Fabylon Press · Bookmaker Mathlab

.fab

matemaattisena kirjaformaattina

Tiivis säiliö muotouskolliselle kirjallisuudelle, energiatehokkaalle jakelulle ja nopealle ensimmäiselle sivulle.

Matematiikka, muodon säilyttäminen ja energiapolku luettua kirjaa kohti

Tekninen artikkeli Fabylonin `.fab`-säiliöteknologiasta. Teksti käsittelee matemaattista rakennetta, pakkausintuitiota ja käytännön energiansäästöä — paljastamatta säiliön sisäisiä laboratoriosalaisuuksia.

1,8×	0,06 ms	15×
pienempi kuin EPUB	ensimmäinen sivu	pienempi kuin PDF

Tiivistelmä

Minimaalinen kirjaesitys ei ole sama kuin minimaalinen HTML-esitys

`.fab` on suunniteltu EPUBin vaihtoehdoksi tilanteisiin, joissa kirja ei ole ensisijaisesti verkkosivu vaan typografinen objekti. Sivut, säkeistöt, rivitys, sisennykset, kuvat, marginaalit, recto-verso-logiikka ja lukujärjestys muodostavat yhdessä matemaattisen rakenteen.

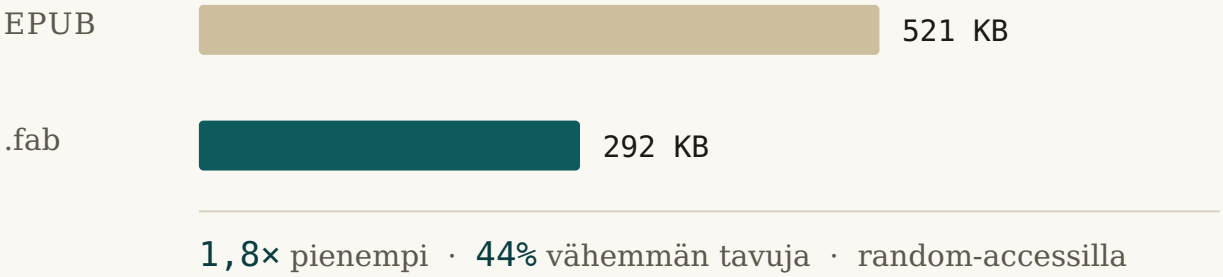
EPUB on W3C-spesifikaation mukaan OCF-ZIP-säiliö, joka sisältää XHTML/SVG-dokumentteja, manifesteja ja metatietoja. Se on laajasti yhteensopiva, mutta käsitteellisesti se pysyy lähellä arkistossa olevia verkkodokumentteja. `.fab` valitsee toisen suunnan: se mallintaa kirjan *itsensä*.

Oikeilla kirjoilla Bookmaker mittaa `.fab`in ja EPUBin väliseksi suhteeksi noin 0,56 — noin **1,8× pienempi**, kun asettelu säilyy, ensimmäinen sivu avautuu noin kymmenesosamillisekunnissa (noin kymmenen kertaa nopeammin kuin EPUB) ja mikä tahansa sivu on suoraan luettavissa. .fab:in erottaa ei vain pakkaus vaan *esitysmelun välttäminen*: nopeus, paikallinen purettavuus ja lohkokohtainen eheys.

Mathlab-mittaus

`fab/epub`	`~0,56`	—
`vs EPUB`	`~1,8×`	pienempi
`vs PDF`	`~15×`	pienempi
`eka sivu`	`~0,06`	ms
`eka vs EPUB`	`~10×`	nopeampi
`satunn. sivu`	`O(1)`	—
`eheys`	`CRC`	/lohko
`layout_roundtrip`	`true`	—

Mittaus oikeilla kirjoilla · Fabylon Bookmaker Mathlab



Kuva 1 — viisi oikeaa kirjaa: saman julkaisun koko EPUB- ja `.fab`-muodossa. `.fab` säilyttää lisäksi minkä tahansa sivun suoran luettavuuden ja lohkokohtaisen eheyden.

1 Ongelmanasettelu

Kirjallinen kirja ei ole kokoelma HTML-tiedostoja. Se on pikemminkin järjestetty typografinen varieteetti:

$$B = (T,\ P,\ L,\ G,\ A,\ M)$$

jossa T on teksti, P sivutus, L lukujärjestys, G geometria, A assetit ja M metatiedot.

EPUB esittää tämän pääosin muodossa XHTML plus CSS plus manifesti plus ZIP. Se on tehokasta ja hyödyllistä mutta epäsuoraa: asettelu rekonstruoidaan selainmaisella järjestelmällä. `.fab` käyttää suorempaa esitystä, jossa kirjan primitiivit ovat ensiluokkaista dataa:

$$B \longrightarrow \mathrm{C}_{\mathrm{fab}}$$

Ero on perustavanlaatuinen. EPUB sanoo: *tässä* ovat verkkodokumentit, jotka voivat yhdessä muodostaa kirjan. `.fab` sanoo: *tässä* on kirja kirjana.

2 Asettelun säilyminen invarianttina

Konversiota varten määrittelemme asetteluprojektion $\Pi(B)$. Hyvän kirjaformaatin on säilytettävä tämä projektio toleranssin ε sisällä:

$$\Pi(B) = (P,\ G,\ L)$$

$$d\big(\Pi(B),\ \Pi(\hat{B})\big) \ \le\ \varepsilon$$

Mathlab-fixturessa pätee `layout_roundtrip = true`. Olennaista on, ettei `.fab`in tarvitse päätellä asettelua CSS:stä: rivit, säkeistöt, span-elementit, sisennys, kuvien sijoittelu ja selkä tallennetaan kanonisena kirjadatana. Näin sisällön ja esityksen ero on terävämpi kuin yleisessä verkkosäiliössä.

Asetteluetäisyyden tarkka normalisointi ja toleranssit pysyvät laboratoriosalaisuutena. Julkisesti olennaista on ennen kaikkea periaate: `.fab` optimoi *muodon säilyttämisen* ehdolla, ei yleisen arkistoinnin ehdolla.

3 Entropia ja esityksen lisäkuorma

Formaatin tallennuskustannukset voidaan kirjoittaa karkeasti informaatiosisällön, formaattilisäkuorman ja redundanssin summana. Tavoitteena on minimoida tämä lisäkuorma ja säilyttää samalla nopea avaaminen:

$$S = H(B) + O_f + R_f$$

$$\min \left(O_f + R_f + \alpha\, t_{\text{open}} + \beta\, t_{\text{first}} \right)$$

Tässä $H(B)$ on kirjan informaatiosisältö, O_f formaatin lisäkuorma ja R_f esitysvalinnoista johtuva redundanssi. EPUBissa lisäkuorma on suhteellisen suuri ZIP-rakenteen, XML/XHTML:n, manifestien, tiedostonimien, säiliösääntöjen ja CSS-serialisoinnin vuoksi — yhteentoimivuuden hinta. `.fab`issa pyritään minimoimaan tämä lisäkuorma ja redundanssi niin kauan kuin asettelun säilyminen pysyy ε:n sisällä.

Sisäistä järjestystä, sanakirjan esitäyttöä, kenttäkoodausta ja delta-normalisointia ei tässä paljasteta tarkoituksella. Tieteellinen väite on abstraktimpi: jos objekti on kirja, myös koodikirjan tulisi olla kirjanmuotoinen.

4 Paikallinen purettavuus

Lukijan ei pitäisi joutua rekonstruoimaan koko kirjaa näyttääkseen sivun 7. Abstraktissa muodossa säiliö koostuu indeksistä, metatiedoista, asseteista ja itsenäisesti purettavista sivulohkoista; ensimmäinen näkyvä sivu vaatii tällöin vain ensimmäisen lohkon sekä indeksin ja metatiedot:

$$C = \left(I,\ M,\ A,\ \{p_i\} \right)$$

$$\text{ensimmäinen sivu} \approx D(p_1) + D(I) + D(M)$$

Ensimmäinen sivu maksaa siis $D(p_1) + D(I) + D(M)$, ei $D(\sum_i p_i)$. Tämä selittää, miksi Mathlab-mittaus pitää arvot `open_ms` ja `first_page_ms` erittäin alhaisina fixturessa.

Paikallinen purettavuus ei tarkoita vain nopeutta. Se vähentää myös akun kulutusta, muistin huippukuormaa ja lämpökuormaa yksinkertaisissa lukulaitteissa.

5 Miksi `.fab` on kirjaluonteinen

`.fab` säilyttää kirjallisen rakenteen kirjallisena rakenteena: säkeistöt pysyvät säkeistöinä, rivit pysyvät riveinä, sisennys pysyy typografisena datana, kuvat pysyvät sijoitteluobjekteina, sivut pysyvät sivuina ja selkä pysyy lukujärjestyksenä.

EPUBissa suuri osa tästä tehdään epäsuorasti HTML:n ja CSS:n kautta. Se toimii usein hyvin, mutta on semanttisesti meluisaa. Romaaneille se on hyväksyttävää; runoudelle, typografiselle proosalle ja painoasulle uskollisille julkaisuille kiertotie tulee näkyväksi.

`.fab` ei siis ole EPUB toisella tiedostopäätteellä. Se on yritys tuoda minimaalinen kirjaesitys lähemmäs kirjan *itsensä* matemaattista rakennetta.

6 Energia ja käytännön säästö

Energiansäästö syntyy neljästä kerroksesta: vähemmän tallennustilaa, vähemmän CDN- ja välimuistireplikaatiota, vähemmän verkkosiirtoa ja vähemmän purkutyötä laitteilla. Yksittäinen kirja merkitsee vähän; miljoonat lataukset *merkitsevät*. Käytämme mallia:

$$E_{\text{save}} = N \cdot D \cdot S_{\text{epub}} \cdot (1 - r) \cdot \eta$$

Tässä N on nimekkeiden määrä, D lataukset nimekettä kohti, S_{epub} keskimääräinen EPUB-koko gigatavuina, $r = S_{\text{fab}}/S_{\text{epub}}$ ja η energiaintensiteetti siirrettyä ja käsiteltyä gigatavua kohti.

> LASKUESIMERKKI
>
> `10 000` kirjalle, joista kullakin on `10 000` latausta ja `2 MB` säästö latausta kohti, syntyy noin `200 000 GB` vähemmän datansiirtoa. Hinnalla 0,01–0,05 kWh/GB se on `2 000–10 000 kWh`. Suuremmilla kirjastoalustoilla tai koulujakelussa tämä kasvaa nopeasti.

IEA raportoi, että datakeskukset käyttivät vuonna 2024 noin 415 TWh ja että datasiirtoverkot käyttivät vuonna 2022 noin 260–360 TWh. Tiedostoformaatit eivät siis ole yksityiskohta: jokainen ylimääräinen tavu tallennetaan, kopioidaan, välimuistitetaan, skannataan, synkronoidaan ja avataan uudelleen.

7 Rajat ja rehellisyys

Rehellisyys luvuista: aiempi synteettinen mikro-fixture näytti suhteen 0,061, mutta se ei ole edustava — se oli keinotekoisen toistuva testidokumentti. Oikeilla kirjoilla suhde on noin 0,56 (noin 1,8× pienempi kuin EPUB), vaihdellen nimekkeittäin (0,40–0,64); runsaasti kuvitetut tai monimutkaiset kirjat voivat olla suurempia. `.fab`:in suurin etu ei siis ole koko EPUBiin verrattuna, vaan paljon nopeampi ensimmäinen sivu, minkä tahansa sivun suora luettavuus ja lohkokohtainen eheys.

Myös EPUB pysyy tärkeänä: se on kypsä vaihtostandardi. `.fab` on tarkoitettu Bookmaker-natiiviksi julkaisuformaatiksi tilanteisiin, joissa muotouskollisuus, kompaktius ja nopea ensimmäinen sivu painavat enemmän kuin yleinen verkkoyhteensopivuus. Oikea vertailu ei siis ole ideologinen: **EPUB on laaja, PDF on painovakaa, `.fab` on kirjanatiivi.**

8 Päätelmä

Ydinhavainto on tämä: minimaalinen kirjaesitys ei ole yhtä kuin minimaalinen HTML-esitys. `.fab` lähtee tästä havainnosta ja käsittelee kirjan muotoa matemaattisena datana. Tämän ansiosta `.fab` voi olla pienempi, nopeampi ja muotouskollisempi Bookmaker-kirjoille. Teknologia on erityinen, koska se optimoi pakkauksen, asettelun invarianssin ja paikallisen purettavuuden *samanaikaisesti*.

> Lupaus on yksinkertainen: *vähemmän tavuja + nopeampi ensimmäinen sivu + parempi muotouskollisuus = pienempi energiapolku luettua kirjaa kohti.*

Lähteet

W3C — EPUB 3.3 -spesifikaatio. Kuvaa EPUBin OCF-ZIP-säiliönä, jossa on XHTML/SVG-sisältödokumentteja ja säiliösääntöjä.

International Energy Agency — Data centres and data transmission networks. Raportoi verkkojen energiankäytöstä ja datansiirron trendeistä.

International Energy Agency — Energy and AI, executive summary. Raportoi datakeskusten sähkönkäytöksi noin 415 TWh vuonna 2024.

Fabylon Bookmaker Mathlab — mittaus oikeilla kirjoilla (viisi nimekettä). `epub=521 KB → fab=292 KB`, `fab_to_epub≈0,56` (1,8× pienempi), `eka_sivu≈0,06 ms`, `satunnainen_sivu=O(1)`, `CRC per lohko`, `layout_roundtrip=true`.

Tekninen huomautus

Tämä julkaisu ei tarkoituksella sisällä kaikkia sisäisen `.fab`-säiliön yksityiskohtia. Sen tarkoitus on tieteellinen positiointi: kirjanatiivi esitys, asettelun invarianssi, paikallinen purettavuus ja käytännön energiahyöty. Tarkat laboratorioparametrit pysyvät osana Fabylon Mathlab-toteutusta.