

COBISS3 – VKLJUČITEV KONTEKSTNE POMOČI

mag. Zdenka Kamenšek, Teodor Veingerl
IZUM – Institut informacijskih znanosti, Prešernova ulica 17, 2000 Maribor
zdenka.kamensek@izum.si, teodor.veingerl@izum.si

Povzetek

V prispevku predstavljamo korake, ki smo jih morali izvesti za implementacijo kontekstno-odvisne pomoči za programsko opremo COBISS3 – od priprave koncepta, izbire označevalnega jezika in orodij, izobraževanja za uporabo orodij, dopolnitve same programske opreme, prenosa in preoblikovanja vsebine obstoječih priročnikov in vse do priprave nove spletne strani in objave novega priročnika.

Pri tem smo sledili več ciljem: uporabnikom ponuditi čim boljše izkušnjo pri uporabi programske opreme, racionalizirati pripravo uporabniške dokumentacije z ločitvijo njene vsebine od oblike, omogočiti direktni dostop do posameznih delov dokumentacije od zunaj, olajšati navigacijo znotraj nje ter jo hkrati ohraniti uporabno tudi za tiskanje.

Pri prehodu na novi format smo morali rešiti kar nekaj težav znotraj programske opreme in pri pripravi pretvorbe v HTML in PDF. Dogovoriti smo se morali za način dodajanja povezav, konvencijo poimenovanj in način označevanja določenih pomembnih delov besedila ter uskladitev prikaza dokumentacije s preostalimi spletnimi stranmi.

Abstract

IMPLEMENTATION OF CONTEXT-SENSITIVE HELP IN COBISS3 SYSTEM

The paper describes the steps that had to be taken to implement context-sensitive help for COBISS3 manuals – from concept design, markup language and tool selection, training for tool use, updating the software, copying and re-designing the contents of the existing manuals up to the creation of the new website and publishing the new manual.

Several goals were followed in the process: to offer the user the best possible user experience when using the software, to rationalise the preparation of user documentation by separating content and layout, to enable direct access to individual parts of documentation from outside, to simplify the navigation within it and to maintain the option of printing the documentation.

When switching to the new format, several issues within the software had to be resolved along with the preparation of the conversion to HTML and PDF. We defined how to add links, naming conventions, markup type for indicating important types of text and harmonised the documentation web pages with our other web pages.

Ključne besede

kontekstna pomoč, kontekstno-odvisna pomoč, priročniki, COBISS, COBISS3

Keywords

context-sensitive help, contextual help, manuals, COBISS, COBISS3

UVOD

Programsko opremo COBISS sestavljata dva glavna sklopa: COBISS+ in COBISS3. Prvi je namenjen bralcem, raziskovalcem in preostalim zunanjim uporabnikom knjižnic ter ga večina uporabnikov dobro pozna. Drugi sklop je COBISS3 [6] (v nadaljevanju C3 – številka 3 za imenom pomeni generacijo programa), ki je širši javnosti skoraj neznan, zato pa je toliko pomembnejši za knjižničarje pri njihovem vsakdanjem delu. Program teče v načinu odjemalec–strežnik in ga uporabniku ni treba posebej namestiti. Nameščeno moramo imeti le programsko okolje Java. C3 zaženemo s klikom na spletno povezavo – pri čemer se bo program samodejno prenesel na naš računalnik, vzpostavil povezavo s strežnikom, nas vprašal za geslo in po nekaj sekundah ga že lahko uporabljamo. Tak način delovanja programa je za uporabnika zagotovo najpreprostejši in zelo učinkovit. Po začetni avtomatski namestitvi se do odjemalca prenašajo le še podatki, ki se trenutno prikazujejo. Toda ker program ni lokalno nameščen, tudi nima direktno dostopnih datotek z uporabniško dokumentacijo. Za izvedbo kontekstne pomoči smo zato morali uporabiti nekaj trikov.

A najprej definirajmo, kako sploh razumemo izraz *kontekstna pomoč* (oz. *kontekstno-odvisna pomoč*, ang. *context-sensitive help*). Najpreprostejša definicija je: *Uporabniku kadar koli prikazati natanko tiste informacije, ki jih v tistem trenutku potrebuje, glede na to, kar takrat počne* [4].

Ideja takšne pomoči ni nova. V programski opremi se pojavlja že desetletja. Izvedena je lahko na zelo različne načine, ki so uporabnikom včasih bolj, včasih manj preprosto dosegljivi. Najpogosteje jo priključimo s pritiskom na namensko tipko (največkrat F1) ali kliknemo na ustrezeni gumb na zaslonu (običajno gumbek z vprašajem ali kaj podobnega). Program prikaže navodila za okno, ki je trenutno odprto, ali za operacijo, ki je trenutno aktivna, lahko pa celo za posamezno polje, ki ga uporabnik trenutno vnaša [4].

Za učinkovito izvedbo kontekstne pomoči moramo nanjo misliti že od samega začetka načrtovanja in razvoja programske opreme. Toda C3 (vključno s prvima generacijama) je v razvoju že dobrih 30 let in v tem času je doživel mnoge spremembe ter dopolnitve. Na začetku tega obdobja je bila kontekstna pomoč zgolj abstraktna ideja, ki je ni poznal še noben program.

Med ponudniki programske opreme je bila zato splošna praksa izdajanje tiskanih priročnikov. Toda te priročnike je bilo treba posodabljati pri vsaki posodobitvi programja, kar je predstavljalo precejšen strošek. Tiste (redke) programerske hiše, ki niso želele svojih uporabnikov prepustiti na milost in nemilost (takrat sicer bujno cvetočega) trga s priročniki, so jih raje kot v obliki vezane monografije izdajale v obliki vpenjalnih map, v katerih je bilo mogoče ob vsaki novi izdaji nadomestiti le spremenjene strani. Seveda je moralo biti temu podrejeno oblikovanje poglavij, njihovo številčenje in še kaj. Praviloma so bila pri vsakem popravku k tem navodilom za programe dodana še navodila za navodila – se pravi navodila, kam naj uporabniki vložijo posamezne liste, da posodobijo svoje priročnike. Vse to danes zveni precej absurdno, a takrat je bila to najoptimalnejša možnost. Priročnike za C3 je bilo še do nedavnega mogoče posodabljati na tak način.

A medtem se je zgodil internet, se ugnezdil povsod ter prinesel nove želje, potrebe in zahteve uporabnikov. Med drugim: dostopnost dokumentacije v elektronski obliki.

Priročniki za C3 so bili od prehoda na sistem Windows narejeni s programom *MS Word*, zato jih ni bilo težko objaviti v formatu PDF. Z drugimi formati je bilo že težje. HTML, ki ga izvozi

Word, je zelo specifičen in težko prilagodljiv. To sicer ni težava, dokler uporabljamo pretežno Microsoftove brskalnike in želimo obliko ohraniti čim bližje originalni (kar je tudi glavni Wordov namen – zato tudi tako kompliciran HTML). Prilagoditi tak dokument drugim spletnim stranem je že precej težje. Še težje pa je takšno obliko prilagoditi potrebam kontekstne pomoči.

Tudi vsebina priročnikov, prilagojena tisku na papir, se ni najbolje obnesla za kontekstno pomoč.

Tako smo se morali sprijazniti s tem, da priročnike, ki bodo omogočali kontekstno pomoč, napišemo bolj ali manj znova – v novi obliki, z novimi orodji in s prilagojeno vsebino, pri čemer od vsega začetka razmišljamo o kontekstni pomoči.

V prispevku so opisani koraki, ki smo jih izvedli za vključitev kontekstne pomoči v C3: od raziskave o uporabi priročnikov, priprave izhodišč za nove priročnike, do objave novega priročnika in priprave navodil za delo. Namenjen je predvsem tistim, ki razmišljajo o uvedbi kontekstne pomoči za svojo programsko opremo, ali se ukvarjajo s pripravo uporabniške dokumentacije. Morda jim lahko naše izkušnje koristijo, da se izognejo nekaterim pastem in težavam, ki so nam povzročile največ sivih las. Naš pristop prikazuje, da je mogoče v relativno kratkem času najti delujočo rešitev tudi brez dragih in zapletenih namenskih programov za kontekstno pomoč.

METODOLOGIJA

Proces priprave prejšnjih priročnikov

Word, ki smo ga uporabljali za pisanje priročnikov, omogoča veliko fleksibilnost glede oblikovanja, vendar je to oblikovanje tesno povezano z vsebino (kar je seveda bistvo orodij *WYSIWYG* [12] – *kar vidiš, to dobiš*). Kot je omenjeno že v uvodu, so način oblikovanja strani, kazal, številčenje itd. izhajali iz zahteve, da lahko nove ali spremenjene strani v obstoječem natisnjem priročniku zamenjamo in posamezna poglavja vedno znova združimo v celotni priročnik.

Po drugi strani je skupinsko delo na istem dokumentu dokaj problematično. Priročnik je bil razdeljen na posamezna poglavja, shranjena v ločenih datotekah. Nazadnje dodana ali spremenjena vsebina je bila posebej označena. To povzroča težave s skupnimi kazali in ne nazadnje s tem, da so dokumenti, sestavljeni iz več pod-dokumentov, pravi magnet za Wordove hrošče.

Word omogoča, da v njem neposredno rišemo in dodajamo elemente na vključene slike. To nam je povzročalo kup nevšečnosti, ko se je osnovna slika premaknila na drugo stran, naknadno dorisano pa ne (ali obratno). Prav tako nam ni ustrezalo, da so v Wordovih dokumentih notranje hiperpovezave nekaj povsem drugega kot tiste na zunanje dokumente. Še večji problem je, da sta formata *doc* in *docx* v bistvu binarna formata in zato nerodna za shranjevanje v repozitorij za verzioniranje izvirne kode (*Version Control System* oz. *VCS*) ter pregledovanje sprememb znotraj *VCS* – morda z izjemo *Sharepointa*, (ki je okoren, zahteva nakup dodatnih licenc in dobro deluje le na odjemalcih v okolju Windows).

Tako se je vedno znova pokazalo, da je priprava same vsebine v celotnem procesu predstavljala le manjši del časa in truda. Precej več sta ga terjali ureditev videza in oblike strani ter kazal in priprava dodatnega kompleta za tiskanje spremenjene vsebine priročnika.

Kako uporabniki COBISS3 uporabljajo priročnike pri svojem delu?

Glede na to, koliko dela je doslej zahtevala priprava in vzdrževanje priročnikov, je bilo pred odločitvijo, kako jih pripravljati v prihodnje, pomembno od uporabnikov pridobiti informacije o njihovi dejanski rabi in uporabnosti. Zato smo našim uporabnikom – knjižničarjem – posredovali anketo z vprašanji o njihovi uporabi. Anketo smo poslali 193 knjižnicam, odgovorile so 104.

Pri vprašanju, kaj je njihova prva izbira za pomoč (izbirali so lahko med priročniki, pomočjo pri sodelavcih in IZUM-ovo pomočjo), se je pokazalo, da jih več kot polovica najprej poseže po priročnikih, sledi posvet s sodelavci, nato pa uporaba IZUM-ove pomoči. Odgovori so pokazali, da tudi tisti, ki jim uporaba priročnikov ne predstavlja prve izbire, priročnike kljub temu uporabljajo.

Odgovor na vprašanje, ali uporabljajo elektronske ali tiskane priročnike, je pokazal, da večina (okoli 85 %) uporablja e-priročnike. Obe obliki uporablja okoli 10 %, zgolj tiskano obliko pa le okrog 5 % knjižničarjev. Pri tem ni nujno, da tiskana oblika predstavlja priročnik v celoti – uporabniki včasih natisnejo le vsebine, ki jih zanimajo oz. jih v nekem trenutku potrebujejo. Odgovori v anketi so tudi pokazali, da niti tisti uporabniki, ki sicer uporabljajo natisnjene priročnike, teh ne ažurirajo več tako, da bi si natisnili le popravke in dopolnitve (pripravljene v ločeni datoteki) ter jih vložili na ustrezna mesta v priročniku.

Uporabniki programske opreme v veliki meri spremljajo obvestila o novostih in dopolnitvah v programski opremi. Redno jih spremlja več kot 60 %, 25 % pa občasno. Ker so opisi pripravljene ločeno po segmentih, uporabniki pogosto preberejo opise dopolnitev in novosti le za segmente, ki jih uporabljajo.

Prejeli smo tudi nekaj mnenj in predlogov uporabnikov. Pomembno je, da je vsebina priročnikov ažurna in napisana na razumljiv način. Večji poudarek mora biti na osnovnih navodilih za uporabo posamezne funkcionalnosti, vključno z opisom njenega namena. Glede na to, da so uporabniki C3 knjižničarji, so poudarili, da je pomembno, da v priročnikih ni slovničnih napak ter da je izrazoslovje poenoteno. Pomembno je, da lahko po priročniku iščejo. Želijo tudi, da bi lahko iskali po ključnih besedah. Ustrezno oblikovano kazalo je bistveno za uporabo priročnika in iskanje po njem, zato želijo, da so kazala pregledna, saj so priročniki precej obsežni. Želijo interaktivne priročnike – takšne, kjer se neposredno prikaže iskani del, z možnostjo prehoda na celotni priročnik. Priročniki naj vključujejo več primerov in slik. Še vedno si želijo možnost tiskanja priročnika – predvsem posameznih delov, ki so za uporabnika pomembni.

IZHODIŠČA ZA NOVO ZASNOVO PRIROČNIKOV

Pri pripravi nove zasnove smo najprej izhajali iz potreb in zahtev uporabnikov, ki jim je uporabniška dokumentacija namenjena. Pri tem smo se opirali na rezultate ankete in na lastne izkušnje pri delu z uporabniki. Že v izhodišču smo sprejeli odločitev, da mora rešitev omogočiti vključitev kontekstne pomoči. Postopek priprave in vzdrževanja priročnikov mora biti preprostejši in hitrejši, kar bo pripomoglo k zagotavljanju njihove ažurnosti. Odločili smo se tudi, da vsebino priročnikov ločimo od njihove oblike, s čimer zagotovimo poenoteni prikaz in oblikovanje, avtorji priročnikov pa ne izgubljajo časa z oblikovanjem in se lahko posvetijo predvsem vsebini. Upoštevati je treba večjezičnost (priročnike pripravljamo v slovenskem,

srbskem in angleškem jeziku). Njihov končni format mora biti uporabniku zlahka dostopen ter omogočati uporabo hiperpovezav znotraj dokumenta in povezav na zunanje dokumente. Rešitev mora omogočati pretvorbo v format PDF, ki olajša prenos dokumentov k uporabniku in tiskanje. Pri iskanju rešitve je treba upoštevati, da je priprava priročnika timsko delo, ki naj bo mogoče s čim manj "verbalne sinhronizacije" in čim manj zaklepanja dokumentov med delom. Sprotno shranjevanje dopolnitev in popravkov naj bo urejeno v skupnem VCS, ki bo nudil hiter pregled sprememb in prav tako olajšal timsko delo. Rešitev mora omogočati tudi možnost namestitve orodij za pretvorbo dokumentov v končne formate na operacijskih sistemih Windows (delovne postaje avtorjev priročnikov) in Linux (strežniki).

Na podlagi teh izhodišč smo se morali odločiti za:

- uporabniški vmesnik za prikaz pomoči in integracijo s C3;
- izvorni format, v katerem bo napisana vsebina pomoči/priročnikov, in urejevalnik za pripravo vsebine v izbranem izvornem formatu;
- orodja za pomoč pri delu s priročniki (npr. predogled v formatih HTML in PDF);
- orodja, ki bodo omogočala dokončno pretvorbo v izhodne formate, namestitev na strežnike in pripravo testnega okolja; - ustrezni VCS za izvirne dokumente.

PRIKAZ KONTEKSTNE POMOČI

Obstajajo namenska orodja za izdelavo kontekstne pomoči (npr. *Microsoft Foundation Classes*, *Eclipse DTP Help* itd.), ki praviloma zahtevajo, da razvoj lastne programske opreme že od začetka prilagodimo uporabi specifičnega orodja. Če tega nismo upoštevali, je potrebna bolj ali manj obsežna predelava. Narava odjemalskega dela C3, ki teče na uporabnikovem računalniku (aplikacija v Javi, ni instalacije, ni lokalnih datotek, ni neposrednega dostopa do baze podatkov), še dodatno otežuje uporabo takšnih orodij. Poleg tega se nismo želeli vezati na posameznega ponudnika, temveč imeti celoten postopek pod lastnim nadzorom.

Ugotovili smo, da lahko z minimalnimi posegi v sam program zagotovimo dovolj dobro rešitev: klic spletnega brskalnika, ki mu posredujemo ustrezno povezavo na primerno oblikovane spletne strani. Te strani bodo vsebovale sidra (ang. *anchors*) za prikaz specifičnih informacij, hkrati pa bo celotna spletna stran lahko delovala kot enovit priročnik za posamezni segment uporabe.

Da bi to delovalo, je bilo treba zagotoviti možnost preprostega prilagajanja preslikav med poimenovanjem objektov v sami aplikaciji ter poimenovanjem HTML-dokumentov in sider v njih. To smo rešili z navadno tekstovno (.ini) datoteko, ki jo C3 ob zagonu prebere s spletnega strežnika. Tako smo se izognili shranjevanju česar koli na uporabnikov računalnik. Te preslikave bi seveda lahko zapisali tudi v podatkovno bazo, vendar bi morali potem izdelati še program za njihovo urejanje. Zdaj zadošča že urejevalnik besedila.

IZBIRA ORODIJ

Naslednji korak je bila izbira ustreznega urejevalnika dokumentov. Word, ki smo ga uporabljali prej, smo izločili zaradi že navedenih razlogov. Raje smo se ozrli po tekstovnih formatih, kot sta npr. *LaTeX* [9] in *Markdown* [2].

LaTeX je močan, a zapleten. Ima strmo učno krivuljo in dokaj zahtevno instalacijo. Celo ljudje, ki jim je bil prvotno namenjen (tj. pisci matematičnih knjig) danes mnogokrat raje pišejo besedilo v preprostejšem jeziku (kot je npr. *MD*) in *LaTeX* uporabljajo le še za pisanje enačb [9].

Markdown [2] (v nadaljevanju MD) je v različnih inkarnacijah postal *de-facto* standard za dokumentacijo odprtokodnih projektov. Ima sicer nekaj pomanjkljivosti:

- ker je tekstovni format, seveda ni *WYSIWYG* [12], a je preprost in v izvorni obliki še vedno mnogo lažje berljiv kot npr. *LaTeX*. Poleg tega nekateri urejevalniki omogočajo takojšen predogled rezultata.
- je dokaj omenjen ter sam po sebi ne omogoča ne hierarhičnega številčenja naslovov poglavij ne avtomatske izdelave kazal in še česa, vendar obstaja kup dodatnih orodij za pretvorbo, ki to pomanjkljivost odpravljajo.

Po drugi strani ima MD kup dobrih lastnosti:

- Za nas je zelo pomembno, da lahko v izvorno datoteko MD dodamo HTML-kodo, ki se bo neposredno prepisala v izhodni HTML (npr. sidra: ``).
- Preprostost MD že sama po sebi pogojuje ločitev vsebine od oblike, saj je izhodni HTML prav tako preprost (brez balasta), obliko pa mu zlahka določimo naknadno z uporabo CSS in jo s tem prilagodimo svojim preostalim spletnim stranem in svojim spletnim aplikacijam.
- Dodatna orodja omogočajo pretvorbo MD v celo vrsto izhodnih formatov (nekatera celo obratno).
- Ker gre za tekstovni format, ga zlahka shranimo v VCS in uporabljamo vsa njegova dodatna orodja (recimo *Diff* za prikaz sprememb v kodi).
- Za povrh so orodja za uporabo MD zastonj in delujejo v različnih operacijskih sistemih.

Tako smo se odločili poskusiti, kako bi se MD v praksi obnesel za kontekstno pomoč. Kot urejevalnik smo uporabili Microsoftov *Visual Studio Code* [7], ki omogoča predogled izhodnega formata že takrat, ko pišemo izvorni MD.

Za oblikovanje gole kontekstne pomoči bi *VS Code* (s kakšnim dodatkom – *plug-in*) popolnoma zadoščal. Toda, kot rečeno, smo želeli posamezna poglavja te pomoči povezati tudi v skupen priročnik – s hierarhičnim številčenjem poglavij, interaktivnim kazalom in pretvorbo v PDF. Izbrali smo orodje *pandoc* [8], ki ga je preprosto instalirati in poleg vsega naštetega omogoča še uporabo številnih različic MD, podpira različne načine pisanja tabel, opremljanje slik in tabel z naslovi (ang. *captions*), vključevanje kode *JavaScript/ECMAScript* v izhodni HTML, poljubno število datotek CSS in še marsikaj.

Vse te možnosti vključujemo/izključujemo s parametri v ukazni vrstici, kar pomeni, da bi jih moral vsak, ki bo *pandoc* zaganjal, poznati – učna krivulja zanj pa je zelo strma. Zato smo pripravili ukazne (ang. *command* oz. *shell*) skripte za Windows (delovne postaje) in Linux (strežniki), ki avtomatsko zberejo izvorne datoteke in poženejo *pandoc* s potrebnimi parametri. Strežniška skripta poskrbi tudi za samodejno namestitev izhodnih datotek na razvojni spletni strežnik.

Dejansko uporabljamo dva spletna strežnika: razvojni in produkcijski. Med razvojem naslednje verzije C3 vzporedno nastaja tudi kontekstna pomoč zanjo. Njeno delovanje lahko preizkušamo na razvojnem strežniku, ob izdaji nove verzije pa datoteke prenesemo na produkcijski strežnik.

Za pretvorbo v PDF *pandoc* potrebuje dodatni konverter. Privzeta je pretvorba v LaTeX kot vmesni format. Kot smo že omenili, je LaTeX precej zahtevno instalirati in tudi sicer se v našem primeru ni najbolje obnesel. Nadomestili smo ga z *WebKit HTML to PDF* [13], ki je precej preprostejši, a zna vse, kar potrebujemo: pretvoriti HTML v PDF, dodati glave in noge strani, oštevilčiti strani, dodati naslovnico ...

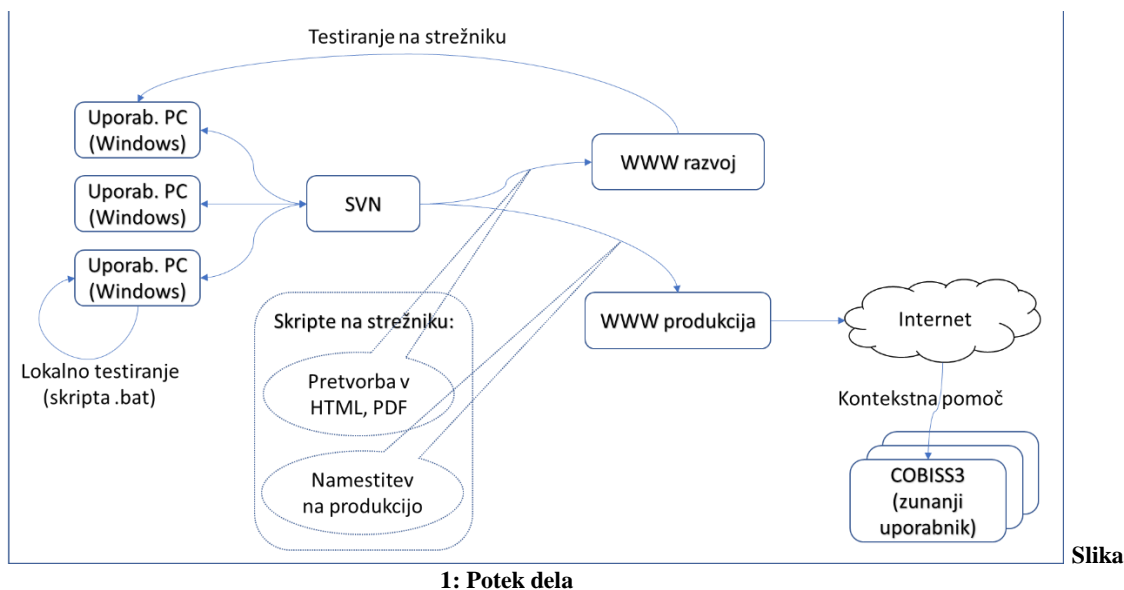
REPOZITORIJI IN ORODJA ZA TIMSKO DELO

Odločitev o shranjevanju priročnikov v repozitorij je odprla nova vprašanja in dileme:

- Kateri VCS uporabiti?
- Kakšna naj bo struktura direktorijev, upoštevajoč številčnost priročnikov (vsak posamezni segment C3 ima svoj priročnik) in večjezičnost?
- Struktura mora omogočati uporabo hiperpovezav, ki ne bodo vezane na jezik priročnika.
- Poimenovanje datotek, ki bo omogočalo, da bodo skripte za pretvorbo in instalacijo lahko samodejno zložile vsebino v pravem vrstnem redu.
- Umestitev druge uporabniške dokumentacije, ki bo sledila (npr. opis novosti, druga navodila za delo ...)

Za shranjevanje, deljenje in verzioniranje izvirne kode v razvojnem oddelku že dolgo uporabljamo VCS *Subversion* [1] oz. *SVN*, nameščen na skupnem strežniku znotraj lokalnega omrežja. Razvijalci odprtokodnih rešitev (zlasti za Linux) imajo praviloma raje *Git* [3], ki je distribuiran (teče lokalno na uporabniških računalnikih), omogoča delo *off-line* in dodatni nadzor pri shranjevanju v skupni repozitorij. Vendar to pomeni tudi nekoliko bolj zapleteno uporabo (dodatni korak oz. dva pri objavi (ang. *commit*) sprememb), kar lahko v začetku zmede celo programerje, uporabnike, ki niso vajeni orodij VCS, pa še prej.

Za dokumentaliste je SVN vsekakor primernejši in preprostejši za uporabo – zlasti z namenski odjemalci, kot je *TortoiseSVN* [10], [11], ki se brezšivno integrirajo v sistemski datotečni brskalnik (*Windows Explorer*). Izkazalo se je, da lahko uporabnikom, ki nikdar prej niso delali z VCS (pisci dokumentacije, prevajalci, lektorji ...), na kratkem enournem tečaju razložimo nov način dela in jih usposobimo za uporabo *TortoiseSVN*.



FAZE PREHODA NA NOV KONCEPT PRIROČNIKOV

Pred začetkom vseh aktivnosti smo sprejeli odločitev o izboru priročnika, ki ga bomo najprej prenesli v novi koncept in obliko. To je bil priročnik COBISS3/Izposoja. Modul za izposajo ima namreč zelo širok nabor funkcionalnosti in ga uporablja veliko število uporabnikov. Večina izmed njih se ne udeleži namenskega tečaja, zato so vprašanja o njegovi rabi pogosta.

V **1. fazi** je ožja delovna skupina v praksi preučila in spoznala uporabo posameznih orodij, način dela in sintaktična pravila označevalnega jezika Markdown, preučila variante v MD za zahteve, ki jih mora priročnik izpolnjevati (označevanje naslovov, alinej, tabel, vključevanje slik, dodajanje hiperpovezav ...). Rezultat dela je bilo nekaj poglavij priročnika, pripravljenih na nov način. Te in sam način dela smo predstavili krogu sodelavcev, ki se ukvarjajo s priročniki za C3.

V **2. fazi** smo izvedli izobraževanje sodelujočih za uporabo posameznih orodij (Visual Studio Code, TortoiseSVN, ukazne skripte), uporabo jezika Markdown, način vnosa vsebin v MD in prilagoditev vsebine novemu konceptu.

Glavna novost pri pisanju vsebine je bila, da za potrebe kontekstne pomoči pri opisu posameznega postopka izhajamo neposredno iz opisa metode, ki jo želi uporabnik izvesti v C3. Po drugi strani smo morali upoštevati, da bo uporabnikom na razpolago tudi priročnik v celoti, s kazalom ter z logičnim zaporedjem opisov postopkov. Pri prejšnjih priročnikih seveda ni bilo pomembno, kje v programu se uporabnik nahaja, ko želi uporabiti priročnik, saj je navigacijo po priročniku predstavljalo njegovo kazalo in ne mesto v programu. Ta sprememba je predstavljala bistveno spremembo miselnosti, kako naj bo vsebina priročnika napisana. Posledično prenos vsebine iz obstoječega priročnika ni bil mogoč s preprostim kopiranjem delov besedila, temveč je bilo vsebino treba prilagoditi. Praviloma smo dodali uvodni opis posamezne operacije in seznam vsebin, ki so v pripadajočem poglavju opisane, v obliki hiperpovezav na te opise. Pogosto je bilo treba na novo strukturirati celotno poglavje.

2. faza se je zaključila s srečanjem vseh sodelujočih. Predstavili smo izkušnje pri delu, prednosti in tudi pomanjkljivosti. Na srečanju smo potrdili novi koncept in se dogovorili o pripravi celotnega priročnika COBISS3/Izposoja na tak način.

3. **faza** je obsegala pripravo vsebine priročnika. Na rednih tedenskih sestankih smo razreševali razna vprašanja in dileme (uporaba orodij, vsebinska vprašanja ...). Ta faza je bila najobsežnejša. Iz tedna v teden smo ugotavljali prednosti novega pristopa, saj smo nenehno nabirali izkušnje in znanje glede rabe orodij ter nove "logike" priprave priročnika. Rezultati dela so bili iz tedna v teden bolj vidni, kar je pomenilo tudi dodatno motivacijo za vse udeležene.

VKLJUČITEV KONTEKSTNE POMOČI V COBISS3

Med pripravo vsebine priročnika smo v razvoju ustrezno dopolnjevali C3. Odločili smo se za dva nivoja kontekstne pomoči:

- Če uporabnik v posameznem oknu pritisne tipko [F1], bo dobil vsebinske informacije o poljih/podatkih v tistem oknu in o postopku vnosa zanje.
- Če pritisne kombinacijo tipk [Shift+F1], pa dobi informacije o vrsti okna (npr. iskalnik, urejevalnik, zagon obdelave, izbira izpisov, predogled ...) ter obnašanju elementov v tovrstnem oknu – skratka opis mehanike delovanja, ki ni vezana na vsebino podatkov v oknu. Te informacije so namenjene predvsem novim uporabnikom, ki se v C3 še ne znajdejo povsem.

Glavno težavo je predstavljala identifikacija točke, kjer se je uporabnik "nahajal" ob pritisku na [F1] in preslikava tega v URL, ki bo v brskalniku skočil na pravo mesto v priročniku.

Večina namenskih orodij za kontekstno pomoč zahteva, da za vsako enoto pomoči (ang. *help item*) določimo ključ (ang. *context ID*), ki ga orodje potrebuje, da najde in prikaže pravo enoto. Včasih so ti ključi numerični, kar pomeni, da je njihovo dodeljevanje programskim modulom in preprečevanje podvajanja, dokaj zahteven postopek, zelo občutljiv na napake. Kadar je tak ključ poljubni niz znakov, je naloga precej lažja, še vedno pa zahteva vnaprej določeno strategijo njihovega dodeljevanja. Toda C3 ima na stotine prepletenih modulov, oken in obdelav, kjer se isto okno lahko prikaže v povsem drugem kontekstu, zato ne pride v poštev, da bi vsakemu oknu dodali zgolj preprost klic metode *nastaviKljucZaPomoc("xyz")* ali kaj podobnega.

Po drugi strani je Java programski jezik, v katerem je mogoče vedno dobiti informacije o razredih, metodah in drugem (ang. *Run-time type information – RTTI*). Zato smo dodali globalno strukturo (seznam), kamor na vsaki pomembni točki postopka, ki ga sproži uporabnik (npr. izbira opcije v meniju, odpiranje okna za potrditev, odpiranje iskalnika, odpiranje vnosnega okna itd.) dodamo ime objekta, ki je na tisti točki aktiven.

Vsako okno, kjer lahko uporabnik pritisne [F1], zdaj potrebuje le še deklaracijo metode za prestržanje tipk (ang. *keyboard event*), ki bo poklicala modul za prikaz kontekstne pomoči. Ta pogleda v zgoraj opisani seznam, preveri, ali je bila pritisnjena tudi tipka [Shift], in z nekaj interne logike iz teh informacij določi ključ za kontekstno pomoč. S spletnega strežnika potegne tekstovno (.ini) datoteko, kjer so navedene preslikave med tem ključem in ciljnim URL. Primeri takšnih preslikav so:

```
CirPatronHold.finder=loan/loan.html#UnavailableMaterial  
CirOverdue.UOPCreateAllOverdues.operation=loan/loan.html#OverdueNotices_creating  
CirCalendar.editor=loan/loan.html#Calendar
```

Levi del (pred enačajem) je kontekstni ključ, desno je končnica URL. Program pred končnico doda le še ime strežnika in oznako jezika ter dobi URL, ki ga posreduje spletnemu brskalniku.

FAZA MAPIRANJA POVEZAV MED VSEBINO PRIROČNIKA IN COBISS3

Datoteko s preslikavami urejajo dokumentalisti sami in so pri tem neodvisni od programerjev. Ker seveda ne morejo poznati internih nazivov vseh objektov v programu, smo zanje dodali še posebno kombinacijo tipk, ki se proži na istem mestu kot [F1] in dokumentalistu posreduje kontekstni ključ ter druge potrebne podatke.

Postopek vključevanja povezav je potekal tako, da smo v C3 testno izvedli vsak opisani postopek, ob tem preverili, kako je v programu definiran ključ kontekstne pomoči, v priročniku poiskali opis tega postopka ter definirali povezavo med kontekstnim ključem in vsebino priročnika. Nazadnje smo preverili pravilnost delovanja in ustreznost izpisane pomoči.

Pri tem smo ugotavljali, da bo treba prilagoditi tako delovanje kontekstne pomoči v C3 kot tudi vsebino priročnika. Pri nekaterih postopkih uporaba kontekstne pomoči ni delovala – gre za posamezne postopke, ki so v programu izvedeni na specifičen način (npr. posebni urejevalniki). Tudi pri vsebini priročnikov smo marsikdaj ugotovili, da bodo potrebne prilagoditve (npr. za izvajanje nekaterih postopkov je bila pomoč opisana razdrobljeno na več mestih v priročniku, nekatere funkcionalnosti v priročniku niso bile opisane, ker izvedba ni zahtevala posebnih pojasnil, postopki niso bili vedno opisani po korakih, opisi so bili predolgi in prepredeni z opozorili in posebnostmi ...).

Postopoma smo odpravili tudi te pomanjkljivosti in sčasoma prišli do točke, ko je pomoč za vse postopke v modulu izposoje delovala in vsebinsko ustrezala. Priročnik v obliki delovne verzije je bil zrel, da ga ponudimo uporabnikom.

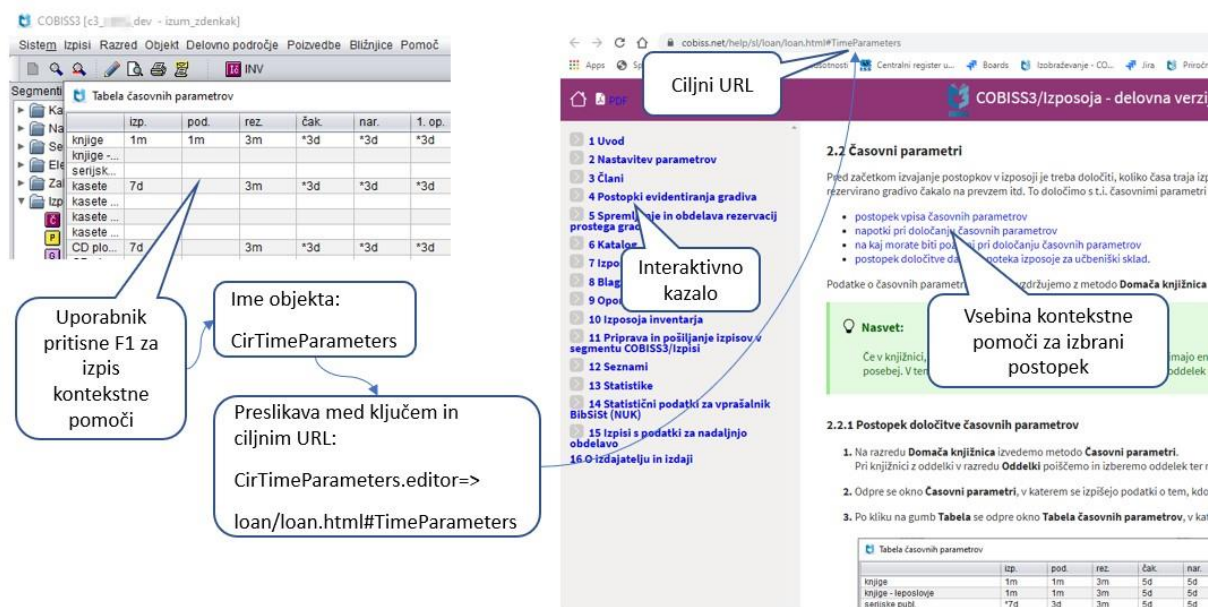
REZULTATI

Uskladitev spletne strani s priročniki s preostalimi spletnimi stranmi

Sledila je vključitev oblikovalcev domačih strani. Do tega trenutka smo uporabljali zgolj začasno oblikovanje, ki je povsem zadoščalo internim potrebam. Zdaj smo se dogovorili o videzu domače strani, posameznih delov priročnikov (npr. primerov, opozoril), kazal in drugih možnosti, ki naj jih domača stran ponuja. Uskladiti smo jo želeli z obstoječimi stranmi, vendar upoštevati, da morajo biti barvna shema in pisave prijazne uporabnikovim očem, očiščene nepotrebnega balasta in jo bodo lahko uporabljali tudi slabovidni in slepi s pomočjo bralnikov zaslona.

Izdaja nove verzije C3 z vključeno kontekstno pomočjo

Novo verzijo C3 in priročnikov smo po zaključku testiranja celotne funkcionalnosti v testnem okolju izdali v produkcijo. Objavili smo novo domačo stran za priročnike in podstran za priročnik COBISS3/Izposoja [5].



Slika 2: Primer odpiranja kontekstne pomoči v C3

AKTIVNOSTI PO IZDAJI PRIROČNIKA

Ob izdaji priročnika v produkcijo je treba v VCS označiti (ang. *tag*) vse dokumente v njem, da se ve, katera njihova verzija je bila v trenutku izdaje aktualna in katere datoteke so spremenjene naknadno.

Priročnik COBISS3/Izposoja je le en od desetih priročnikov za področje lokalnih aplikacij. V postopek priprave preostalih priročnikov so vključeni še drugi sodelavci, zato je sledilo pisanje navodil, ki obsegajo popis orodij, dogovore in pravila pri pripravi vsebine v MD, označevanje besedil za lekturo, predaja v lektoriranje, opis urejanja vsebine datoteke s preslikavo povezav

med COBISS3 in priročnikom, opis postopkov pri nameščanju priročnikov na testni in produkcijski strežnik ter vsi preostali postopki, povezani s priročnikom.

ZAKLJUČKI

Verzijo programske opreme C3, ki podpira kontekstno pomoč, in novi priročnik smo izdali 12. junija 2021. A napredek ni bil le v tem, kar je vidno na zunaj. Izbrani koncept in orodja omogočajo piscem priročnikov, da se posvetijo pripravi vsebine, oblikovanje pa prepustijo oblikovalcem. Posebej pozitivno se je izkazalo delo z VCS.

Pripravljena podpora v obliki skript za oblikovanje priročnikov v formatih HTML in PDF in samodejni prenos na testni ali produkcijski strežnik pomenita, da lahko vse postopke z minimalno podporo izvedejo avtorji vsebine priročnikov sami. To omogoča, da je v proces priprave priročnikov vključeno manj ljudi, kar ves proces poenostavlja in racionalizira.

Ker zaposleni v razvoju pri svojem delu prav tako uporabljamo te priročnike, smo tudi sami zadovoljni, da je dostop do pomoči hitrejši in preprostejši. Seveda pa je priročnik v prvi vrsti namenjen končnim uporabnikom naše programske opreme. Zato je pomembno, da jim kontekstno pomoč približamo, jo ustrezno predstavimo in jih navadimo na njeno uporabo. V uporabniškem servisu so uporabnike že začeli seznanjati z njo. Predstavili jo bomo tudi v procesih izobraževanja uporabnikov.

Čeprav je bil C3 z novim priročnikom izdan na začetku počitnic, ki pomenijo počitniški način dela tudi za knjižničarje, smo v tem času zabeležili približno 1000 dostopov do priročnika za izposajo, od tega četrtno za format PDF. Sodelavci uporabniškega servisa ugotavljajo, da kontekstna pomoč tudi njim olajša svetovanje uporabnikom, saj je dostop do nje zelo preprost, priročnik je lažje berljiv in bolj pregleden, predvsem pa so koristne hiperpovezave.

Priprava priročnika COBISS3/Izposoja in vključitev kontekstne pomoči za postopke v tem modulu sta bila prva koraka. Nadaljujemo delo s pripravo priročnika s splošnimi navodili za delo v C3, vključitvijo kontekstne pomoči za te postopke, pripravo opisa novosti ob izdajah (ang. *release notes*) v formatu MD in nadaljnje izboljšave strani s priročniki.

VIRI IN LITERATURA

- [1] Apache Software Foundation. *Subversion*. 2021. Dostopno na: <https://subversion.apache.org/> [25.08.2021]
- [2] A Matt Cone project. *Markdown Guide*. 2021. Dostopno na: <https://www.markdownguide.org/> [25.08.2021]
- [3] *GitHub*. 2021. Dostopno na: <https://github.com> [25.08.2021]
- [4] Heath, Catherine, 2019. *What is Contextual Help, and how it can lead to happier users*. Dostopno na: <https://www.knowledgeowl.com/home/contextual-help> [25.08.2021]
- [5] IZUM – Institut informacijskih znanosti. *COBISS3/Izposoja – delovna verzija*. 2021. Dostopno na: <https://www.cobiss.net/help/sl/loan/loan.html> [25.08.2021]
- [6] IZUM – Institut informacijskih znanosti. *Programska oprema COBISS3*. 2021. Dostopno na: <https://www.cobiss.si/c3/> [25.08.2021]
- [7] Microsoft. 2021. *Visual Studio Code*. Dostopno na: <https://code.visualstudio.com/> [25.08.2021]
- [8] *Pandoc a universal document converter*. 2020. Dostopno na: <https://www.pandoc.org/> [25.08.2021]
- [9] *The LaTeX Project*. 2021. Dostopno na: <https://www.latex-project.org/> [25.08.2021]

- [10] *TortoiseSVN the coolest interface fo (Sub)version control*. 2021. Dostopno na: <https://tortoisesvn.net/> [25.08.2021]
- [11] *TortoiseSVN-dev*. 2021. Dostopno na: <https://groups.google.com/g/tortoisesvn-dev> [25.08.2021]
- [12] Tsai, Michael, 1998. *WYSIWYG: Is it What You Want?*. Dostopno na: <http://www.atpm.com/4.12/page7.shtml> [25.08.2021]
- [13] *WKHTMLTOPDF*. 2021. Dostopno na: <https://wkhtmltopdf.org/> [25.08.2021]