

PREDRAG CVETKOVIĆ

## NORMA KAO PROGRAMSKI KOD: UVODNA RAZMATRANJA

*Programiranje pravnih pravila je proces pretvaranja pravne norme u neki od programskih jezika. Pravno pravilo koje postoji i u formi teksta i u formi kompjuterskog koda/programa naziva se pravilom u formatu „Norma kao programski kod“ (NKK). Optimalni cilj NKK metodologije je da norma u pisanoj formi i njena verzija u kodu imaju isto značenje. Domet i značaj NKK metodologije prepoznata je i u pravnom prostoru EU: postoje i sistemski dokumenti koje su pojedine države usvojile kao vodič za njenu primenu. Uspeh navedenog procesa prati se na tri načina: kroz uvođenje „kontrolora“, sproveden je testiranje i primenu metoda regulatornog igrališta. Implementacija NKK metodologije zahteva multidisciplinarni i multisektorski pristup (učesće javnog i privatnog sektora): takav pristup je neophodan s obzirom na dinamiku i stalne promene u razvoju tehnologije koja prati opisani proces. Uloga pravne ekspertize u ovom slučaju je paralelna onoj koju imaju tehnološki eksperti: te uloge nisu isključive, već bi trebalo da imaju sinergijski efekat.*

*Ključne reči: norma kao kod, pravo i programiranje, predlog Uredbe EU o veštačkoj inteligenciji, regulatorno igralište*

### U V O D

Pravne norme su mogući predmet programiranja različitim fazama (definiisanje, sprovođenje, praćenje, kontrola, tumačenje). Programiranje pravnih pravila je proces pretvaranja pravne norme u neki od programskih jezika.<sup>1</sup>

---

Prof. dr Predrag Cvetković, redovni profesor Pravnog fakulteta Univerziteta u Nišu, e-mail: [pepi@prafak.ni.ac.rs](mailto:pepi@prafak.ni.ac.rs). Rad je nastao kao rezultat finansiranja od strane Ministarstva prosvete, nauke i tehnološkog razvoja RS prema ugovoru evidencioni broj 451-03-68/2021-14/200120.

<sup>1</sup> Programiranje, pored poznavanja programskog jezika u pitanju, uključuje primenu logike, korišćenje matematike i druge metode koje idu izvan prostog učenja programskog jezika. Međufaza

Pravno pravilo koje postoji i u formi teksta i u formi kompjuterskog koda/programa za potrebe ovog rada naziva se pravilom u formatu „Norma kao programski kod“ (dalje i: NKK). Postoji mogućnost automatizacije određenih faza pravnog tumačenja pomoću računarskog koda, kao i usvajanje prakse testiranja softvera kako bi se osigurao kvalitet pravne norme kroz testove nacrti i predloga pravnih tekstova. Optimalni cilj NKK metodologije jeste da se obezbedi da tekstualne formulacije budu prevedene u formu razumljivu programerima: uz to, trebalo bi da postoji mogućnost da pravnici bez znanja programiranja koriguju pravila na način koji ne utiče na funkcionisanje koda.<sup>2</sup> Konačni efekat je da norma u pisanoj formi i programski kod imaju isto značenje: verzija pravne norme u formi koda se smatra autentičnom i sa pravnom snagom jednakom onoj koju ima verzija u pisanom tekstu. Funkcionalna jednakost teksta i programskog koda naziva se izomorfizam.<sup>3</sup>

ovog procesa jeste algoritmiranje: izraz podrazumeva prevođenje ugovora u format koji je razumljiv za programere koji se bave pretvaranjem pravne norme u kod. Načelno se razlikuju tri pristupa algoritmiranju: korišćenje pravila formalne logike, korišćenje „dijagrama toka“ i pseudokoda. V. više u: Predrag Cvetković, „Ugovor kao algoritam: uvodna razmatranja“, *Zbornik radova Pravnog fakulteta u Nišu*, br. 92, Niš, 2021, 15–34.

<sup>2</sup> V. infra deo rada „Kontrolor primene pravila u formi programskog koda“.

<sup>3</sup> Zastupnici izjednačavanja programskog koda i prava opisanu činjenicu uzimaju kao argument za novo razumevanje pojma pravne regulative: ono je moguće iskorakom izvan tradicionalnih shvatanja prava i njegovih elemenata. Nosilac novog razumevanja pravnog regulisanja je „novi“ regulator: programski kod. V. Lawrence Lessig, *Code: Version 2.0*, Basic Books, New York, 2006, 5. Primer uticaja gorenavedenog stava na shvatanje pravne norme je uvođenje pojma digisprudencije u vokabular prava. Digisprudencija je pojam koji definiše uslove normativnosti programskog koda. Normativni značaj (i legitimnost) koda trebalo bi da se osigura već u momentu nastanka pravne norme: to znači da bi pravnici u fazi definisanja sadržine pravnog pravila trebalo da kao polaznu pretpostavku imaju da će to pravilo biti predmet digitalizacije, algoritmiranja i kodiranja. Gore navedeno je posledica funkcionisanja pravne norme u digitalnom okruženju. Transformativno dejstvo ove činjenice na pravnu normu može se predstaviti na primeru ugovora. Ugovor nastaje uz pretpostavku upliva tehnologije u nastanak, funkcionisanje i kontrolu ugovornih odredbi. Ugovor u tradicionalnom („analognom“) smislu je ispunjen kada su izvršene obaveze njime predviđene. Stoga je cilj programiranja ugovora da automatizuje izvršenje predviđenih obaveza: cilj nije analiza teksta ili interpretacija ugovornih normi. Instrukcije programerima moraju da budu precizne i nedvosmislene. Automatizuje se izvršenje određenih zadataka sadržanih u ugovornim obavezama (npr. plaćanje, isporuka, slanje obaveštenja o prijemu itd.). Programiranje ugovora ima za rezultat automatizaciju izvršenja: ova automatizacija praćena je garancijom da će izvršenje u pitanju biti učinjeno upravo kako je ugovorom usaglašeno. Za razliku od tradicionalnog („analognog“) ugovora, koji razlikuje tekst ugovora i njegovo izvršenje, programski kod dopušta da tekst kontrakta u programskom jeziku bude istovremeno i akcija njegovog ispunjenja. Time se potencijalno izjednačava ugovorno regulisanje i ispunjenje ugovora: samim postankom koda (normiranjem) ono što bi trebalo da bude učinjeno kao rezultat ugovora već je izvršeno s obzirom na nepromenljivost programskog koda i njegovu

Problemi integrisanja prava u kod jesu višeslojni.

Prvo, programeri po pravilu nemaju pravničko znanje.

Drugo, razlikuje se dinamika analize programera i pravnika. Programer primenjuje *a priori* znanje o programskim jezicima i algoritmima: ovo je znanje objektivizirano programskim jezikom nezavisnim od iskustva. Sintaksa i semantika programskih jezika su striktni i ne dozvoljava se fleksibilnost formulacija. Pravna norma se, pak, primenjuje uz *a posteriori* znanje (korišćenje iskustva, empirijske dokaze i rezultate opservacija).

Sa druge strane, pravno programiranje kao integrisanje perspektiva prava i programiranja kako bi se obezbedili zajednički obrasci funkcionisanja pravnog i programskih jezika, omogućavaju dve činjenice.

Prvo, zajednički imenitelj pravnog i programskog jezika jeste struktura zaključivanja: i jedan i drugi jezik bazirani su na pravilima čijom primenom se dolazi do određenih obrazloženih stavova čiji nastanak može da se rekonstruiše. Programski jezik prati striktna pravila programiranja; jezik pravne proze, pak, zasnovan je na jezičkim principima prirodnog jezika u funkciji formulisanja pravnog pravila.

Drugo, sličnost pravnog pravila i programiranja je u tome što njihov proces nastanka i norme i koda ima definisanu strukturu i pravila kontrole. Programer, kao i pravnik, procesuirao određene podatke postupajući u skladu sa zadatim pravilima strukture i formata.<sup>4</sup>

---

zatvorenost za intervenciju posrednika. Ugovorni dokument u formi koda više nije otelotvorenje sporazuma već sredstvo njegovog ispunjenja. V. Lawrence Edward Diver, *Digisprudence: the affordance of legitimacy in code-as-law*, University of Edinburgh, 2019, doctoral thesis, 304. <https://era.ed.ac.uk/bitstream/handle/1842/36567/Diver2019.pdf?sequence=1>, 21. 2. 2021. Gorenaveđena tvrdnja je relevantna pre svega za pametne ugovore („smart contracts“). O pametnim ugovorima v. više u: Predrag Cvetković, „Liability in the context of blockchain-smart contract nexus: Introductory considerations“, *Collection Papers Fac. L. Nis*, No. 89, 2020, 83.

<sup>4</sup> Ima i drugih sličnosti koje mogu da se prime. Principi modularnosti, kompozicionalnosti i ekstenzibilnosti. *Modularnost* podrazumeva da je sistem podeljen u subsisteme (module) koji su povezani međusobno na način koji u određenim okolnostima omogućava njihovo odvajanje i korišćenje u drugim sistemima (programima). Teorija modularnosti je zasnovana i nastala u kibernetici. Vremenom je postala centralni princip dizajna računarskih sistema. V. Carlis Y. Baldwin, Kim B. Clark, *Design rules: The power of modularity*, Vol. 1, MIT press, 2000. U pravnoj teoriji ima radova kojima se analizira modularnost pravnih dokumenata. U smislu pravne tehnike, modularnost se javlja kod „upućivanja“ (gde se određene definicije koriste za determinisanje sadržine ostalih odredbi: npr. definicija pismena koja je integralni deo ugovornih odredbi kojima se definiše zaključenje ugovora; u programiranju postoji definisanje sadržine određenog programskog elementa u zavisnosti od promenljive – varijable). Modularnost je prisutna i u sistemskom zakonodavnom okviru: ona je otelotvorena u pojmu „incorporation by reference“ kojim se odredbe jednog zakona pozivanjem uključuju u indirektnu sadržinu drugog zakona. Koncept modularnosti je istraživao i u polju ugovornog prava.

## PRAVILA KOJA SU PODOBNA ZA KODIRANJE

Nisu sva pravila jednako podobna za definisanje u formi programskog koda.

Standardizacija i ponavljanje procesa (replikacija), uz masovnu upotrebu i nedvosmislenost (što smanjuje potrebu za dodatnom interpretacijom od strane drugih organa i sudova) karakteristike su pravila koja mogu da budu prevedena u NKK formi. Uz to, korišćenje NKK formata može da identifikuje kontradiktornosti u zakonodavnom aktu ili u odnosu određenog propisa sa drugim propisom iz okvira određenog pravnog sistema.

Naredbodavna pravila, budući nedvosmislena i bez tereta dodatnih tumačenja (reč je o pravilima u binarnoj formi: ponašanje je ili dozvoljeno ili nije; određena situacija je ili pravno dozvoljena ili je zabranjena) posebno su podobna da budu predmet programiranja. Efekti i posledice primene (odnosno neprimene) ovih pravila su merljivi. Uz to, programski kod funkcioniše u binarnoj formi (ukoliko je ispunjen određen uslov kod se izvršava; ukoliko nije, kod se zaustavlja). Ovo ne znači da je NKK format neprimenljiv za druge tipove pravnih pravila: radi se samo o činjenici da je pretvaranje norme u kod *de lege lata* moguće u oblastima u kojima takva pravila preovlađuju. Proces prihvatanja NKK-a može da uzrokuje da legislatori, koriste u većoj meri naredbodavne norme kako bi obezbedili primenu NKK formata na propis u pitanju.<sup>5</sup>

Modularni uslovi i grupe uslova mogu da se koriste za ograničenje toka informacija između različitih segmenata ugovora, ograničavajući time njegovu kompleksnost. V. Jane M. Radin, „Boilerplate Today: The Rise of Modularity and the Waning of Consent“, *Mich. L. Rev.*, 2005, 104: 1223. *Kompozicionalnost* je princip koji značenje složenog izraza definiše kroz značenje elemenata koji ga čine i pravilima koji definišu njihovu vezu: *ex ante* je moguće odrediti sve elemente jednog izraza, njihov odnos, značenje i rezultat (normu koju ovi elementi tvore). Na drugom polu kompozicionalnosti je princip kontekstualnosti: kontekst je okvir koji „okružuje“ činjenično stanje u pitanju i relevantan je za interpretaciju. Primer kontekstualnosti u pravu je korišćenje standarda, (npr. standard razumnog lica): o sadržini standarda nadležna tela odlučuju *ex post facto* prema okolnostima konkretnog slučaja. Sadašnji nivo razvoja veštačke inteligencije (pa samim tim i njena primena na pravni diskurs) zasnovan je na principu kompozicionalnosti; stoga je upitno da li veštačka inteligencija i programiranje mogu da se primene na pravne norme koji nose kontekst kao izvor interpretacije. *Ekstenzibilnost* je načelo koje omogućava da se računarskom programu dodaju nove funkcionalnosti ili mogućnosti. Ovaj princip ekstenzibilnosti je analogan mogućnosti aneksiranja ugovora odnosno noveliranja zakona.

<sup>5</sup> Takođe, za očekivati je da se predmet kodiranja budu pravila čiji se format upotrebe ponavlja (replicira), a ne pravila koja se koriste jednokratno ili samo ograničeno vreme. Na primer, uspostavljanje statutarnog organa u preduzeću je pravilo koje bi se retko primenjivalo: nije beznačajno ni to što je reč o kompleksnoj proceduri koja je pod uticajem različitih varijabla (vrsta preduzeća, vrsta članova, struktura odlučivanja), koja zahteva korišćenje značajnih programerskih resursa. U ovom i sličnim slučajevima, ulaganje vremena i resursa verovatno neće nadmašiti koristi od primene NKK metodologije.

Postoje i sistemski dokumenti koje su pojedine države usvojile kao vodič za primenu metodologije usaglašavanja norme i koda (npr. dokument „Better Rules for Government Discovery Report“ čiji je autor institucija nazvana „The Service Innovation Lab“ koja radi u okviru vlade Novog Zelanda). Ovaj dokument upućuje na sledeće odredbe koje su podobne za transformaciju u format programskog koda:

- pravila koja sadrže određene jednačine i formule (npr. pravila za obračun poreza);
- pravila o utvrđivanju (ne)postojanja određenog prava i njegovih konstitutivnih obeležja (radi se o pravima koja se ustanovljavaju popunjavanjem određenih formula za prijavu u cilju dobijanja određenih socijalnih beneficija);
- pravila kojima se uređuju standardizovani i iterativni/ponovljivi procesi;
- pravila koja uređuju uslove kao „korake“ u procesu s ciljem utvrđivanja njegove usklađenosti sa pravnim procesom;
- pravila koja se svojom strukturom odmah mogu implementirati u digitalnom obliku.<sup>6</sup>

U kontekstu prevoda pravne norme u kod, fokus akata EU je oblast javnih usluga (ovo je naglašeno i u predlogu Uredbe o veštačkoj inteligenciji EU).<sup>7</sup> Javne usluge (npr. prijavljivanje ili primanje socijalnih beneficija) i pravosuđe navedene su kao oblasti od interesa koje mogu imati koristi od korišćenja sistema veštačke inteligencije. Štaviše, sistemi veštačke inteligencije koji se „[...] koriste za određivanje da li takve beneficije i usluge treba da se uskrate, smanje, ukinu ili potraže od strane vlasti [...] smatraće se visokorizičnim kao i njihova upotreba“.<sup>8</sup> Ovi sistemi mogu imati značajan uticaj na egzistenciju ljudi. Njihova neadekvatna primena može da

---

<sup>6</sup> V. The Service Innovation Lab (LabPlus), „Better Rules for Government Discovery Report“, NZ Digital Government, March 2018, <https://www.digital.govt.nz/dmsdocument/95-better-rules-for-government-discovery-report/html#summary>, 20. 3. 2022. Navedeni izveštaj ima karakter seminalnog dokumenta kojim se na nivou jedne državne jurisdikcije obezbeđuju kriterijumi za procenu potencijala određenog pravila da bude predmet kodiranja. Ukoliko se na osnovu tih kriterijuma utvrdi opravdanost za kodiranje, onda se ta pretpostavka unosi kao premisa u procesu definisanja pravnog akta u pitanju. Digitalizacija pravnih akata se koristi kao metod podrške razvoju pravnih politika otelotvorenih u tekstovima zakona, te metodima kako će ti zakoni (pa samim tim i javne politike) da budu sprovedene u digitalnom okruženju.

<sup>7</sup> V. Proposal for a Regulation Of The European Parliament And Of The Council Laying Down Harmonised Rules On Artificial Intelligence (Artificial Intelligence Act) And Amending Certain Union Legislative Acts, COM/2021/206 final, <https://eur-lex.europa.eu/legal-content/EN/TXT/DOC/?uri=CELEX:52021PC0206&from=EN>, 1. 9. 2022 (dalje i: predlog Uredbe). Pregovori još uvek traju: finalizacija teksta očekuje se do polovine 2023. godine, a primena (uz razvoj i usvajanje tehničkih standarda krajem 2024. ili početkom 2025. godine.

<sup>8</sup> V. §37 Preamble predloga Uredbe.

naruši njihova osnovna prava, kao što su pravo na socijalnu zaštitu, nediskriminaciju, ljudsko dostojanstvo ili delotvoran pravni lek.

Kada je reč o ugovorima, potencijalni objekt kodiranja su odredbe koje su deljive u korake (sekvence) i čija primena daje rezultat u binarnoj formi. Ilustrativni primer klauzula podobnih za programiranje su odredbe o plaćanju: one mogu da se kodiraju tako što se funkcionisanje programa uslovi sadržinom objektivnih informacija iz računovodstvenog/knjigovodstvenog sistema: time se obezbeđuje nezavisni parametar ispunjenja obaveze plaćanja. Perfektno translatoвање ugovorne proze u kod je nemoguće zbog razlike u prirodi programskog i prirodnog jezika. Moguća je isključivo funkcionalna konverzija: konvertuje se ugovor kao forma zasvedočavanja sporazuma u programsku naredbu (algoritam) koja pokreće kompjuterske operacije usmerene na izvršenje ugovorne odredbe. Umesto pitanja da li linija koda odražava tekstualno značenje ugovorne odredbe, ključno je sledeće: da li izvršenje koda obezbeđuje rezultat koji su strane definisale ugovorom.

*Summa summarum*, da bi se omogućilo da norma u formi programskog koda bude izvršena, svi elementi odnosnog koda moraju da budu identifikovani, kao i njihovi odnosi: stoga norma koja se pretvara u kod (ili koja nastaje paralelno sa kodom) mora biti koherentna, nekontradiktorna i bez suvišnih izraza, podobna za opisivanje, identifikovanje i ponovno definisanje korišćenjem programskog koda. Pretvaranje norme u kod uslovljava da se dovoljno širok skup ulaznih informacija dobije u najranijoj mogućoj fazi postupka. Neke od njih će morati da obezbedi subjekat koji pokreće postupak pretvaranja norme u kod; drugi podaci mogu doći iz baza podataka u vlasništvu javne uprave (npr. zvaničnih registara iz kojih se podaci preuzimaju preko odgovarajućeg interfejsa). Obim i vrsta prikupljenih podataka odražavali bi znanja o datoj temi (opšte i specijalizovano), ciljeve uvođenja datog propisa i (ako je moguće) druge vrednosti koje je zakonodavac smatrao neophodnim za adekvatnu primenu. Kao rezultat toga, norma u formi programskog koda biće zasnovana na objektivnim, merljivim i dostupnim informacijama.<sup>9</sup>

<sup>9</sup> U kontekstu pravnog sistema, vertikalnu (hijerarhijsku) konzistentnost pravnih pravila moguće je očuvati i u kodu: tome doprinosi činjenica da programski jezici i sami imaju ugrađen princip usklađivanja programskih naredbi višeg i nižeg reda. Problem pretvaranja norme u kod nastaje kada je potrebno da se obezbedi upućivanje na pravne izvore izvan pravnog akta u kome norma funkcioniše, pri čemu izostaje hijerarhija u primeni (npr. nesaglasnost dva systemska zakona o istom pravnom pitanju). Izlaz može da bude tzv. mapiranje postojećih i mogućih funkcionalnih odnosa norme sa drugim izvorima i kontekstima. Ovo mapiranje je moguće kada se radi o pitanju ugovora s obzirom na jednostavniju strukturu ugovora u odnosu na propise. V. npr. Spencer Williams, „Contract Maps“, *UMKC Law Review*, 2022, [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4051985](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4051985), 2. 8. 2022.

## TUMAČENJE PRAVNE NORME U KONTEKSTU PROGRAMSKOG KODA

Čini se da je obezbeđivanje izomorfizma (funkcionalnog identiteta pravila u pisanoj formi i pravila u formi programskog koda) osnovni izazov koji stoji na putu procesa pretvaranja norme u kod. Tekst norme i njegova interpretacija služe ostvarivanju pojma pravde koji nije podložan egzaktnom definisanju: pravna proza je iz razloga ostvarivanja fleksibilnosti nekada dvosmislena, a nekada neodređena.

Programski kod služi ostvarivanju određenog, specifičnog cilja: on koristi ulazne vrednosti (inpute), transformiše ih i generiše rezultat (output) bez korišćenja spoljnih vrednosnih sistema (moralnih, etičkih itd.). Za razliku od prava koje koristi diskrecione i nejasne termine sa namerom obezbeđivanja fleksibilnosti, programski kod je nedvosmislen.<sup>10</sup>

Vreme formulisanja norme i vreme njene primene/tumačenja razlikuju se (npr. vreme usvajanja zakona i vreme njegove primene od strane, recimo, suda): u vremenskom periodu od usvajanja do primene, okolnosti na kojima počiva pravna norma (npr. zakon) mogu da se promene. Nasuprot ovoj karakteristici pravne norme, programski kod koji sadrži pravnu normu izvršava se odmah i ima neodložno pravno dejstvo: ne postoji tzv. interpretativni jaz, koji dozvoljava da se neadekvatnost norme proceni i zakon izmeni u cilju usaglašavanja norme sa nameravanim posledicama njene primene.

Važno je da u procesu pretvaranja pravila u kod postoji odgovarajući „osigurac“ koji bi definisao sistem upravljanja „kvalitetom“ programskog koda čime bi se sprečile štetne posledice njegove primene. Predlog Uredbe EU o veštačkoj inteligenciji predviđa da bi zvanična verzija programskog koda kojim je norma prevedena u programski jezik trebalo da bude javno dostupna (transparentna) kroz formu API protokola za programiranje aplikacija (reč je o sredstvu za programiranje aplikacija (Application Programming Interface) ili bilo kog drugog tehničkog sredstva koje omogućava pristup na daljinu (bez fizičke veze sa mestom programiranja)).<sup>11</sup> Regulatornim i nadzornim telima trebalo bi da se odobri potpuni pristup

---

<sup>10</sup> Ova karakteristika je refleksija principa kompozicionalnosti: v. više i supra u napomeni br. 4.

<sup>11</sup> V. čl. 64. st. 1. predloga Uredbe. Protokol API sastoji se od skupa uobičajenih postupaka, protokola i programskih alata koji se koriste u izradi softverskih aplikacija. Korišćenje API-ja omogućava programerima da koriste rad drugih programera štedeći vreme i trud koji je potreban da se napiše neki složeni program, pri čemu svi programeri koriste iste standarde. Umesto da se novi programi pišu novi iz temelja, nadograđuje se rad drugih programera.

skupovima podataka za testiranje koje je koristio kreator programskog koda.<sup>12</sup> Time se smanjuje nivo rizika primene koda (programske aplikacije) koji je nastao kao rezultat transformacije pravnog pravila u NKK format.

#### METODE PROVERE TAČNOSTI PRENOSA TEKSTA U PROGRAMSKI KOD

Primena pravila u formi programskog koda pretpostavlja sprovođenje određenih postupaka da bi se obezbedilo da rezultat primene takvog pravila bude identičan rezultatu koji ima primena pravnog pravila u pisanoj formi. Tri su moguća načina ostvarenja gorenavedenog cilja: ljudsko učešće u kontroli primene pravnog pravila (uvođenje „kontrolora“), testiranje i metod „regulatornog igrališta“.

##### *Kontrolor primene pravila u formi programskog koda*

Postojanje norme u formi programskog koda i teksta ne znači isključenje ljudskog učešća kao kontrolora ispravnosti konvertovanja pravne proze u programsku naredbu (algoritam). Rešenja koja podrazumevaju primenu zakona, uključujući i rešavanje sporova, primenom veštačke inteligencije su u fazi testiranja. Dok se ne dobiju zadovoljavajući rezultati, ne bi trebalo dozvoliti programskim naredbama da direktno opredeljuju pravni položaj subjekata zbog niskog procenta ispravnosti odluka i/ili etičkih razloga.

Uloga kontrolora jeste da uporedi rezultate primene jezičkih i sistemskih uputstava sadržanih u pisanoj verziji pravne norme sa rezultatima izvršavanja koda. Kontrolor je taj koji donosi konačnu odluku o validnosti pretvaranja normativnog izraza u NKK format. Ovo rešenje pretpostavlja odlučujuće učešće ljudskog bića u procesu: opisani koncept je sličan odnosu virtuelnih pomoćnika sudija zasnovanih na veštačkoj inteligenciji, sa jedne, i samih sudija, sa druge strane. Kontrolor se koncentriše na fazu tumačenja pravne norme: ova faza zahteva znanje koje u ovom trenutku (čak i ako postoje određene naznake njegovog ustanovljavanja) nije prisutno kao funkcionalni alat u formi veštačke inteligencije.

Opravdano je da se zaključi da je nivo potrebne ljudske intervencije u procesu pretvaranja norme u programski kod, određen koherentnošću, jasnoćom i prirodom pravila koje je predmet NKK formatiranja. Pravna norma koja je jasna i logički koherentna zahteva minimalnu ljudsku intervenciju prilikom pretvaranja

---

<sup>12</sup> O testiranju i drugim metodama provere tačnosti prenosa teksta u programski kod videti više infra u sledećem delu rada.



u format programskog koda. Nasuprot tome, pravilo koje se oslanja i zavisi od diskrecije u tumačenju, ima unutrašnju kontradikciju ili je nekonzistentno sa ostalim delom pravnog okvira u kome funkcioniše, nepodobno je za kodiranje.

Uloga kontrolora prepoznata je i u tekstu predloga Uredbe EU o veštačkoj inteligenciji. U predlogu Uredbe navedeno je da se u cilju upravljanja rizicima potencijalnih predrasuda, grešaka i netransparentnosti u pretvaranju pravila u kod, visokorizičnim sistemima veštačke inteligencije smatraju oni koji funkcionišu kao pomoćnici pravosudnim organima u istraživanju i interpretiranju činjenica i primene prava na te činjenice. Ova kvalifikacija ne bi trebalo da se odnosi na postojeće sprovođenje administrativnih postupaka koji ne utiču na sprovođenje postupaka u individualnim slučajevima, kao što su „anonimizacija ili pseudoanonimizacija sudskih odluka, dokumenata ili podataka, komunikacija osoblja, administrativni zadaci ili alokacija resursa rešavanja sporova“.<sup>13</sup>

### *Testiranje programskog koda*

Pravljenje test slučajeva je uobičajen način provere računarskih programa. Test slučaj je imaginarna ili stvarna lista ulaznih podataka. U kontekstu NKK formata, slučaj koji se testira ima strukturu pravnog akta: opisuje ulazne činjenice zajedno sa odlukom o tome šta je nameravani rezultat funkcionisanja programskog koda.<sup>14</sup>

Zajedno sa okvirom za proceduru obezbeđenja kvaliteta NKK-a, trebalo bi da se kreiraju skupovi test slučajeva kao ulazni podaci (činjenična stanja) Test slučajevi bi se postepeno proširivali situacijama iz stvarnog života. Vremenom bi se pojavio sveobuhvatan skup testova, koji bi (ako bi bili automatizovani) mogli da se izvode u razmerama i brzinom nedostižnoj za testiranje koje bi sproveli ljudi. Zadatak održavanja baze test slučajeva zahtevao bi posvećen tim ljudi koji bi periodično sprovodili preglede i ažuriranja. Bio bi neophodan i određeni izbor test slučajeva: u suprotnom obimne baze podataka (pri čemu svi elementni te baze nisu od značaja za proces testiranja) stvarale bi visoke troškove zbog potrebe njihovog adekvatnog (i često kompleksnog) održavanja.

---

<sup>13</sup> V. §. 40 Preambule predloga Uredbe.

<sup>14</sup> Testiranje NKK, kao i bilo kog računarskog koda, može povećati konzistentnost i smanjiti konflikte sa prethodnim propisima tako što se izvode testovi integracije i regresije. Test integracije se deli na fazu testiranja integracije komponenti i njihove međusobne interakcije i povezanost i test integracije koji se fokusira na povezanosti i komunikaciju između sistema, programskih paketa, integraciju sa spoljnim sistemima poput web-servisa). Testovi regresije obezbeđuju da novi elementi sistema nisu narušili kvalitet i strukturu prethodno ugrađenih komponenti.

Testiranje pravnog pravila u formi koda može da bude automatizovano do neke mere, baš kao i za svaki računarski kod. Već danas ima primera da sistemi testiranja sprovode statičku analizu koda u realnom vremenu, što omogućava otkrivanje, identifikaciju i ispravljanje grešaka u programu dok je još u fazi razvoja.<sup>15</sup>

Test slučaj pomaže da se identifikuju („mapiraju“) različiti scenariji koji mogu nastati kada se ulazni elementi unesu u program u cilju pretvaranja u naredbu programskog koda. Nedostatak testiranja je to što se njima omogućava pronalazanje grešaka, ali ne i njihovo odsustvo: činjenica da je primena norme u formi programskog koda dala rezultat koji je očekivan, ne znači da ne postoje okolnosti u kojima taj rezultat ne bi odgovarao očekivanjima pravnika koji su definisali normu. Potencijalno pogrešni rezultati testiranja su mnogobrojni. Nemoguće je da se program testira za svako moguće ponašanje, kao što ni norma ne može da bude formulisana na način koji obuhvata sve moguće slučajeve i modalitete pitanja koje uređuje. Ilustracije radi: u određenim državama postoje programi koji izračunavaju iznos poreza koje domaćinstvo dužuje u zavisnosti od njegovog prihoda. U Francuskoj, verzija ovog programa ima više hiljada ulaznih varijabla. Testiranje bi zahtevalo ne samo da se testira svaka varijabla (ulaz, input), već i da se ta varijabla testira u kombinaciji sa drugim ulaznim varijabilama. Uloga pravnika u definisanju test slučajeva uključuje znanje i pravničku intuiciju: kako nije moguće da se izradi test za svaki slučaj, moraju da anticipiraju koji su to slučajevi koji mogu da budu problem za pretvaranje u programski kod. U ovoj situaciji pravnici bi trebalo da primenjuju metod obrnutog inženjeringa, što nije znanje koje se uobičajeno stiče na u toku pravničkog obrazovanja.<sup>16</sup> Kako god bilo, testiranje norme kao koda u cilju obezbeđenja njene ispravne primene je metod koji nije potpuno pouzdan.

### *Regulatorno igralište*

Osiguranje adekvatnog rezultata pretvaranja norme u programski kod moguće je i sprovođenjem metoda „regulatornog igrališta“ („Regulatory sandbox“), Regulatorno igralište se definiše kao skup pravila koja omogućava inovatorima da testiraju svoj proizvod ili poslovni model u okruženju koje ih privremeno izuzima od poštovanja nekih ili svih zakonskih zahteva.

<sup>15</sup> Primer je program DeepCode. V. DeepCode tool: <https://www.deepcode.ai/>, 1. 9. 2022.

<sup>16</sup> *Obrnuti inženjering (reverse engineering)* jeste postupak otkrivanja principa uređaja, predmeta ili sistema putem temeljnih analiza njegove konstrukcije, funkcije i načina rada. Cilj je da se dođe do zaključaka o konstruktivnim odlukama koje su dovele do nastanka određenog fenomena (najčešće proizvoda, tehnologije ili softvera) uz malo ili nimalo predznanja o postupcima koji su upotrebljeni u originalnom procesu konstruisanja i proizvodnje.

Zauzvrat, ovi akteri su često obavezni da upravljaju svojim poslovnim modelom na ograničen način, na primer preko kontrolisanog broja klijenata ili izloženosti riziku, i pod strogim regulatornim nadzorom. Tehnika je osmišljena tako da bude obostrano korisna za regulatore i regulisane u cilju smanjenju pravne nesigurnosti za ove poslednje (one kojima je norma upućena). Regulator se nada da će kroz ovakvo testiranje stimulisati inovacije kroz eksperiment različitim varijantama pravnog okvira. Ovaj pristup uglavnom nalazi primenu u sektoru finansijskih tehnologija („FinTech“).<sup>17</sup> Ovaj primer ilustruje situaciju u kojoj promena tehnologije inicira i promenu regulative. Pristup je prvi put primenjen u Velikoj Britaniji 2015. godine, koja je 2016. odobrila ispitivanje efekata usluga iz oblasti FinTecha u okviru regulatornog igrališta. Inovatorima je omogućeno da testiraju nove tehnologije u jasno regulisanom okruženju pod bliskim nadzorom i tokom definisanog vremenskog perioda.<sup>18</sup>

Regulatorno igralište dolazi sa sopstvenim skupom prednosti i mana. Ono može da bude „crna kutija“ bez transparentnosti. Važno je i da ovaj metod bude neutralan u pogledu tehnologije.

Činjenica da akteri prelaze sa opšteg pravnog režima na selektivni i privilegovani režim koji je dostupan samo nekolicini takođe postavlja izazove u vezi sa ulaskom i izlaskom iz tog režima. Na primer, šta je sa potrošačima koji su koristili određenu uslugu pre nego što je firma ušla u „igralište“ misleći da će njihovi odnosi biti obuhvaćeni opšteprimenljivim režimom, ali onda to više nisu. Regulatorno igralište kao metod, ograničeno je na jednu jurisdikciju, što može stvoriti značajne poteškoće za one koji nude usluge na međunarodnom nivou (što je jedna od ključnih karakteristika digitalizacije).

Jasna prednost regulatornog igrališta jeste u tome što omogućava regulatorima da dobiju dodatni vremenski okvir da nastave da posmatraju i uče iz tehnologije i srodnih početnih poduhvata, dok u isto vreme podstiče inovacije pružanjem pravne sigurnosti i formalnih puteva za dijalog između regulatora i onih čija je aktivnost regulisana: ti su subjekti u obavezi da blisko sarađuju sa državnim organima i poštuju utvrđena pravila. Novi propisi se razvijaju empirijski, kao rezultat iskustva učesnika u „igri“ na regulatornom „igralištu“. Goreopisani model može da

---

<sup>17</sup> FinTech (skraćenica od „Financial Technology“), naziv je za nove tehnologije koje se koriste za automatizaciju svih aspekata finansijskih usluga koje se pružaju pojedincima i pravnim licima.

<sup>18</sup> Ovaj primer nije usamljen. Švajcarski nadzorni organ za finansijska tržišta ustanovio je novu kategoriju licenciranja za inovativne kompanije, koja uključuje zaštićeno okruženje bez licence. Kanadska uprava za hartije od vrednosti je takođe pokrenula regulatornu inicijativu, a Australija je nedavno otvorila istraživački centar za blokčejn tehnologiju i planira da otvori regulatorni sandbox koji bi omogućio firmama da testiraju proizvode.

se primeni u postupku kodiranja i testiranja propisa koji su u formatu programskog koda. Zamislivo je postojanje odvojenih regulatornih igrališta, formiranih u cilju testiranja pravila u NKK formatu čija je primena ograničena na određenu pravnu oblast (npr. regulativa finansijskog sektora) ili čak određenog zakona (npr. zakon o javnim nabavkama).

Metod „igrališta“ je važan za veštačku inteligenciju i NKK format posebno u svetlu primene metoda „norma kao programski kod“ u pitanjima koja su identifikovana kao rizična.<sup>19</sup> Veštačka inteligencija je tehnologija koja zahteva nove oblike regulatornog nadzora i siguran prostor za eksperimentisanje, istovremeno obezbeđujući odgovornu inovaciju i integraciju odgovarajućih zaštitnih mera i smanjenje rizika koji prate transformaciju norme u programski kod.

#### AKTERI IMPLEMENTACIJE PRETVARANJA PRAVNOG PRAVILA U PROGRAMSKI KOD

Važnu ulogu u implementaciji NKK formata potencijalno imaju akteri kako iz javnog, tako i iz privatnog sektora.

Napore u okviru javnog sektora trebalo bi da predvode zakonodavna tela i regulatorne institucije. Iz prirode NKK formata proizilazi zaključak da njegova implementacija podrazumeva multidisciplinarno znanje koje će na pravi način da optimizira inovativne prednosti pretvaranja pravnog pravila u programski kod: stoga je poželjno da metodologija bude primenjena od strane različitih tela (ministarstava, agencija) sa nepodudarnim, ali komplementarnim nadležnostima i kapacitetima u kontekstu primene NKK metodologije.

Uloga privatnog sektora ogleda se u mogućnosti da on, u odsustvu aktivnosti u javnom sektoru, sam definiše pravila u formatu NKK-a.<sup>20</sup> Ukoliko ovo bude slučaj, države bi trebalo da imaju u vidu postojanje rizika tzv. komercijalnog zaključavanja: da jedna kompanija svoj „privatni standard“ (platformu koju je razvila u cilju pretvaranja pravila u kod) iskoristi za sticanje položaja na tržištu koji je protivan propisima o zaštiti konkurencije. U odsustvu vladinih intervencija u ovoj oblasti, verovatno je da će tržište proizvesti sopstvena rešenja koja će pomoći implementaciji inicijativa sa karakterom NKK metodologije. Konture i intenzitet navedenog uključivanja zavisiće od pozicije svake države ponaosob. Valjano je rešenje

---

<sup>19</sup> V. supra napomenu 13 i prateći tekst.

<sup>20</sup> Situacija je gotovo analogna sa nastankom *lex mercatoria*: kao i u slučaju NKK metodologije, „čutanje“ države kao zakonodavca otvorilo je prostor da privatni subjekti generišu sopstvena pravila postupanja.

da države definišu minimalne zahteve i standarde za postupak i rezultat procesa u formi NKK-a kako bi očuvale integritet javnog interesa u ovoj oblasti, bez ograničenja i postavljanja okvira koji bi mogli da imaju za rezultat suboptimalna rešenja.

Prof. Dr. PREDRAG CVETKOVIĆ  
Full Professor, Faculty of Law  
University of Nis

## LAW AS PROGRAM CODE: INTRODUCTORY REMARKS

### Summary

Coding of legal rules is the process of converting a legal norm into one of the programming languages. A legal rule that exists in both text and computer code/program form is labeled as the „Law as Code“ (LAC). The optimal goal of the LAC methodology is to enable the meaning of the norm in written form to be preserved in its version in programming code. The scope and importance of the LAC have been recognized in the EU Law: there are also general documents adopted by individual states as a guide for its application. The success of the above-mentioned process is monitored in three ways: through the introduction of „controllers“, the implementation of testing and the application of the regulatory sandbox methodology. The implementation of the LAC methodology requires a multidisciplinary and multisectoral approach (public and private sector involvement: such an approach is necessary considering the dynamics and constant changes in the development of technology that accompanies the described process. The role of legal expertise in this case is parallel to that of technological experts: these roles are not exclusive, but should have a synergistic effect.

*Key words:* norm as code, law and programming, proposal for EU Regulation on artificial intelligence, regulatory sandbox

### Literatura

- Baldwin C. Y., Clark K. B. and Clark K. B., *Design rules: The power of modularity*, Vol. 1, MIT press, 2000.
- Cvetković P., „Ugovor kao algoritam: uvodna razmatranja“, *Zbornik radova Pravnog fakulteta u Nišu*, br. 92, Niš, 2021.
- Cvetkovic P., „Liability in the context of blockchain-smart contract nexus: Introductory considerations“, *Collection Papers Fac. L. Nis*, 89, 2020, 83.
- Diver L. E., *Digisprudence: the affordance of legitimacy in code-as-law*, University of Edinburgh, 2019, doctoral thesis, 304. <https://era.ed.ac.uk/bitstream/handle/1842/36567/Diver2019.pdf?sequence=1>.
- Lessig L., *Code: Version 2.0*, Basic Books, New York, 2006.
- Radin M. J., „Boilerplate Today: The Rise of Modularity and the Waning of Consent“, *Mich. L. Rev.*, 2005.

The Service Innovation Lab (LabPlus), „Better Rules for Government Discovery Report“, NZ Digital Government, March 2018, <https://www.digital.govt.nz/dmsdocument/95-better-rules-for-government-discovery-report/html#summary>.

Williams S., „Contract Maps“, *UMKC Law Review*, Vol. 91, 2022, [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4051985](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4051985).

Datumi prijema i prihvatanja rada

Primljen: 27.09.2022.

Prihvaćen: 18.10.2022.

PREGLEDNI RAD