstw i włączanie widoczności wszystkich, co powoduje nieakceptowalny bałagan.

Wreszcie zdarza się, że przeglądarki wywracają się ze względu na zbyt dużą liczbę różnych warstw OCG w pliku PDF. Dla kilkuset pieśni po kilka tonacji w pliku PDF mamy około tysiąca warstw. Nie należy się spodziewać, aby projekty inżynierskie lub architektoniczne używały aż takiej liczby warstw.

Pomimo to mechanizm warstw użyty niezgodnie z przeznaczeniem działa i przeglądarka Adobe Acrobat Reader DC radzi sobie z nim świetnie. Doskonale też obsługuje go starsza przeglądarka Adobe Reader 9, którą przy pewnym wysiłku można zainstalować pod różnego rodzaju Linuksami.

Można mieć nadzieję, że w ciągu najbliższych lat producenci przeglądarek PDF dociągną je do standardu *ISO 32000-2:2020 Document management — Portable document format — Part 2: PDF 2.0*, który w klauzuli 8.11 specyfikuje obsługę warstw OCG.

4 Zakładki na życzenie

Każda książka nieodmiennie kojarzy się z zakładkami, które od dawien dawna wkładano pomiędzy kartki, aby później kontynuować przerwane czytanie. Jest też inny rodzaj zakładek, który znał każdy student piszący przed laty pracę magisterską; nazywano je *fiszkami*⁶. Istniał jeszcze jeden rodzaj realizacji zakładek, którego używali muzycy: to podział na oddzielne arkusze lub niewielkie zeszyty z nutami konkretnych utworów, znanymi też jako *sheet music* tak, aby układając je przygotować materiał do programu koncertu.

Zakładki książkowe są realizowane zarówno przez format PDF, jak i przez przeglądarki tego formatu.

4.1 Rodzaje zakładek

4.1.1 Hiperłącza

Format PDF przewiduje hiperłącza, których kliknięcie przenosi nas w inne miejsce dokumentu. Takie hiperłącza nazwiemy statycznymi, co znaczy, że są tworzone podczas składu dokumentu. Hiperłącze sprowadza się do obszaru klikalnego i połączonego z nim punktu w innym miejscu dokumentu i w pewnym sensie spełnia funkcję zakładki.

4.1.2 Zakładki pliku PDF

Hiperłącza takie można też umieścić w sekcji zakładek struktury pliku PDF. System składu zazwyczaj umieszcza w zakładkach rozdziały i podrozdziały. Zakładki takie także adresują konkretne punkty na stronach publikacji. Podstawowe wersje przeglądarek nie przewidują możliwości edycji statycznych hiperłączy i zakładek. Dopiero płatne wersje niekiedy umożliwiają dodawanie do sekcji zakładek własnych zakładek użytkownika. Wersje mobilne np. pod Androidem takiej funkcjonalności raczej nie mają.

4.1.3 Zakładki aplikacji

Sama przeglądarka PDF-ów może zapamiętać numery stron pod zakładkami przez siebie organizowanymi i ewentualnie zapamiętywanymi dla każdego pliku PDF oddzielnie na urządzeniu, na którym przeglądamy plik PDF. Takie zakładki są zazwyczaj domeną niektórych przeglądarek mobilnych, instalowanych na tabletach i smartfonach. Jednak przeznaczenie takich zakładek do zapamiętania, gdzie skończyliśmy czytać, czyni trudnym wykorzystanie ich do scenariusza programu.

⁶Fiszka (fr. *fiche*) — kartka w kartotece dotycząca określonego hasła i zawierająca związane z nim informacje lub notatki. Fiszki wykorzystuje się m.in. w katalogach bibliotecznych. Te tzw. *fiszki bibliograficzne* zawierają tytuł książki, jej autora, miejsce i rok wydania itp. (zob. zapis bibliograficzny). Fiszki wykorzystuje się także w innego rodzaju katalogach. Mianem tym można w znaczeniu ogólniejszym określić dowolną, zwykle niewielkich rozmiarów, kartkę zawierającą notatki.

Może to zrobić zewnętrzna aplikacja (np. Setlist Helper and Song Book), która pamięta, w jakich PDF-ach, i na których stronach są jakie utwory, teksty czy nuty i w zgodzie z programem koncertu będzie otwierać strony z zaplanowanych plików. Takich rozwiązań używają zawodowi muzycy.

4.2 I to czwarte... niezgodnie z przeznaczeniem

Żadne z powyższych rozwiązań nie daje wygody w planowaniu przez kantora-psalmistę śpiewów na liturgię. Pozostaje zorganizować zakładki samemu, czyli skakać do etykiety lub strony wywołaniem przez JavaScript. Jednak sekcja zakładek pliku PDF nie przewiduje modyfikacji *ad hoc* ani tym bardziej jakiegokolwiek obsługi przez skrypty.

Rozwiązaniem jest Formularz zdefiniowany przez format PDF, jako warstwa, na której użytkownik wypełnia wyspecyfikowane pola, aby je potem wydrukować albo nawet wyemitować na jakiś serwer. Zgodnie z obietnicą w tytule prezentacji, formularz zdecydowanie nie jest przeznaczony do organizacji zakładek. Poza tym ręczne wypełnianie formularza informacjami o kolejnych pieśniach nie jest wygodnym rozwiązaniem. W sukurs przychodzi język JavaScript, którego polecenia można podczepić pod łącza.

Należy pamiętać, że nie każda przeglądarka PDF-ów będzie: obsługiwać skrypty w języku JavaScript; zapewniać pełną obsługę oraz dostęp z JavaScript API do formularzy; dawać dostęp do wszelkich obiektów i metod wyspecyfikowanych przez standard Adobe Acrobat PDF, a dotyczących tak obsługi struktur opisujących formularze, jak i przemieszczanie się po dokumencie; umożliwiać korzystanie z *Document Level JavaScript*, czyli dać dostęp interpreterowi przeglądarki do kodu w tym języku osadzonym podczas budowy dokumentu w pliku PDF.

Niekiedy obsługę skryptów JavaScript trzeba włączyć w preferencjach przeglądarki, gdy domyślnie jest wyłączona. Uważa się, że skrypty, jako programy komputerowe, w swojej naturze mogą stanowić zagrożenie dla bezpieczeństwa systemu komputerowego i sieci.

4.2.1 Realizacja zakładek w formularzu

Podstawą są operacje deklarowania formularza oraz umieszczania kodu JavaScript w pliku PDF. Do deklarowania formularzy wystarcza w zupełności pakiet `hyperref`, który stworzył nieodżałowany Sebastian Rahtz, a aktualnie prowadzi Heiko Oberdiek. Operacje tam zdefiniowane umożliwiają nie tylko deklarowanie pól formularza, ale i wiązania z nimi kodu JavaScript. Rzecz w tym, że kod ten nie może być nazbyt długi i rozbudowany.

Dlatego drugie cenne rozwiązanie stworzył Donald P. Story i jego zespół, tworząc `AcroTeX`. Używam tutaj pakietu `insdljs`, który umożliwia wyeksportowanie z `TeX`a kodu JavaScript, a następnie umieszczenie go w pliku PDF jako *Document Level JavaScript* tak, aby ze zdefiniowanych tutaj funkcji mogły korzystać obsługi podczepione do pól formularza.

Aby dodać pieśń do formularza, wystarczy kliknąć w symbol zakładki przy tytule pieśni. Podczepione tam wywołanie funkcji JavaScript z przekazanymi przez parametry informacje o tytule i położeniu dopisze te informacje do wolnego wiersza tabeli z zakładkami. Użycie przycisku w tejże tabeli otwiera książkę na tej pieśni.

Dopisanie kilkuset wierszy kodu JavaScript pozwoliło rozwinąć funkcjonalność formularza z zakładkami, jak eksportowanie zawartości, sortowanie tablicy po kolumnach czy wczytywanie wyeksportowanych treści i ich kompilacja. Efekty te można obejrzeć w złożonych plikach śpiewników w katalogu <http://spiewnik.andrzej.odyniec.info>, ze słowem Alpha w nazwie pliku. Należy zdecydowanie zalecić tutaj przeglądarkę Adobe Acrobat Reader DC instalowaną ze strony producenta.

4.2.2 Ograniczenia i niespodzianki

Jak to zwykle bywa z używaniem rzeczy niezgodnie z przeznaczeniem, podróżujemy po nieprzetartych szlakach i napotykamy na różne kłopoty.

Tytuły pieśni w LuaT_EXu, jak cała zawartość tekstowa, jest utrzymywana w kodowaniu UTF-8. Tymczasem łańcuchy w JavaScript osadzonym w PDF operują kodowaniem UTF-16LE. Jak długo w tytułach mamy znaki ASCII, wszystko działa idealnie. Gdy pojawią się polskie (włoskie, hiszpańskie) znaki diakrytyczne, w formularzu znajdziemy „robaczki”. Oczywiście nie wszędzie. W pozycjach menu do pól wyboru kombi (ang. *combo box*) z polskimi diakrytykami jest dobrze. W treści parametrów funkcji już nie. Na dodatek dziwności konwersji kodowania są różne dla różnych przeglądarek PDF w różnych systemach operacyjnych. Protezą okazało się napisanie własnej funkcji korygującej błędną kowersję między kodowaniami.

To, które funkcje API są dostępne i będą działać, jest kwestią mniej lub bardziej przypadkową. Np. metody `Alert` wyświetlające powiadomienia i ostrzeżenia działają różnie (lub nie działają) w zależności od przeglądarki i systemu operacyjnego. Wczytywanie plików albo działa, albo nie działa. Nie mówimy tutaj o zapisywaniu plików, które, powszechnie uznane jako niebezpieczne, stanowiące zagrożenie, nie działa, chyba że mamy wersję płatną przeglądarki.

Zapewne z racji niewielkiego lub żadnego wykorzystania pewnych funkcjonalności, niektóre są obciążone dyskwalifikującymi ich używanie błędami. Do sortowania tablic z zakładkami naturalne wydawało się użycie wbudowanej w standard JavaScript metody `sort` dla tablic. Ale w przeglądarkach Xodo dla Androida takie sortowanie daje efekty przypadkowe, żeby nie powiedzieć: komiczne. Rozwiązaniem okazało się napisanie własnej funkcji sortującej.

5 Podsumowanie

Najmniej przykrych niespodzianek sprawiają przeglądarki Adobe Acrobat pod Windows, instalowane ze strony producenta, bynajmniej nie ze sklepu Microsoft. Jest to trochę rozczarujące, bowiem dzisiaj przenośnym urządzeniem do prezentacji książek elektronicznych jest jednak tablet z systemem Android lub iOS,

JavaScript i warstwy OCG są jednak funkcjonalnościami w PDF dość młodymi, bo liczącymi zaledwie dziesiątki lat. Można mieć nadzieję, że niebawem ich implementacje będą coraz doskonalsze i coraz bardziej zgodne ze standardami.