

Domenski jezici i formati podataka za formulisanje rasporeda časova

Samir Ribić

Elektrotehnički fakultet u Sarajevu
Univerzitet u Sarajevu
Sarajevo, Bosna i Hercegovina
Samir.ribic@etf.unsa.ba

Sažetak—Raspored časova u školama je tema interesantna ne samo sa stanovišta rješavanja, nego i formulisanja. Obilje različitih vrsta ograničenja nad raspoređivanjem resursa koja se mogu javljati dovodi do potrebe za definisanjem posebnih jezika za njihovo opisivanje. U ovom preglednom radu dat je osvrt na devet specijalizovanih jezika kojima se mogu opisati zahtjevi za raspored časova u školama. Ukazano je na osnovne elemente sintakse, prednosti i nedostatke formata.

Ključne riječi—*timetabling, DSL, raspored;*

I. UVOD

Pored programskih jezika opšte namjene (C, C++, Java, Fortran, Lisp,...) u komunikaciji između čovjeka i računara koristi se široka klasa specijalizovanih jezika koji se zovu domenski specifični jezici. Ovi jezici ne moraju imati karakteristike koje imaju programski jezici opšte namjene poput uslova, potprograma, izraza i varijabli, nego su prilagođeni da bolje odgovaraju određenoj primjeni. Najpoznatiji primjeri takvih jezika su HTML i SQL. Oni nemaju čak ni primitivnu programsku petlju, ali su zato u svojim oblastima, iscertavanja web stranica ili pretraživanju relacionih baza podataka nezamjenjivi.

Automatsko generisanje rasporeda časova u školama je interesantan problem za istraživanje u računarskim naukama. Iako je problem u opštem slučaju NP complete, razvijeno je više heurističkih metoda za njegovo rješavanje: cjelobrojno linearno programiranje, simulirano hlađenje, genetski algoritam, kolonija mrava, kolonija pčela,... te kombinacije raznih metoda. Ovaj problem je zanimljiv i sa stanovišta njegove formulacije. Školski sistemi se međusobno dosta razlikuju. Pored uobičajene potrebe da se formuliše koliko časova sedmično se drži pojedini predmet u određenom razredu, različite škole mogu, ali i ne moraju imati zahtjeve za željenim i zabranjenim terminima nastavnika, opterećenjima prostorija i odjeljenja, distribuiranosti lokacija, istovremenim predavanjima u više odjeljenja itd. Kako je definisanje podataka u tekstualnim datotekama nekad praktičnije od unosnog grafičkog interfejsa (npr. zbog neopterećenosti interfejsa rijetko korištenim opcijama ili lakše copy/paste mogućnosti) i definisanje zahtjeva za raspored časova je dovelo do razvoja više specifičnih jezika za ovu svrhu.

U nastavku rada opisaće se vrste potreba koje se javljaju u programima za definisanje rasporeda časova i dati osvrt na formate podataka odnosno domenske jezike kojima se ti zahtjevi opisuju.

II. FORMULACIJA RASPOREDA ČASOVA

A. Elementi formulacije

Na prvi pogled, formulisanje zahtjeva za raspored časova je jednostavan zadatak, pogotovo u poređenju sa procesom njegovog rješavanja, npr. da je dovoljno navesti spisak odjeljenja, nastavnika, predmeta i fond časova u formi jedne matrice. Međutim, zbog različitosti škola, raspoloživosti resursa, ciljeva, rada u smjenama, preferencija termina, pitanje formulacije rasporeda je znatno složenije i treba da uključi definiciju raspoloživih resursa, vremenskih perioda, ograničenja i ciljeva.

B. Resursi

Resursi predstavljaju sve što je potrebno za održavanje časa u nekom od termina. Tri najvažnija resursa su odjeljenja ili drugi skupovi učenika/studenata kojima se drži nastava, nastavnici i prostorije. Zavisno od vrste školskog programa moguće je da se neki od ovih resursa ne moraju specificirati. Npr. sesije samostalnog učenja ne zahtijevaju nastavnike, sastanci nastavnika ne trebaju odjeljenja, a u nekim školama prostorije su automatski vezane za nastavnika ili odjeljenje pa se ne moraju navoditi. Pored ova tri osnovna resursa, moguće je zahtijevati i druge vrste resursa, poput projektora ili motornog vozila. Resurse je moguće grupisati, npr. svi nastavnici koji predaju isti predmet ili sve prostorije istog kapaciteta.

C. Vremenski periodi

Termin je osnovna jedinica raspoređivanja resursa u rasporedu časova. Za razliku od npr. raspoređivanja na željeznici, gdje je vrijeme kontinualno, rasporedi časova su diskretni. Resurs je zauzet u toku termina. U nekim slučajevima postoji i preklapanje između dva termina, npr. na nekim fakultetima se dešava da predavanje počinje u 10:00 i traje 45 minuta, a laboratorijske vježbe u 10:30 i traju 90 minuta bez pauze. Termini se dalje spajaju u veće vremenske

jedinice: smjene (opcionalne), dane, nastavne sedmice, i veće vremenske jedinice poput alternirajućih sedmica ili polugodišta. Pored ovoga, mogu se formulirati i manje očigledne grupe termina: rani ili kasni termini, termini koji pripadaju dva susjedna dana, termini bitni za kulturne ili vjerske potrebe, termini koji mogu biti usklađeni sa školskim autobusom itd.

D. Ograničenja

Ograničenja koja se mogu postaviti na dobijeni raspored časova su brojna i veoma različita u školama. Ograničenja se dijele na čvrsta, ona koja moraju biti zadovoljena da bi raspored bio korektan i meka, čije ispunjenje je poželjno za bolji kvalitet rasporeda. U radu [1] evidentirane su sljedeće vrste ograničenja:

- 1) Odjeljenja moraju biti raspoređena za potreban sedmični broj časova po predmetu.
- 2) Nastavnik mora biti raspoređen za potreban broj sastanaka sa svakim razredom.
- 3) Alocirani su posebni periodi za pauze.
- 4) Razred se može alocirati najviše jednom po terminu.
- 5) Soba se može alocirati najviše jednom po terminu.
- 6) Nastavnik se može alocirati najviše jednom po terminu.
- 7) Ako se škola nalazi u više odvojenih lokacija, obezbijediti vrijeme za putovanje.
- 8) Nastavnici mogu imati termine kada su nedostupni.
- 9) Sobe imaju kapacitet koji ne smije biti prevaziđen.
- 10) Sve sobe trebaju biti korištene u nekim školama.
- 11) Neki časovi se mogu držati samo u specijalizovanim sobama (kompjuterski kabinet, fiskulturna sala).
- 12) U školama sa više zgrada, svi časovi istog predmeta se trebaju držati u istoj zgradi
- 13) Sobe mogu imati termine kada su nedostupne.
- 14) Odjeljenja mogu imati termine kada su nedostupna.
- 15) Nastavničko opterećenje (dnevni broj časova) može imati limit ili minimum
- 16) Učeničko opterećenje (dnevni broj časova) može imati limit ili minimum koji može zavisi od uzrasta
- 17) Dnevni limit se može postaviti i na vrstu predmeta koje učenik može slušati
- 18) Postoje ograničenja na broj pauza, tj. termina kada se ne drži nastava, za odjeljenja.
- 19) Ograničenja na broj pauza se mogu postaviti i za nastavnike.
- 20) Moguće su različite vrste raspodjele časova nekog predmeta za određeno odjeljenje po sedmici, poput zahtjeva da se svi časovi tog predmeta drže u odvojenim danima, ili da se pravi razmak od po jednog dana, odnosno da se grupišu u isti dan.

21) Raspoređivanje nastave po predmetima može se definisati i za nastavnike.

22) Za pojedine časove mogu se definisati preferirani termini.

23) Nastavnici mogu imati svoje preferirane termine kada im više odgovara držanje nastave.

24) Za časove se mogu fiksirati termini kada se drže, ali i navoditi kada ne smiju biti raspoređeni.

25) Neki časovi se moraju održati nakon drugih.

26) Kod određenih časova potrebno je forsirati dvočase i tročase.

27) Odjeljenja se mogu dijeliti u grupe. Više odjeljenja može imati zajedničku nastavu, pa i više grupa iz različitih odjeljenja

28) Neki časovi se moraju održati istovremeno.

Pored ovoga uočene su i sljedeće vrste ograničenja:

29) Neki razredi ili grupe ne mogu imati istovremeno nastavu jer postoje učenici koji pripadaju u dvije grupe.

30) Prostorije mogu imati cijenu.

31) Razredi trebaju pripadati smjenama

32) Termini koji se djelimično preklapaju ne mogu se dodijeliti istom resursu

33) Postoje parovi nastavnika koji ne mogu biti istovremeno angažovani u nastavi.

34) Opterećenje može zavisi od vrste časa.

E. Ciljevi i evaluacija

Ispunjenje svih zahtjeva u rasporedu može biti izvodivo ili ne, a ako je izvodivo algoritam može, ali i ne mora naći jedno ili više rješenja u razumnom vremenu. Ako je nađeno više rješenja potrebno je odabrati ono koje se smatra najboljim. Ovdje je riječ o višekriterijalnoj optimizaciji. Ocjena rasporeda može biti bazirana na broju ispunjenih čvrstih i mekih ograničenja sa težinskim faktorima, broju pauza, ispunjenosti preferencija za terminima, brzini rješenja, cijeni dodijeljenih prostorija... Zahtjevi za raspored časova trebaju uključiti i izbor između ovih kriterija.

F. Rješenja i instance

Pojedini jezici za formulisanje rasporeda časova omogućavaju i čuvanje riješenog rasporeda radi kasnije evaluacije, kao i mogućnost evidentiranja više zahtjeva i rješenja rasporeda u jednoj datoteci.

III. LINIJSKI BAZIRANI FORMATI I JEZICI ZA FORMULISANJE RASPOREDA ČASOVA

A. ITC 2007

Za potrebe takmičenja u softveru za rješavanje rasporeda časova 2007 predložen je jednostavan format [2] koji se odlikuje jednostavnim parsiranjem (prilagođen tipičnim

naresbama za čitanje podataka u klasičnim programskim jezicima). Definisana su tri različita formata: za raspored ispita, raspored grupa po studentima i raspored časova po predmetima. Treći spomenuti format se bavi tematikom ovog članka. Kako se vidi u sljedećem primjeru u ovom formatu parametri svake od linija su međusobno razdvojeni razmacima. U sekciji za opis predmeta navodi se ime predmeta, nastavnika, broj časova u sedmici, broj sedmičnih sesija i broj učesnika na predmetu. U sekciji za opis rasporeda CURRICULA navode se pojedini predmeti koji se slušaju. Format je lagan za parsiranje i razumljiv, ali je pogodan samo za opis rasporeda jednostavnijih fakulteta, ne i osnovnih i srednjih škola, dok ne podržava opis nikakvih naprednijih ograničenja. Ako se isti predmet sluša u različitim odjeljenjima u odvojenim sesijama, mora se definisati kao poseban predmet za svaku sesiju.

```
Name: ToyExample
Courses: 4
Rooms: 2
Days: 5
Periods_per_day: 4
Curricula: 2
Constraints: 8
COURSES:
Phys Peric 3 2 42
Math Jovic 5 4 40
Chem Hadzic 5 4 18
ROOMS:
A 32
B 50
CURRICULA:
Cur1 2 Math Phys
Cur2 2 Phys Chem
UNAVAILABILITY_CONSTRAINTS:
Fizika 2 0
Fizika 2 1
Hemija 4 2
END
```

IV. JEZICI ZA OPIS RASPOREDA ČASOVA BAZIRANI NA PROGRAMSKIM JEZICIMA OPŠTE NAMJENE

A. TTLIB

U radu [3] je predložen format namijenjen više istraživačima algoritama za rješavanje rasporeda časova, nego krajnjim korisnicima takvih aplikacija. Jezik podržava klase sa nasljeđivanjem, funkcije, skupove, sekvence, cijele i realne brojeve. Sintaksno podsjeća na jezik Java, i pruža precizniji opis potrebnih ograničenja. Sljedeći primjer definiše klasu Lesson koja opisuje čas kao trojku (razred, prostorija, termin) uz dodatno ograničenje da prostorija koja se bira mora imati dovoljan kapacitet.

```
class Lesson
{
  Students : set of student;
  Room : room;
  TimeSlot : time;
  Function TestCapacity() : boolean;
}
Maths1 : Lesson;
```

```
Maths1[Students={Smith,Kovacs,Kowalsky}
, Room=?, TimeSlot=?];
function TestCapacity() : boolean;
{
  Room.Capacity >=Count(Students);
}.
```

Dobra osobina ovog domenskog jezika je u mogućnosti definisanja nestandardnih ograničenja. Mana mu je što je upotrebljiv samo korisnicima sa odgovarajućim predznanjem iz programiranja i u potrebi za ručnim definisanjem nekih očiglednih ograničenja, a samim tim i dosta velikim opisom realnih rasporeda.

B. STTL

STTL [4] je sličan TTLIB, po tome što poznaje klase sa nasljeđivanjem, skupove, cijele brojeve i druge tipove podataka. Namijenjen je za specificiranje šire klase problema, ne samo rasporeda časova. Format razlikuje opis problema, instance i rješenja koji se pišu u posebnim datotekama. Tako na primjer, u specifikaciji problema srednjoškolskog rasporeda nastavnik se definiše kao klasa koja nasljeđuje klasu RESOURCE, uz dodatno definisana polja što se vidi iz sljedećeg primjera, koje uz svakog nastavnika pridružuje njegovo sedmično opterećenje, termine kada je nedostupan za nastavu, kategorije kojim pripada i funkcije koje mjere kvalitet rasporeda za tog nastavnika..

```
class TEACHER inherit RESOURCE
load :INTEGER
unavailable : SET[TIME]
categ : SET [TEACHER_CATEGORY]
functions
defects : SEQ[DEFECT] = (
defect_check(TeacherClash,clashes)+
defect_check(TeacherOverload,overload )
)
end
```

U instanci rasporeda, nastavnik se inicijalizira sa navođenjem njegovih atributa. U ovom primjeru, prvi je naslijeđen od klase RESOURCE i predstavlja njegovo ime, dok se navode atributi load, unavailable i categ.

```
TEACHER("Milic", 22, {}, {English})
```

U primjeru za srednje škole, klasa MEETING predstavlja događaje koji se vremenski raspoređuju: nastavne časove i sastanke, od kojih svaka vrsta događaja ima svoju klasu nasljednika klase MEETING. Tada, da definišemo da nastavnik Smith predaje čas sa varijablom xxa, 4 puta sedmično sa jednim dvočasom, u instanci se piše

```
xxa.teachers_requirements=
[FixedTeacher(Smith)]
xxa.times_required:=4
xxa.times_condition:=Double
```

Iako razdvajanje formulacije i instance sakriva implementacijske detalje od korisnika sintaksa je i dalje pogodnija za poznavaoce programskih jezika nego za krajnje korisnike.

C. Adaptirani CLIPS

CLIPS [5] je dodatak jeziku C za pravljenje ekspertnih sistema. U radu [6] korišten je kao ugrađeni domenski jezik za definisanje pravila rasporeda časova. Sintaksa je slična Lisp-u. Osnovna je defrule, koja navodi skup pravila koja dovode do određenog zaključka.

Sljedeći primjer opisuje kako se definiše skup uslova koje jedan čas mora zadovoljiti da bi bio raspoređen.

```
(defrule Schedule-A-Course::T22-L1-rule
  (and
    (Course (id ?id1)
      (name ?name1)
      (session ?session1)
      (class ?class1)
      (credit-hours ?credit-hours1)
      (meeting $?meeting1)
      (location $?location1)
    )
    (Instructor
      (id ?id2)
      (status ?status)
      (exclusive-set $?exclusive-set2)
      (preference-set $?preference-set2)
      (course $?course2)
    )
  )
  (test (member$ ?id1 $?course2))
  (forall
    (Course
      (id ?id1)
      (name ?name1)
      (session ?session1)
      (class ?class1)
      (credit-hours $?credit-hours1)
      (meeting $?prel ?one1 $?post1)
      (location $?location1)
    )
    (test (not
      (member ?one1 $?exclusive-set2))
    )
  )
  =>
  (assert (constraint-satisfaction
    (constraint T22-L1)))
  )
```

Zbog slabije čitljivosti ovakvog formata, sistem koristi Java korisnički interfejs iz koga se generišu podaci u relacionoj bazi i odgovarajuća ograničenja.

D. ECLⁱPS^e

Constraint Logic Programming je generalizacija paradigme logičkog programiranja, čiji je glavni predstavnik Prolog. Ograničenja se inkrementalno generišu tokom izvršavanja programa i zatim prepuštaju nekom sistemu za njihovo razrješavanje poput linearnog programiranja. ECLⁱPS^e je jezik ovakve paradigme, korišten u radu [7]. Sintaksa opisa zahtjeva je u formi Hornovih klauzula, poznatih iz jezika Prolog. Na primjer, klauzulom

```
teaches(einstein,physics).
```

definiše se koji nastavnik predaje koji predmet, dok se nastavnici i predmeti navode u formi liste.

Na prvi pogled sintaksa izgleda prilično intuitivno. No, neki elementi jezika su nepraktični i nastali su kao posljedica dizajna logičkih programskih jezika. Za svako odjeljenje je potrebno navoditi listu termina, a takođe i listu uređenih parova termina i prostorija. Elementi liste Rooms ne predstavljaju prostorije, nego listu varijabli čiji ukupan broj jednak sumi ukupnog broja termina svih odjeljenja.

```
Periods1=[P1,P2,P3,P4,P5,P6,P7],
Periods2=[P8,P9,P10],
Rooms=[R1,R2,R3,R4,R5,R6,R7,R9,R10],
Rooms1=[(P1,R1),(P2,R2),...,(P7,R7)],
Rooms2=[(P8,R8),(P9,R9),...,(R10,R10)]
noclash(Rooms1st,Rooms2nd)
```

V. JEZICI BAZIRANI NA XML

A. TTML

XML je zbog jednostavnog načina izražavanja kompleksnih struktura podataka čest bazni jezik za domensko specifične jezike. Tako su i za definisanje zahtjeva za raspored časova definisana dva jezika bazirana na XML. Timetabling Markup Language (TTML) opisan u radu [8]. TTML je baziran na MathML, Korjenski element se zove <time-tabling>, a ulazni podaci se navode unutar elementa <input-data>. Opis rasporeda se sastoji od varijabli (element <variables>), domena (element <domains>) i ograničenja (element <constraints>). Unutar varijabli i domena koriste se skupovi atributa i vektori koji omogućavaju navođenje više vrijednosti. Na primjer, skup predmeta i broj studenata koji ih slušaju navode se sljedećim XML elementima.

```
<declare> <ci>V</ci>
  <vector>
    <ci duration="2"> CSE211.01</ci>
    <ci duration="2"> CSE211.02</ci>
    <ci> CSE311.01</ci>
    <ci> CSE462.01</ci>
  </vector>
</declare>
<declare> <ci>noOfStudents</ci>
  <vector>
    <ci> 34</ci>
    <ci> 27</ci>
    <ci> 20</ci>
    <ci> 25</ci>
  </vector>
</declare>
```

U okviru domena definišu se skupovi vrijednosti koje varijable mogu imati, npr. lista svih vremenskih termina ili prostorija koje se mogu dodijeliti pojedinom času. Ograničenja koriste funkcije koje poznaje MathML, uz dodatak devet specijalizovanih <notsame> (dodijeljene varijable su međusobno različite), <nooverlap> (vremenski periodi varijabli se ne preklapaju), <preset> (skup predstavlja domen varijable), <exclude> (skup se isključuje iz domena varijable), <ordering> (varijabla je manja, jednaka ili veća od druge varijable), <eventsp> (opseg razlike između dvije varijable), <fullspr> (ograničenje broja dodijeljenih varijabli),

<freespr>(ograničenje broja nedodijeljenih varijabli)
<attrcomp> (opseg vrijednosti atributa varijabli), <resnoclash>
(bez preklapanja domena) i <chksum> (provjera koliko je polja
u varijabli dodijeljeno)

Prednost XML baziranih formata je u njihovoj proširivosti i
popularnosti, ali se ne mogu smatrati preglednim za ručno
generisanje.

B. XHSTT

Na takmičenju za rješavanje problema rasporeda časova
2011, prosti format opisan u [2] zamijenjen je kompleksnim i
fleksibilnim formatom opisanim u [9]. Za razliku od TTML
koji koristi relativno apstraktan pojam varijable i ograničenja
vezanih za njihovo poređenje, XHSTT časove koje treba
dodijeliti naziva pojmom Event, dok su njemu pridruženi
nastavnici, razredi, prostorije, pa i drugi eventualno potrebni
objekti obuhvaćeni pojmom Resource. Format uključuje
formulaciju zahtjeva i rješenje, u cilju evaluacije. Kroz
različite verzije formata došlo je do izmjene korijenskog XML
elementa, kako bi postalo moguće čuvati više instanci
problema i rješenja u rasporedu. Unutar svake instance definišu
se termini (element <Time>), grupe termina čiji je specijalan
slučaj dan (element <TimeGroup>), tipovi resursa
(<ResourceType>), grupe resursa (<ResourceGroup>) i sami
resursi (<Resource>). Glavni dio <Instance> elementa je opis
samih časova (<Event> i <EventGroup>).

Unutar elementa <Constraints> navode se ograničenja koja
raspored mora ispuniti: <AssignResourceConstraint> (skup
događaja kojim se dodjeljuje resurs), <AssignTimeConstraint>
(skup događaja kojim se dodjeljuje termin),
<SplitEventsConstraint> (limit na broj odvojenih termina, na
koje se rastavlja događaj), <LinkEventsConstraint>
(istovremeni časovi), <DistributeSplitEventsConstraint> (broj
blok časova datog događaja), <PreferResourcesConstraint>
(željeni nastavnici ili prostorije), <OrderEventsConstraint>
(redosljed časova), <AvoidSplitAssignmentsConstraint>
(ograničenja na održavanje događaja koji traje više termina u
istoj prostoriji), <SpreadEventsConstraint> (raspoređivanje
časova istog predmeta unutar sedmice),
<AvoidClashesConstraint> (sprečavanje istovremenog
korištenja resursa), <AvoidUnavailableTimesConstraint>
(termini kada je resurs nedostupan),
<LimitIdleTimesConstraint> (ograničenja na rupe u
rasporedu), <ClusterBusyTimesConstraint> (ograničenje
resursa na skupove termina, npr da razred ima nastavu u 5 od 7
dana u sedmici), <LimitBusyTimesConstraint> (ograničenje
resursa na broj termina, npr da razred ima nastavu u samo
jednoj smjeni) i <LimitWorkloadConstraint> (ograničenje
težine predmeta).

Na primjer, ograničenje da svi đaci imaju maksimum pet
rupa u rasporedu postiže se sljedećom definicijom.

```
<IdleTimesConstraint  
Id="StudentIdleTimes">  
<Name>At most five idle times for  
students per week</Name>  
<Required>>false</Required>  
<Weight>10</Weight>  
<CostFunction>Quadratic</CostFunction>  
<TimeSlotGroups>  
<TimeSlotGroup Reference="Monday"/>  
<TimeSlotGroup Reference="Tuesday"/>  
<TimeSlotGroup Reference="Wednesday"/>  
<TimeSlotGroup Reference="Thursday"/>  
<TimeSlotGroup Reference="Friday"/>  
</TimeSlotGroups>  
<Minimum>0</Minimum>  
<Maximum>5</Maximum>  
<ResourceGroups>  
<ResourceGroup  
Reference="AllStudents"/>  
<ResourceGroups>  
</IdleTimesConstraint>
```

Ovaj format se sve više nameće kao standard za definisanje
rasporeda časova zbog svoje snage, prilagođenosti problemu i
postojanje implementacije više algoritama za rješavanje
problema rasporeda časova baziranih baš na ovom formatu.
Međutim, njegova kompleksnost negativno utiče na čitljivost
formata, kako za parsiranje i implementaciju, tako i za unos
podataka.

VI. JEZICI SA SINTAKSOM BLISKOM GOVORNOM JEZIKU

A. Unilang

Kao alternativa tehnički orjentisanim formatima i jezicima,
razvijeni su i jezici čija sintaksa više liči na govorni jezik.
Ovakvi jezici su stoga pogodniji za unos zahtjeva za raspored,
mada su manje pogodni za čuvanje zahtjeva dobijenih na drugi
način poput GUI-a. Prvi takav jezik je Unilang [10]. Zahtjevi
za raspored izgledaju kao u sljedećim rečenicama.

```
teacher smith have exactly 5  
consecutive events in a day  
periods are autumn spring  
events are math101 eng350 his20  
events hs20 math101 are simultaneously  
Implementacija uključuje konverziju u ECLPSE. Jezik nije  
proširivan opisima naprednijih zahtjeva, poput dvočasa ili  
odjeljenja, koja ne mogu imati istovremeno nastavu. U  
spomenutom radu je istaknuto da se nema namjera proširivati  
jezik.
```

B. REDOSPLAT

Jezik opisan u [11] je u međuvremenu proširen da omogućiti
formulisanje svih vrsta ograničenja spomenutih u istraživanju
[2], kao i rada u više smjena. U prvoj verziji nije imao posebno
ime, pa mu je sada dato ime koje predstavlja REadable
Domain SPECific LAnguage for Timetabling. Implementiran je
u C++ koristeći cjelobrojno linearno programiranje pomoću
biblioteke GLPK, a u planirano je i korištenje drugih

algoritama. Način formulacije je jednostavan. Zahtjevi se formulišu rečenicama kao u primjeru:

Ia is a class.

R101 is a room.

Mike is teacher.

Tu2 is timeslot number 2 on Tuesday.

Mike is unavailable on Tu2.

Mike teaches Physics to Ia, 4 times a week, with 2 doubles.

Podržane su rečenice za definisanje časova (događaja), dana, smjena, termina, zgrada, razreda, grupa, spojenih razreda, nastavnika, predmeta i njihovog međusobnog redoslijeda, nedostupnosti, želja za terminima, lokacija, ograničenja na istovremenu nastavu, dnevnih limita, vremena putovanja, preferencija u cilju, zahtjeva za pauzama. Opis časa može uključiti nastavnika ili listu nastavnika (ako rade u paru ili se bira jedan od njih), sedmični fond časova, odjeljenje ili listu odjeljenja kome se drži nastava, preporučene prostorije između kojih se bira slobodna, način raspoređivanja časova unutar sedmice, broj dvočasa i tročasa, preferirane i fiksirane termine i zahtjeve za istovremenosti ili vremenski pomak u odnosu na druge časove.

Jezik ne podržava pojedinačne đake/studente i nestandardne resurse, kao ni imperativne elemente za programiranje novih ograničenja.

VII. POREĐENJE

Poređenje ovih domenskih jezika dato je u tabeli 1. Tabela uključuje manji skup ukupnih mogućnosti ovih jezika. Vrijednosti u poljima znače: Y: u potpunosti, M srednji nivo podrške, N niski nivo podrške, P potrebno programirati.

TABELA I. POREĐENJE MOGUĆNOSTI

	ITC'07	TTLI B	STTL S	CLIP S	ECL ¹ PS*	TTML	XHST T	Unilang	REDO SPLAT
Čitljivost	M	M	M	N	M	N	M	Y	Y
Vlastita ograničenja	N	Y	Y	Y	Y	N	N	N	N
Nastavnici	Y	P	P	P	Y	Y	Y	Y	Y
Odjeljenja	Y	P	P	P	Y	Y	Y	Y	Y
Prostorije	Y	P	P	P	Y	Y	Y	Y	Y
Studenti	N	P	P	P	P	N	Y	Y	N
Ostali resursi	N	P	P	P	P	Y	Y	N	N
Smjene i spec. periodi	N	P	P	P	P	N	Y	N	Y
Nedostupnost resursa	Y	P	P	P	P	Y	Y	Y	Y
Preferencije termina	N	P	P	P	P	N	Y	N	Y
Blok časovi	N	P	P	P	P	N	Y	N	Y
Fiksiranje termina	N	P	P	P	P	Y	Y	N	Y
Udaljene lokacije	N	P	P	P	P	N	N	N	Y
Istovremeni časovi	N	P	P	P	P	Y	Y	Y	Y
Kontrola pauza	N	P	P	P	P	Y	Y	Y	Y
Opšti tipovi podataka	N	Y	Y	N	N	N	N	N	N
Kompaktan zapis	Y	N	N	N	N	N	N	N	N
Smještaj rješenja	N	N	N	N	N	N	Y	N	N
Ugradiv u aplikaciju	Y	N	N	N	N	N	Y	N	Y

VIII. ZAKLJUČAK

Različita istraživanja algoritama za rješavanje rasporeda časova dovela su i do više domenskih jezika koji opisuju potrebne zahtjeve. Mogućnosti, čitljivost i proširivost devet opisanih jezika varira. XHSTT je trenutno najkompletniji i najviše održavani domenski jezik ali za njegovu upotrebu je potreban dodatni korisnički interfejs za unos i generisanje XML datoteka ili dopuna čitljivijim jezikom kao što su REDOSPLAT ili Unilang.

LITERATURA

- [1] N. Pillay. A survey of school timetabling research. *Annals of Operations Research*. 1;218(1):261-93., 2014 Jul
- [2] L. Di Gaspero, B. McCollum, A. Schaerf. The second international timetabling competition (ITC-2007): Curriculum based course timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0, Queen's University, Belfast, United Kingdom, 2007.
- [3] E.K.Burke, J.H.Kingston, P.A.Pepper. A standard data format for timetabling instances. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 213-222). Springer Berlin Heidelberg, 1997 Aug 20..
- [4] J. H. Kingston. Modelling timetabling problems with STTL. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 309-321). Springer Berlin Heidelberg, 2000 Aug 16.
- [5] G. Riley, CLIPS: An Expert System Building tool, In *NASA, Washington, Technology: The Second National Technology Transfer Conference and Exposition, Volume 2* p 149-158, 2001
- [6] L. F. Lai LF, C. C. Wu, N. L. Hsueh, L.T. Huang LT, S.F. Hwang. An artificial intelligence approach to course timetabling. *International Journal on Artificial Intelligence Tools*. 2008 Feb;17(01):223-40, 2008.
- [7] M. Kambi, and D. Gilbert., *Timetabling in constraint logic programming*, INAP., 1996
- [8] E. Özcan. Towards an XML-based standard for timetabling problems: TTML. In *Multidisciplinary Scheduling: Theory and Applications* (pp. 163-185). Springer US, 2005..
- [9] G. Post G, S. Ahmadi, S. Daskalaki, J.H. Kingston, J. Kyngas, C. Nurmi, D. Ranson. An XML format for benchmarks in high school timetabling. *Annals of Operations Research*.;194(1):385-97, 2012 Apr 1.
- [10] L. P. Reis, E. Oliveira E. language for specifying complete timetabling problems. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 322-341). Springer Berlin Heidelberg, 2000 Aug 16.
- [11] Ribić S. A school timetable description language. In *Telecommunications Forum (TELFOR), 2012 20th* (pp. 1709-1712), IEEE, 2012 Nov 20

ABSTRACT

Timetable generation in schools is a subject of interest not only from the standpoint of solving, but also from the standpoint of formulation, There is a plenty of different types of constraints on the resource scheduling. They resulted with the need for defining a specific language for describing them. This survey paper focuses on nine specialized language that can describe the requirements for timetabling in schools. It points to the basic syntax elements, advantages and disadvantages of the formats.

Domain Specific Languages and Data Format for School Timetable Requirements Formulation

Samir Ribić