



SVEUČILIŠTE U ZAGREBU

GEODETSKI FAKULTET

SAŠA VRANIĆ

**Modeliranje transakcijskoga sustava za  
upravljanje tijekom rada nad prostornom  
sastavnicom katastarskih čestica**

DOKTORSKI RAD

Zagreb, 2018.



SVEUČILIŠTE U ZAGREBU  
GEODETSKI FAKULTET

SAŠA VRANIĆ

**Modeliranje transakcijskoga sustava za  
upravljanje tijekom rada nad prostornom  
sastavnicom katastarskih čestica**

DOKTORSKI RAD

Mentor:

Doc. dr. sc. Hrvoje Matijević

Zagreb, 2018.



UNIVERSITY OF ZAGREB

FACULTY OF GEODESY

SAŠA VRANIĆ

**Modelling transactional workflow management  
system over spatial component of cadastral  
parcels**

DOCTORAL THESIS

Supervisor:

Assist. prof. Hrvoje Matijević

Zagreb, 2018.

---

## I. AUTOR

---

<b>Ime i prezime</b>	Saša Vranić
<b>Datum i mjesto rođenja</b>	20. ožujka 1986., Sisak
<b>Sadašnje zaposlenje</b>	Asistent na Geodetskom fakultetu Sveučilišta u Zagrebu

---

## II. DOKTORSKI RAD

---

<b>Naslov</b>	Modeliranje transakcijskoga sustava za upravljanje tijekom rada nad prostornom sastavnicom katastarskih čestica
<b>Broj stranica</b>	138
<b>Broj priloga</b>	0
<b>Broj slika</b>	71
<b>Broj tablica</b>	2
<b>Broj bibliografskih podataka</b>	126
<b>Ustanova i mjesto gdje je rad izrađen</b>	Geodetski fakultet Sveučilišta u Zagrebu
<b>Znanstveno područje</b>	Tehničke znanosti
<b>Znanstveno polje</b>	Geodezija
<b>Znanstvena grana</b>	Primijenjena geodezija
<b>Mentor</b>	Doc. dr. sc. Hrvoje Matijević
<b>Oznaka i broj rada</b>	93

---

## III. OCJENA I OBRANA

---

<b>Datum prijave teme</b>	26. ožujka 2015. godine
<b>Datum sjednice Fakultetskog vijeća na kojoj je doktorski rad prihvaćen</b>	26. travnja 2018. godine
<b>Sastav povjerenstva koje je ocijenilo doktorski rad</b>	Prof. dr. sc. Miodrag Roić Doc. dr. sc. Dražen Tutić Doc. dr. sc. Zdravko Galić
<b>Datum obrane doktorskog rada</b>	14. svibnja 2018. godine
<b>Sastav povjerenstva pred kojim je branjen doktorski rad</b>	Prof. dr. sc. Miodrag Roić Doc. dr. sc. Dražen Tutić Doc. dr. sc. Zdravko Galić

---

# Zahvala

Prije svega zahvaljujem mentoru doc. dr. sc. Hrvoju Matijeviću na korisnim savjetima, konstruktivnim kritikama, ohrabrivanju, pomoći i strpljenju. Zahvaljujem se dr. sc. Mariju Mađeru na čitanju rada te korisnim savjetima i primjedbama. Na uloženoj se trudu i vremenu zahvaljujem članovima povjerenstva za ocjenu rada: prof. dr. sc. Miodragu Roiću, doc. dr. sc. Draženu Tutiću i doc. dr. sc. Zdravku Galiću.

Veliku zahvalnost dugujem obitelji na strpljivosti i podršci tijekom poslijediplomskoga studija, a posebno tijekom završne izrade doktorskog rada.

Ovaj je doktorski rad u potpunosti podržan od strane Hrvatske zaklade za znanost u okviru projekta HRZZ-IP-2013-11-7714, Development of Multipurpose Land Administration System (DEMLAS).

# Sažetak

Podaci u sustavima za upravljanje zemljištem održavaju se kroz formalno definirane postupke koji moraju osigurati sigurnu transformaciju iz jednog ispravnoga stanja u drugo. Sustavi za upravljanje bazama podataka i transakcijski modeli osiguravaju ispravnost, ali im nedostaje fleksibilnosti kod modeliranja poslovnih procesa te pružaju slabu podršku za heterogena informatička okruženja (web servisi, različiti računalni programi i sustavi). Transakcijski sustavi za upravljanje tijekom rada pružaju fleksibilnost i mogu osigurati ispravnost podataka. Model područja upravljanja zemljištem pruža odličnu osnovu za modeliranje statičke komponente sustava za upravljanje zemljištem, ali ne pruža elemente za modeliranje dinamičke komponente, odnosno procesa.

U ovome je radu razvijen konceptualni model sustava za upravljanje tijekom rada koji podržava transakcijske koncepte te podržava upravljanje procesima nad prostornom sastavnicom katastarskih čestica. Razvijeni model podataka omogućava pohranjivanje elemenata Petrijevih mreža te je podijeljen na dva dijela, opći i prošireni. Opći model osigurava ispravnost procesa na razini objekta te je primjenjiv na katastarske podatke, ali ne uzima u obzir prostornu sastavnicu. Prošireni model osigurava ispravnost prostornom definicijom zahvaćenoga područja procesa te se koristi za modeliranje procesa na katastarskim česticama čija se prostorna sastavnica temelji na poligonima. U skladu s modelom podataka, pomoću elemenata tijekom rada definirani su razni tijekomovi rada (protokoli) (pesimistični, optimistični i altruistični) koji podržavaju različite razine konkurentnosti procesa.

Prema razvijenom je konceptualnome modelu implementiran testni sustav kako bi se na praktičnim primjerima dokazala ispravnost hipoteza te provjerila primjenjivost razvijениh koncepata. Ispravnost prostorne sastavnice katastarskih čestica je osigurana primjenom ACID (Atomicity, Consistency, Isolation, Durability) svojstava na procese, a integracijom uvjeta ispravnosti u sustav za upravljanje tijekom rada je osigurana automatizacija i ubrzavanje kontrole ispravnosti procesa te je povećana učinkovitost sustava koja se može dodatno povećati primjenom jednog od razvijениh tijekomova rada.

# Abstract

Land administration system data are maintained through formally defined procedures which need to provide a safe transformation from one consistent state to another. Database management systems and transaction models ensure consistency, but lack the flexibility in business modelling and provide poor support for heterogeneous IT environments (web services, various computer programs and systems). Transactional workflow management systems provide flexibility and can provide consistency of data. The land administration domain model provides an excellent basis for modelling static component of land administration system but doesn't provide elements to model dynamic component, i. e. the processes.

In this paper, a conceptual model of workflow management system is developed, which supports transactional concepts and supports the management of the spatial component of cadastral parcels. The developed data model allows storing Petri nets elements and is divided into two parts, generic and extended. Generic model ensures the consistency of processes on object level and is applicable to cadastral data, but doesn't take into account the spatial component. The extended model ensures consistency by spatially defining the affected area of a process and is used to model processes on cadastral parcels spatially represented by polygons. According to the data model, by using workflow elements different workflows (protocols) have been defined (pessimistic, optimistic and altruistic). Each of them supports different level of process concurrency.

According to developed conceptual model, a test system has been implemented to demonstrate on practical examples the correctness of the set hypothesis and to check the applicability of the developed concepts. The consistency of the spatial component of cadastral parcels is ensured by applying ACID (Atomicity, Consistency, Isolation, Durability) properties to processes. By integrating the consistency criteria into the workflow management system, automation and enhancement of the consistency checking is ensured. Also, system efficiency is increased which can be further increased by applying one of the developed workflows.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Pregled dosadašnjih istraživanja . . . . .	2
1.2	Povod za istraživanje . . . . .	3
1.3	Cilj i hipoteze istraživanja . . . . .	4
1.4	Opseg istraživanja . . . . .	4
1.5	Metodologija istraživanja . . . . .	5
1.6	Korišteni alati i podaci . . . . .	6
1.7	Organizacija doktorskog rada . . . . .	7
<b>I</b>	<b>Teoretska osnova i pregled dosadašnjih istraživanja</b>	<b>9</b>
<b>2</b>	<b>Baze podataka i transakcijski modeli</b>	<b>10</b>
2.1	Osnovni transakcijski model . . . . .	11
2.1.1	Definicija transakcije . . . . .	12
2.1.2	Pesimistični protokoli za upravljanje konkurentnim transakcijama . . . . .	13
2.1.3	Optimistični protokoli za upravljanje konkurentnim transakcijama . . . . .	19
2.1.4	Upravljanje konkurentnim transakcijama verzioniranjem . . . . .	20
2.1.5	Izolacija transakcije . . . . .	22
2.2	Napredni transakcijski modeli . . . . .	24
2.3	Upravljanje konkurentnim transakcijama nad prostornim podacima . . . . .	28
2.4	Rekapitulacija . . . . .	36
<b>3</b>	<b>Sustavi za upravljanje tijekom rada</b>	<b>38</b>
3.1	Terminologija . . . . .	39
3.2	Referentni model sustava za upravljanje tijekom rada . . . . .	40
3.3	Komponente tijeka rada . . . . .	42
3.4	Modeliranje i grafički prikaz procesa . . . . .	44
3.4.1	Notacija za modeliranje poslovnih procesa . . . . .	45
3.4.2	UML dijagrami aktivnosti . . . . .	46
3.4.3	Petrijeve mreže . . . . .	47
3.4.4	Osvrt na jezike za modeliranje procesa . . . . .	50
3.5	Transakcijski sustavi za upravljanje tijekom rada . . . . .	51



3.6	Prostorni sustavi za upravljanje tijekom rada . . . . .	57
3.7	WFMS i učinkovitost poslovnih procesa . . . . .	60
3.8	Rekapitulacija . . . . .	61
<b>II</b>	<b>Konceptualni model i korištenje testnoga sustava</b>	<b>63</b>
<b>4</b>	<b>Primjena WFMS-a na katastarske podatke</b>	<b>64</b>
4.1	Pregled ulaznih postavki . . . . .	65
4.2	Dijagram klasa modela procesa . . . . .	70
4.3	Povezivanje modela WFMS-a s katastarskim česticama . . . . .	74
4.4	Modeliranje procesa . . . . .	80
4.4.1	Pesimistični tijek rada . . . . .	81
4.4.2	Optimistični tijek rada . . . . .	83
4.5	Poboljšanje konkurentnosti procesa uvođenjem međnih točaka . . . . .	86
4.5.1	Prepoznavanje promjena . . . . .	87
4.5.2	Altruistični tijek rada . . . . .	93
4.6	Integracija WFMS-a u LADM . . . . .	95
4.7	Rekapitulacija . . . . .	98
<b>5</b>	<b>Arhitektura testnoga sustava i provjera slučajeva korištenja</b>	<b>99</b>
5.1	Model podataka . . . . .	99
5.2	Izvorni podaci i učitavanje u bazu podataka . . . . .	100
5.3	Arhitektura testnoga sustava . . . . .	101
5.4	Definicija tijeka rada u sustavu . . . . .	103
5.4.1	Pesimistični tijek rada . . . . .	103
5.4.2	Optimistični tijek rada . . . . .	107
5.4.3	Altruistični tijek rada . . . . .	108
5.5	Ispitivanje tijekova rada na primjerima . . . . .	108
5.5.1	Konkurentni procesi sa zajedničkom susjednom česticom . . . . .	108
5.5.2	Konfliktni konkurentni procesi . . . . .	111
5.6	Rekapitulacija . . . . .	112
<b>III</b>	<b>Rezultati istraživanja</b>	<b>114</b>
<b>6</b>	<b>Zaključak</b>	<b>115</b>
6.1	Izvorni znanstveni doprinos . . . . .	116
6.2	Daljnja istraživanja . . . . .	117
	<b>Literatura</b>	<b>118</b>
	<b>Popis tablica</b>	<b>128</b>

<b>Popis slika</b>	<b>129</b>
<b>Popis kratica</b>	<b>132</b>
<b>Životopis</b>	<b>134</b>

# 1. Uvod

Većina je znanstvenih istraživanja u domeni geoinformacijskih sustava tradicijski usmjerena na analitičke funkcionalnosti, kako u teoriji, tako i u praksi. Temporalna je sastavnica bitna za sustave za upravljanje zemljištem i u tome su području u posljednjih 30-ak godina provedena brojna istraživanja u području modeliranja temporalnih podataka u bazama podataka, geoinformacijskim i katastarskim sustavima (Al-Taha, 1992; Peuquet i Wentz, 1994; Langran i Chrisman, 1988; Peuquet i Duan, 1995). Međutim, metodologija ažuriranja prostornih podataka, posebno s transakcijskim svojstvima i nad velikim skupovima podataka, slabo je istražena. Implementacije transakcijskih sustava u sustavima za upravljanje zemljištem često nemaju jasnu koncepcijsku osnovu. U sustavima za upravljanje zemljištem neophodno je da postupci u kojima se mijenjaju podaci budu sigurni i učinkoviti budući da katastarski podaci predstavljaju društvene odnose formalizirane imovinskim pravom (prava, ograničenja i obaveze (eng. *right, restriction, responsibility - RRR*)).

Sustav za upravljanje zemljištem (eng. *Land Administration System*) omogućava učinkovito gospodarenje zemljištem. Za potrebe upravljanja zemljištem i interesima na njemu postoje razni upisnici (eng. *register*) u koje se upisuju razne informacije. Upisnici su službeni zapisi podataka koji se uspostavljaju i vode na temelju propisa, a osobe ostvaruju prava ili imaju obaveze na temelju zapisa u njima. Najvažniji je upisnik zemljišta katastar, jer podržava sve zadaće učinkovitog upravljanja zemljištem. Zbog složenosti modela i procesa katastra te raznovrsnosti konkretnih izvedbi često se naziva i katastarskim sustavom (Roić, 2012). Konceptualni model koji opisuje sustave za upravljanje zemljištem jest Model područja upravljanja zemljištem (eng. *Land Administration Domain Model (LADM)*) (ISO, 2012a). Iako LADM pruža vrlo dobru osnovu za modeliranje statičke komponente sustava za upravljanje zemljištem, ali ne pruža elemente za modeliranje dinamičke komponente, odnosno procesa. Kod sustava za upravljanje zemljištem temporalna komponenta je bitna, jer se odnosi između osoba i zemljišta kroz vrijeme mijenjaju (kroz kupoprodaju, nasljeđivanje i razne tehničke promjene poput spajanja i dijeljenja katastarskih čestica).

## 1.1. Pregled dosadašnjih istraživanja

Većina sustava za upravljanje zemljištem koji upravljaju podacima na elektronički način koriste sustave za upravljanje bazama podataka (SUBP) (eng. *Database Management System (DBMS)*) za pohranjivanje podataka. Uglavnom su to relacijski, odnosno objektno-relacijski SUBP. Relacijski i objektno-relacijski modeli baza podataka utemeljeni su na relacijskoj algebri (Codd, 1970), a upravljanje se podacima obavlja kroz transakcije (svojstva Atomicity, Consistency, Isolation, Durability (ACID)) (Gray, 1981; Härder i Reuter, 1983) i osiguravaju transformaciju podataka iz jednoga konzistentnog stanja u drugo što je za sustave za upravljanje zemljištem neophodno. Iako transakcije osiguravaju ispravnost, tradicijski one podrazumijevaju kratko trajanje transakcije s ciljem što veće konkurentnosti transakcija. Zbog ograničenja osnovnoga transakcijskog modela u skladu sa svojstvima ACID, razvijeni su napredni transakcijski modeli koji proširuju određene segmente osnovnoga transakcijskog modela (primjerice, podrška za duge transakcije, ugnježđivanje transakcija, povezivanje transakcija u kompleksnije strukture) (Moss, 1981; Garcia-Molina i Salem, 1987; Pu i dr., 1988; Weikum i Schek, 1991; Dayal i dr., 1991). Međutim, napredni transakcijski modeli koriste funkcionalnosti SUBP-a koje ne pružaju dovoljnu razinu fleksibilnosti koja je potrebna za modeliranje poslovnih procesa.

Istraživanje se transakcija nad prostornom sastavnicom katastarskih podataka svodi na tek nekoliko radova, a tematika je kompleksna, jer su prostorni podaci kompleksni (zbog različitih geometrijskih i topoloških struktura, zbog eventualnoga utjecaja na susjedne objekte ili zbog implicitnih prostornih odnosa s drugim skupovima prostornih podataka). U nekoliko se postojećih radova (van Oosterom, 1997; Matijević i dr., 2008; Vranić i dr., 2015), autori najčešće bave definiranjem uvjeta integriteta prostorne sastavnice, a konkurentni su procesi zanemareni ili samo spomenuti. Nedostatak teoretske osnove i u praksi često uzrokuje implementaciju sustava koji ima ograničenu fleksibilnost, zahtijeva puno manualnoga rada operatera te je teško raspodijeliti rad između raznih dionika. Iako postoje radovi koji se bave primjenom transakcijskih modela na prostorne podatke (Kuo i dr., 1998), oni pružaju ograničenu podršku za osiguravanje konzistentnosti prostorne sastavnice. Uvjeti integriteta ne mogu se definirati na razini transakcije već određeni uvjet integriteta mora vrijediti na razini čitave baze podataka što je također ograničavajući faktor, jer je u sustavima za upravljanje zemljištem moguće da različite vrste transakcija podliježu različitim uvjetima integriteta (primjerice, dvije zgrade ne smiju se preklapati, dok se dva interesa smiju (primjerice, pravo služnosti zbog prolaska voda i pravo prolaska katastarskom česticom). Detaljna je analiza istraživanja u domeni transakcija u SUBP-u te o upravljanju transakcijama nad prostornim podacima objašnjena u poglavlju 2.

Jedan od razloga razvoja sustava za upravljanje tijekom rada (eng. *Workflow Management System (WFMS)*) jest potreba za učinkovitijim i fleksibilnijim upravljanjem poslovnim procesima, a jedna je od glavnih prednosti WFMS-a fleksibilnost. Tijek rada se sastoji od niza povezanih zadataka koji djelomično ili u cijelosti automatiziraju poslovni proces. Zadaci mogu biti dodijeljeni različitim dionicima, a zadatak može predstavljati transakciju

u SUBP-u, web servis ili računalni program. Na taj se način zadaci mogu brže obaviti, jer se mogu distribuirati različitim dionicima što smanjuje njihovu opterećenost poslom, a podrška za različite vrste zadataka (poput web servisa) ubrzavaju interakciju s dionicima i vanjskim sustavima, a sve s ciljem povećavanja opće učinkovitosti. Jedan je od glavnih nedostataka WFMS-a slaba podrška za transakcijske koncepte s ciljem osiguravanja ispravnosti i oporavka podataka. Taj se nedostatak pokušao riješiti integracijom transakcijskih koncepata u WFMS, odnosno definicijom koncepta transakcijskoga WFMS-a (Alonso i dr., 1996; Eder i Liebhart, 1994; Grefen i Vonk, 2006). Međutim, transakcijski WFMS-i pružaju ograničenu podršku za osiguravanje ispravnosti i oporavka podataka, pogotovo u domeni prostornih podataka gdje praktično nema istraživanja na tu temu. Detaljna je analiza istraživanja u domeni sustava za upravljanje tijekom rada objašnjena u poglavlju 3.

## 1.2. Povod za istraživanje

S obzirom na to da primjena WFMS-a općenito omogućava podizanje učinkovitosti poslovnih procesa automatizacijom i korištenjem informatičke infrastrukture (WFMC, 1995; Georgakopoulos i dr., 1995), primjenom je WFMS-a u sustavu za upravljanje zemljištem također moguće povećati učinkovitost sustava. Van Osch i Lemmen (2004) objašnjavaju uvođenje WFMS-a u nizozemski katastar s ciljem povećanja učinkovitosti, pouzdanijeg održavanje podataka te veće fleksibilnosti kod upravljanja osobljem na državnoj i lokalnoj razini. Becker i zur Mühlen (1999) definiraju kriterije kojima se može mjeriti učinkovitost upravljanja poslovnim procesima od kojih su dva relevantna za ovaj rad: učinkovitost procesa i učinkovitost resursa. Prvi kriterij definira smanjivanje vremena trajanja procesa ili poštivanje zadanih rokova procesa, dok drugi definira optimalnu raspodjelu zadataka između dionika kako bi se potpomoglo ispunjenje prvoga kriterija. Dumas i dr. (2013) dijele svaki zadatak na dva dijela: vrijeme čekanja i vrijeme obrade. Omjer vremena obrade i ukupnoga trajanja zadatka definira kriterij učinkovitosti zadatka. Dakle, smanjivanjem vremena čekanja na pojedinome zadatku se povećava učinkovitost procesa.

S obzirom na navedena istraživanja, temeljni povod za ovo istraživanje je pronalaženje odgovora na pitanje: može li se transakcijski WFMS primijeniti na procese nad prostornom sastavnicom katastarskih čestica. Integracijom transakcijskih koncepata u skladu sa svojstvima ACID u WFMS bi se osigurala ispravnost prostorne sastavnice katastarskih čestica, a samom bi se primjenom WFMS-a omogućilo povećanje učinkovitosti sustava za upravljanje zemljištem. U skladu s navedenim kriterijima učinkovitosti, automatizacijom kontrole ispravnosti procesa i uvođenjem podrške za višekorisnički rad uz optimalnu raspodjelu opterećenja je moguće dodatno povećati učinkovitost.

### 1.3. Cilj i hipoteze istraživanja

Istraživanje obavljeno u okviru ovoga doktorskog rada ima cilj: pronaći odgovor je li moguće implementirati transakcijski WFMS za upravljanje procesima nad prostornom sastavnicom katastarskih čestica čime bi se općenito omogućilo povećavanje učinkovitosti takvoga sustava uz osiguravanje ispravnosti podataka.

Na temelju toga na početku istraživanja su postavljene hipoteze:

*Konzistentnost prostorne sastavnice moguće je osigurati primjenom koncepta transakcijskoga sustava za upravljanje tijekom rada.*

*Primjenom koncepta transakcijskoga sustava za upravljanje tijekom rada moguće je povećati učinkovitost te ubrzati i automatizirati kontrolu ispravnosti transakcija na prostornoj sastavnici katastarskih čestica.*

### 1.4. Opseg istraživanja

Definicijom opsega istraživanja dane su smjernice koja će se područja obuhvatiti i koje će radnje biti obavljene u okviru istraživanja:

1. vektorski (2D) podaci temeljeni na poligonima, odnosno multipoligonima u skladu s normom ISO 19125 (ISO, 2004a),
2. prostorna sastavnica kojoj je glavni uvjet integriteta ispravnost ravninske particije,
3. prostorna sastavnica katastarskih čestica temeljena na poligonima,
4. definicija protokola istovremenog rada na prostornoj sastavnici katastarskih čestica,
5. definicija različitih tijekova rada (protokola) koje omogućavaju različite razine konkurentnosti,
6. integracija WFMS-a u model područja upravljanja zemljištem (eng. *Land Administration Domain Model (LADM)*)
7. primjena koncepta upravljanja tijekom rada na prostornu sastavnicu katastarskih čestica,
8. izrada testnoga sustava (provjera ispravnosti hipoteza te primjenjivost razvijenih koncepata).

## 1.5. Metodologija istraživanja

U skladu s definiranim ciljem, u nastavku se opisuje metodologija istraživanja s ciljem potvrđivanja ispravnosti hipoteza. U okviru istraživanja uzet će se u obzir samo procesi nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Glavni će uvjet integriteta prostorne sastavnice biti ispravnost ravninske particije. S obzirom na to da priprema promjena nad katastarskim česticama može potrajati duže vrijeme, priprema promjene i provjera njezine ispravnosti razdvaja se na zasebne faze, što omogućava modeliranje procesa u skladu s konceptom prijelaznoga neto učinka (eng. *transition net effect*). Razmotrit će se i mogućnost integracije transakcijskoga WFMS-a u model područja upravljanja zemljištem (eng. *LADM*) kako bi se provjerila primjenjivost razvijenoga modela na sustave za upravljanje zemljištem.

Metodologija istraživanja je podijeljena u četiri faze:

1. pregled relevantnih istraživanja,
2. izrada konceptualnoga modela,
  - prva faza u kojoj se ispravnost prostorne sastavnice osigurava na pesimistični način zaključavanjem svih katastarskih čestica zahvaćenih procesom,
  - druga faza u kojoj je model nadograđen kako bi se omogućila potpuna automatizacija kontrole ispravnosti procesa primjenom naprednijih protokola za upravljanje konkurentnim procesima uz uvođenje međnih točaka u model podataka,
3. implementacija testnoga sustava i
4. provjera koncepata korištenjem testnoga sustava.

Nakon što su postavljeni cilj, svrha te opseg istraživanja prepoznata su područja iz kojih je potrebno proučiti stanje tehnologije kako bi se utvrdilo trenutno stanje istraživane problematike te uočili nedostaci dosadašnjih znanstvenih istraživanja (eng. *scientific gap*). Relevantna će istraživanja biti proučena po sljedećim područjima:

- modeliranje prostornih podataka (geometrije i topologije),
- uvjeti integriteta prostornih podataka,
- baze podataka i transakcijski modeli,
- sustavi za upravljanje tijekom rada i
- modeliranje poslovnih procesa.

Izrada konceptualnog modela je podijeljena u dvije faze. U prvoj će se fazi razviti konceptualni model WFMS-a koji podržava transakcijske koncepte u skladu sa svojstvima ACID te podržava upravljanje procesima nad prostornom sastavnicom katastarskih čestica. U prvoj se fazi neće uzeti u obzir učinkovitost već će se primijeniti pesimistični pristup i zaključati sve katastarske čestice zahvaćene procesom kako bi ispitivanje ispravnosti

prostorne sastavnice prema uvjetima integriteta definiranim u (Vranić i dr., 2015) osiguralo ispravnost i u slučaju konkurentnih procesa, odnosno kako bi se osigurala izolacija procesa prema svojstvima ACID. Ako se to uspije, prva će hipoteza biti dokazana.

U drugoj će se fazi konceptualni model transakcijskoga WFMS-a razvijen u prvoj fazi nadograditi kako bi se omogućila potpuna automatizacija kontrole ispravnosti procesa te što veća mogućnost istovremene pripreme procesa primjenom optimističnog i altruističnoga protokola za upravljanje konkurentnim procesima uz uvođenje međnih točaka u model podataka. Istovremenom se pripremom promjena na katastarskim česticama smanjuje vrijeme čekanja jednoga procesa da se oslobodi određeni podatak koji je zaključan u drugome procesu. Ako se to uspije, i druga će hipoteza biti dokazana.

Implementacija testnoga sustava će biti izvedena izradom web aplikacije pomoću programskoga jezika PHP kako bi se ispitala ispravnost hipoteza te primjenjivost razvijenih koncepata. Implementirat će se samo moduli neophodni za definiciju i upravljanje procesima te će biti razvijene metode za ispitivanje ispravnosti prostorne sastavnice katastarskih čestica temeljenih na poligonima. Ostale će se komponente sustava poput upravljanja korisnicima implementirati u nužnom opsegu za funkcioniranje sustava. Nakon implementacije sustava kreirat će se procesi prema razvijenim tijekovima rada te će se na konkretnim primjerima provjeriti ispravnost hipoteza.

## 1.6. Korišteni alati i podaci

Doktorski je rad napisan pomoću programskoga jezika  $\text{\LaTeX}$  korištenjem računalnoga programa TeXstudio. Dijagrami klasa Unified Modeling Language (UML) napravljeni su korištenjem računalnoga programa Enterprise Architect.

Testni je sustav implementiran korištenjem sljedećih alata i podataka:

- **Feature Manipulation Engine (FME)** je programski paket koji omogućava transformaciju podataka između velikoga broja formata podataka.
- **PHP** je skriptni jezik otvorenoga kôda, namijenjen za pisanje dinamičkih web stranica. Korištenjem raznih proširenja omogućava širok skup funkcionalnosti te podržava sve važnije SUBP-e, poput PostgreSQL-a, MySQL-a, Oracle-a, SQLite-a.
- **geoPHP** je PHP biblioteka otvorenoga kôda za upravljanje geometrijskim podacima i obavljanje geometrijskih operacija. Omogućava kreiranje PHP objekata iz raznih formata poput WKT, GeoJSON, KML i slično. Omogućava i pozivanje određenih GEOS<sup>1</sup> funkcija što rezultira povećanjem performansi.
- **PostgreSQL** je sustav za upravljanje bazama podataka otvorenoga kôda. **PostGIS** je dodatak koji omogućava da se u PostgreSQL bazu pohranjuju prostorni podaci

<sup>1</sup>GEOS: <https://geos.osgeo.org/doxygen/>



(geometrijska i topološka struktura). Geometrija se pohranjuje u skladu s normom ISO 19125.

- **GeoServer** je server otvorenoga kôda za dijeljenje prostornih podataka. Omogućava objavu podataka iz svih zastupljenijih formata podataka poput Shapefile, PostgreSQL, Oracle, i tako dalje. Podaci se objavljuju korištenjem otvorenih standarda (poput WMS, WCS, WFS).
- **OpenLayers** je JavaScript biblioteka otvorenoga kôda za izradu interaktivnih web karata. Podržava veliki broj izvora podataka i protokola.
- **jQuery** je još jedna JavaScript biblioteka koja omogućava lakše upravljanje HTML kôdom, animacijama, pisanje kôda koji se može izvršiti na svim preglednicima i slanje AJAX zahtjeva prema serveru.
- **Testni skup podataka.** Uključuje podatke pribavljene u okviru projekta DEMLAS na lokaciji Starogradskog polja na otoku Hvaru. Podaci o katastarskim česticama su dobiveni u CAD formatu te konvertirani u objektno-relacijski model SUBP PostgreSQL proširenu prostornim dodatkom PostGIS odakle su objavljeni pomoću GeoServer-a.

## 1.7. Organizacija doktorskog rada

Doktorski je rad podijeljen u šest poglavlja. U nastavku je kratko objašnjen sadržaj svakoga poglavlja.

**Poglavlje 1** pruža uvod u problematiku kojom se bavi rad. Kroz pregled relevantnih istraživanja prikazuje se važnost teme te nedostaci dosadašnjih istraživanja i pristupa. Uz to je objašnjen povod za istraživanje te se daje pregled glavnih ciljeva rada. Na temelju definirana cilja su postavljene hipoteze istraživanja i definirana metodologija te odabrani alati i podaci za provođenje istraživanja.

**Poglavlje 2** daje pregled transakcijskih modela kao i protokola za upravljanje konkurentnim transakcijama. Daje se i pregled pristupa za kontroliranje konkurentnih transakcija nad prostornim podacima.

**Poglavlje 3** daje pregled teorijske osnove vezane uz sustave za upravljanje tijekom rada. Objašnjene su različite notacije za modeliranje i prikaz poslovnih procesa te je učinjena njihova usporedba. Objašnjen je i pojam transakcijskoga sustava za upravljanje tijekom rada. Prikazan je pregled stanja istraživanja u području prostornih sustava za upravljanje tijekom rada te objašnjen pojam učinkovitosti upravljanja poslovnim procesima u WFMS-u.

**Poglavlje 4** daje konceptualni model dinamičke komponente sustava za upravljanje zemljištem. Model podataka se temelji na konceptu WFMS-a s dva osnovna zahtjeva: treba podržavati transakcijske koncepte potrebne za rad nad katastarskim podacima i podržavati konkurentne procese. Razvijena su dva modela: općeniti i prošireni. Prošireni model omogućava podršku za procese nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Upravljanje konkurentnim procesima je ostvareno definicijom pesimističnog i optimističnoga tijeka rada. U model podataka su uvedene i međne točke koje omogućavaju povećanje konkurentnosti procesa uvođenjem altruističnoga tijeka rada koji omogućava istovremeno pripremanje procesa te razmjenu podataka između procesa čak i prije njihove provedbe. Na kraju je prikazan prijedlog integracije razvijenoga modela WFMS-a u LADM s ciljem provjere primjenjivosti postavljenih koncepata.

**Poglavlje 5** daje kratki pregled arhitekture testnoga sustava. Dan je i pregled slučajeva korištenja kojim je provjerena ispravnost postavljenih hipoteza te primjenjivost razvijenoga konceptualnog modela. Prikazani su primjeri koji ilustriraju prednosti i nedostatke pojedinoga tijeka rada kako bi se mogli izvesti zaključci o primjenjivosti razvijenih modela.

**Poglavlje 6** daje koncizni pregled postignutih rezultata istraživanja, izvorni znanstveni doprinos te zaključke i prijedlog mogućih smjerova daljnjih istraživanja.

DIO I:

## TEORETSKA OSNOVA I PREGLED DOSADAŠNJIH ISTRAŽIVANJA

*U ovom je dijelu rada objašnjena teoretska osnova i terminologija koja se koristi dalje u radu te su predstavljeni dosezi postojećih istraživanja. Pregled dosadašnjih istraživanja obuhvaća područja baza podataka, transakcijskih modela, upravljanja konkurentnim transakcijama te sustava za upravljanje tijekom rada. Pregledom dosadašnjih istraživanja su prepoznati nedostaci dosadašnjih pristupa kod modeliranja procesa nad katastarskim česticama čija je prostorna sastavnica predstavljena poligonima.*

## 2. Baze podataka i transakcijski modeli

Baza podataka skup je međusobno povezanih podataka, pohranjenih u memoriji računala. Sustav za upravljanje bazama podataka (SUBP) (eng. *Database Management System (DBMS)*) poslužitelj je baze podataka. Podaci u bazi podataka su organizirani u skladu s određenim modelom podataka koji predstavlja skup pravila koja određuju logičku strukturu baze podataka. Sustav za upravljanje bazom podataka oblikuje fizički prikaz baze podataka u skladu s traženom logičkom strukturom. Dosadašnji sustavi za upravljanje bazama podataka podržavaju neke od sljedećih modela:

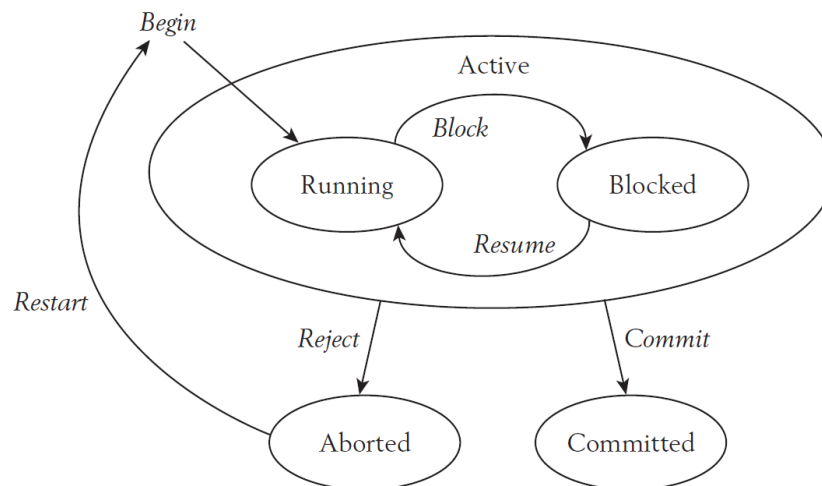
- Mrežni. Baza podataka predočena je usmjerenim grafom. Čvorovi su tipovi zapisa, a lukovi definiraju veze među tipovima zapisa.
- Hijerarhijski. Specijalni slučaj mrežnoga modela. Baza podataka predočena je jednim stablom ili skupom stabala. Čvorovi su tipovi zapisa, a hijerarhijski odnos "nadređeni-podređeni" izražava veze među tipovima zapisa.
- Relacijski. Temelji se na relacijskoj algebri (Codd, 1970).
- Objektno-relacijski. Proširenje relacijskoga modela koji predstavlja pokušaj spajanja najboljeg iz relacijskog i objektno-orijentiranoga pristupa.
- Objektni. Baza podataka skup je perzistentnih objekata koji se sastoje od atributa i metoda (operacija) za rukovanje tim podacima. Svaki je objekt primjerak određene klase, a između klasa se uspostavljaju veze.

Hijerarhijski i mrežni model upotrebljavali su se 1960-ih i 1970-ih godina. Od 1980-ih do današnjih dana prevladava relacijski, odnosno objektno-relacijski model. Objektni model do danas nije zaživio, ali u proteklih 10-ak godina NoSQL baze podataka (eng. *Not Only SQL (NoSQL) database*) (Evans, 2009a,b; Tauro i dr., 2012) dobivaju na popularnosti zbog mogućnosti uvođenja fleksibilnije strukture i manje strožih pravila vezanih uz integritet podataka. Budući da se objektno-relacijski model često koristi za pohranjivanje prostorne sastavnice katastarskih podataka te ima jasnu formalnu osnovu, u ovome su doktorskom radu podaci modelirani i pohranjeni unutar objektno-relacijskoga sustava za upravljanje bazama podataka.

## 2.1. Osnovni transakcijski model

Relacijske baze podataka temelje se na relacijskoj algebri kao što je objašnjeno u Codd (1970). Sustav za upravljanje bazom podataka upravlja podacima pomoću transakcija. U nastavku se definira pojam transakcije u smislu relacijskih baza podataka te načini osiguravanja njezine konzistentnosti (eng. *consistency*), trajnosti (eng. *durability*) i mogućnosti oporavka (eng. *recoverability*). Oporavak transakcije može biti u smjeru natrag ili naprijed. Oporavak-natrag (eng. *backward-recoverability*) osigurava da će transakcija u slučaju prekida vratiti stanje podataka koje je vrijedilo prije njezinoga početka (eng. *rollback*), odnosno oporavak-natrag omogućava svojstvo atomičnosti (eng. *atomicity*). Oporavak-naprijed (eng. *forward-recoverability*) omogućava da se pomoću točaka pohranjivanja (eng. *savepoint*) definiraju konzistentna stanja podataka. U slučaju prekida transakcija se nastavlja od zadnjega konzistentnog stanja. Ovaj se pristup često koristi kod dugih transakcija.

Svaka transakcija ima nekoliko mogućih stanja (Slika 2.1). Između početka i završetka transakcija prolazi kroz različita stanja, a transakcija je ispravna jedino ako su promjene na  $n$ -torkama u potpunosti provedene uspješno.



Slika 2.1: Stanja transakcije (Weikum i Vossen, 2001)

Transakcija postaje aktivna s pokretanjem operacija i označava početno stanje transakcije. Djelomično provedena transakcija označava stanje transakcije od trenutka kada se počnu pokretati operacije do završetka posljednje operacije. Tek se nakon završene posljednje operacije provjerava ispravnost operacija u odnosu na postavljene uvjete ispravnosti baze podataka. Nakon što se ispravnost provjeri, moguća su dva ishoda: u slučaju ispravne transakcije se stanje transakcije mijenja u provedenu u potpunosti (eng. *committed*); inače se stanje transakcije označava kao odbačeno (eng. *aborted*). Transakcije provedene u potpunosti mogu se uspješno završiti, a odbačene se transakcije otkazuju i na taj način završavaju.

### 2.1.1. Definicija transakcije

Transakcija predstavlja osnovnu jedinicu rada unutar sustava za upravljanje bazama podataka te obuhvaća skup operacija koje transformiraju skup podataka iz jednoga konzistentnog stanja u drugo u skladu s ACID svojstvima definiranim u (Härder i Reuter, 1983). Transakcija se može sastojati od skupa operacija čitanja (dohvaćanja podataka) i/ili operacija pisanja (zapisivanje, ažuriranje ili brisanje). Transakcije koje samo dohvaćaju ne sadrže operacije pisanja, ali svejedno transformiraju podatke iz oblika u kojemu su zapisani u oblik definiran upitom kojim se podaci dohvaćaju. Operacije se mogu definirati pomoću jezika za pisanje upita Structured Query Language (SQL) (ISO, 2016). Härder i Reuter (1983) definiraju transakciju kao transformaciju podataka koja ima sljedeća svojstva:

- Atomičnost (eng. *Atomicity*)
- Konzistentnost (eng. *Consistency*)
- Izolacija (eng. *Isolation*)
- Trajnost (eng. *Durability*)

Atomičnost transakcije je svojstvo koje zahtijeva da se transakcija provede u potpunosti (sve ili ništa). Ako se jedna operacija transakcije ne uspije provesti, stanje podataka ostaje nepromijenjeno. Navedeno svojstvo vrijedi i u slučajevima poput pada sustava prilikom nestanka struje ili drugih utjecaja. Konzistentnost transakcije je najvažnije svojstvo transakcije koje osigurava da će transformacija podataka unutar transakcije biti iz jednoga ispravnog stanja u drugo. Konzistentnost osigurava da će zapisani podaci biti u skladu s definiranim pravilima, ograničenjima ili pravilima referencijalnog integriteta. Ostala svojstva transakcije pomažu osigurati konzistentnost transakcije. SUBP posjeduje osnovne mehanizme za osiguravanje konzistentnosti, a ako je potrebno korisnik može definirati dodatna pravila za osiguravanje konzistentnosti podataka na razini  $n$ -torke, odnosno retka. Izolacija je svojstvo koje osigurava da provedba konkurentnih transakcija ima isti učinak kao i sekvencijalna provedba transakcija. Konkurentna transakcija je svaka aktivna transakcija koja pristupa  $n$ -torkama u trenutku  $t_1$  kojima je ranije pristupila druga transakcija koja je i dalje aktivna u trenutku  $t_1$ . Osiguravanje izolacije je glavna zadaća protokola za upravljanje konkurentnim transakcijama. Izolacija je svojstvo koje je važno, ali se uz svojstvo atomičnosti najčešće "opušta" sa svrhom povećanja konkurentnosti transakcija. Trajnost je svojstvo koje osigurava da promjene koje je transakcija učinila trajno ostanu pohranjene nakon što transakcija uspješno završi. Čak i u slučaju nestanka struje, pada sustava ili drugoga vanjskog utjecaja odmah po završetku transakcije podaci moraju ostati pohranjeni. Ako se takav prekid dogodi tijekom izvođenja transakcije, svojstvo atomičnosti osigurava da se promjene neće djelomično zapisati. Jednom kada se transakcija provede nije moguće poništiti njezin učinak. Jedini je način da se to učini jest kroz drugu transakciju. Takve se transakcije nazivaju kompenzacijske transakcije (eng. *compensating transaction*).

Kako bi se spriječilo uvođenje nekonzistentnosti u podatke uslijed konkurentnoga pristupa podacima (poput problema izgubljene vrijednosti ili čitanja neprovedenih podataka) razvijeni su protokoli za upravljanje konkurentnim transakcijama. Povećanjem konkurentnosti transakcija se povećava učinkovitost upravljanja transakcijama. Kako bi se omogućila veća konkurentnost transakcija (odnosno učinkovitost upravljanja transakcijama), poželjno je da transakcije budu što kraćega trajanja te da se zaključava što manji skup podataka. Povećanjem broja zaključanih  $n$ -torki ili dužim trajanjem transakcije, povećava se i mogućnost za pojavu zastoja (eng. *deadlock*). Zastoj nastaje kada transakcija  $t_1$  zaključa  $n$ -torku  $a_1$ , ali treba dohvatiti i  $n$ -torku  $a_2$  koja je zaključana u transakciji  $t_2$ , koja uz  $a_2$  treba i  $a_1$ . Na ovaj način jedna transakcija čeka drugu da otključa  $n$ -torku, ali se nijedna transakcija ne može završiti. Da ovakve situacije ne bi trajale vječno, postoje razni algoritmi za otkrivanje i sprječavanje zastoja koji se temelje na blokiranju ili ponovnome pokretanju transakcije.

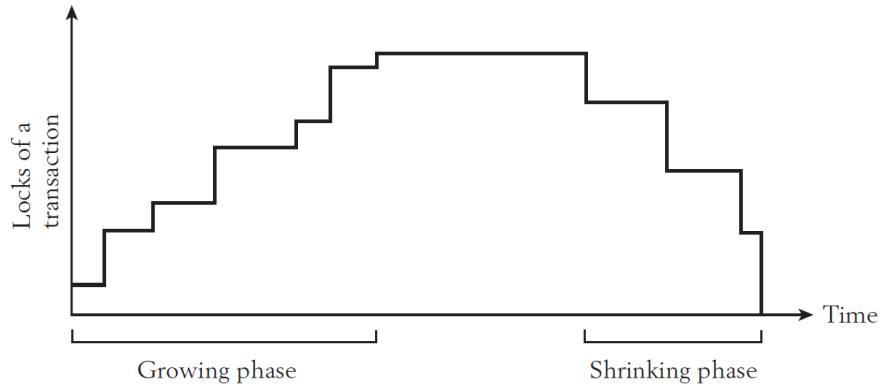
### 2.1.2. Pesimistični protokoli za upravljanje konkurentnim transakcijama

Kod osnovnoga transakcijskog modela se ispravnost transakcije prilikom konkurentnoga pristupa istim  $n$ -torkama postiže primjenom određenoga protokola za upravljanje konkurentnim transakcijama koji se općenito mogu podijeliti na pesimistične i optimistične. Svrha je tih protokola osiguravanje izolacije transakcija, a razlikuju se prema trenutku detektiranja konflikta i načinu njegova rješavanja. Pesimistični protokoli pretpostavljaju najgoru mogućnost – da će više transakcija pokušati ažurirati istu  $n$ -torku u isto vrijeme. S druge strane, optimistični pristup pretpostavlja da su konflikti rijetki te provjerava postoje li konkurentne transakcije koje su pristupile istim  $n$ -torkama neposredno prije provedbe transakcije. Ako ne postoje konkurentne transakcije, promjene se zapisuju.

#### Zaključavanje u dvije faze

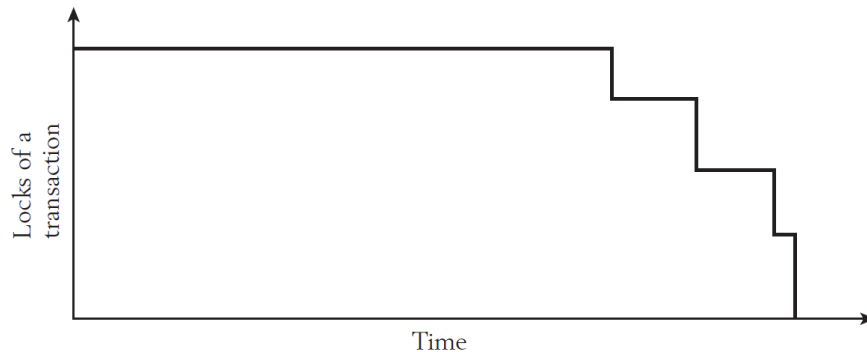
Zaključavanje u dvije faze (eng. *Two-phase locking (2PL)*) protokol je koji se najčešće upotrebljava u sustavima za upravljanje bazama podataka (Slika 2.2). Protokol zaključavanja u dvije faze sastoji se od dvije jasno odvojene i uzastopne faze:

- Rast broja zaključanih  $n$ -torki (eng. *growing phase*):  $n$ -torke se zaključavaju (broj zaključanih  $n$ -torki može samo rasti, odnosno  $n$ -torke se u ovoj fazi ne otključavaju).
- Smanjivanje broja zaključanih  $n$ -torki (eng. *shrinking phase*):  $n$ -torke se otključavaju. U ovoj se fazi ne mogu zaključavati nove  $n$ -torke.



Slika 2.2: Zaključavanje u dvije faze (Weikum i Vossen, 2001)

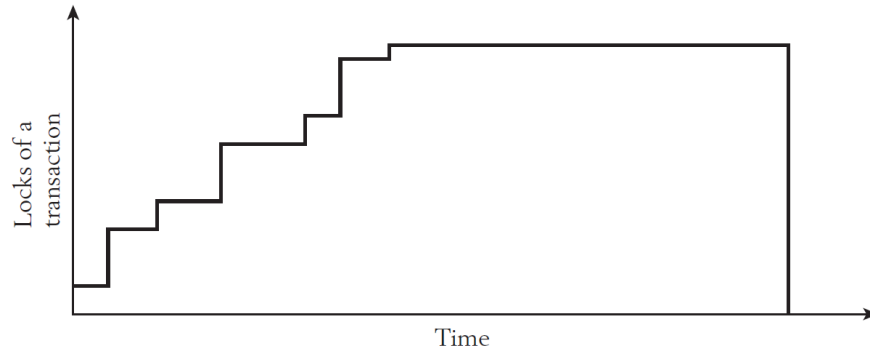
Razne varijacije 2PL protokola omogućavaju da se ove dvije faze međusobno preklapaju. Konzervativno zaključavanje u dvije faze (eng. *Conservative two-phase locking (C2PL)*) se razlikuje od standardnoga zaključavanja u dvije faze po tome što se kod C2PL protokola sve  $n$ -torke zaključavaju prije početka transakcije kako bi se osiguralo da druga transakcija ne blokira transakciju ako već drži zaključane neke  $n$ -torke čime se u potpunosti sprječavaju zastoji (Slika 2.3)



Slika 2.3: Konzervativno zaključavanje u dvije faze (Weikum i Vossen, 2001)

Postoje još protokoli strogoga zaključavanja u dvije faze (eng. *Strict two-phase locking (S2PL)*) i jako strogoga zaključavanja u dvije faze (eng. *Strong strict two-phase locking (SS2PL)*) (Slika 2.4). S2PL se razlikuje od 2PL protokola po tome što se zaključavanje čitanja otpušta po završetku prve faze, a zaključavanje pisanja se otpušta tijekom druge faze 2PL protokola. SS2PL se razlikuje od S2PL po tome što se zaključavanja čitanja i pisanja otpuštaju tek nakon završetka druge faze 2PL protokola, odnosno po završetku transakcije.

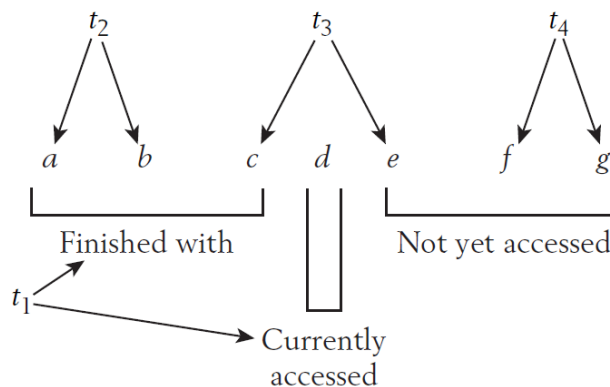




Slika 2.4: Strogo zaključavanje u dvije faze (Weikum i Vossen, 2001)

### Altruistično zaključavanje

Altruistično zaključavanje (eng. *Altruistic locking (AL)*) proširenje je 2PL protokola namijenjeno za korištenje kod transakcija koje obavljaju veliki broj operacija čitanja i pisanja na većemu skupu podataka (Slika 2.5). Takve transakcije uglavnom traju duže nego većina ostalih transakcija i korištenje 2PL protokola može znatno utjecati na konkurentnost transakcija. Glavna je ideja altruističnoga zaključavanja da kratke transakcije pristupaju podskupovima podataka koje čita ili zapisuje duga transakcija. Za razliku od 2PL protokola, AL protokol omogućava da određeni podatak bude zaključan u više transakcija pod određenim uvjetima.



Slika 2.5: Altruistično zaključavanje (Weikum i Vossen, 2001)

AL protokol osim operacija zaključavanja i otključavanja definira operaciju **doniraj** koja definira kada podatak više nije potreban dugoj transakciji kako bi druga transakcija dobila pravo obavljanja izmjena na podatku. Korištenje doniranja na podatku  $x$  u transakciji  $t_i$  koja je donirala podatak  $x$  (označava se s  $d_i(x)$ ) mora poštovati određena pravila (Weikum i Vossen, 2001):

AL1: Podatak  $x$  ne može biti pročitan ili zapisan u transakciji  $t_i$  nakon što ga je donirala; odnosno ako se  $d_i(x)$  i  $o_i(x)$  događaju prema rasporedu  $s$ , a operacija  $o$  je

operacija čitanja ili pisanja ( $o \in \{r, w\}$ ), tada mora vrijediti  $o_i(x) < d_i(x)$ .

AL2: Donirane  $n$ -torke s vremenom se otključavaju ( $ou_i(x)$ ); odnosno ako se  $d_i(x)$  događa prema rasporedu  $s$  iza koje slijedi operacija  $o_i(x)$ , tada je i  $ou_i(x)$  također u rasporedu  $s$  i vrijedi  $d_i(x) < ou_i(x)$ .

AL3: Transakcije ne mogu držati konfliktne ključeve istovremeno, osim ako podatak nije ranije doniran; odnosno, ako su  $o_i(x)$  i  $p_j(x)$  (vrijedi  $i \neq j$ ) konfliktne operacije prema rasporedu  $s$  i vrijedi  $o_i(x) < p_l(x)$ , tada je moguće ili  $ou_i(x) < pl_j(x)$  ili je  $d_i(x)$  također u rasporedu  $s$  i vrijedi  $d_i(x) < pl_j(x)$ .

Ako transakcija  $t_j$  zaključa podatak koji joj je doniran i nije otključan u transakciji  $t_i$ ,  $i \neq j$ , tada je transakcija  $t_i$  svjesna transakcije  $t_j$  (engleski je termin formuliran obrnuto  $t_j$  in the wake of  $t_i$ ). Formalna definicija glasi:

- Transakcija  $t_i$  je svjesna operacije  $p_j(x)$  transakcije  $t_j$  u rasporedu  $s$  ako  $d_i(x) \in op(s)$  i  $d_i(x) < p_j(x) < ou_i(x)$ .
- Transakcija  $t_i$  je svjesna transakcije  $t_j$  ako je  $t_i$  svjesna bilo koje operacije iz  $t_j$ . Transakcija  $t_i$  je u potpunosti svjesna transakcije  $t_j$  ako je  $t_i$  svjesna svih operacija iz  $t_j$ .
- Transakcija  $t_j$  je zadužena (eng. *indebted*) kod transakcija  $t_i$  u rasporedu  $s$  ako je  $t_i$  svjesna operacije  $p_j(x)$ . Uz to su  $o_i(x)$  i  $p_j(x)$  u konfliktu ili je treća operacija  $q_k(x)$ , takva da vrijedi  $d_i(x) < q_k(x) < p_j(x)$  u konfliktu s obje operacije  $o_i(x)$  i  $p_j(x)$ .

Ovi su pojmovi osnova za sljedeće pravilo AL protokola:

AL4: Kada je transakcija  $t_j$  zadužena kod druge transakcije  $t_i$ ,  $t_i$  mora ostati svjesna transakcije  $t_j$  dok ne počne otključavati podatke. Odnosno za svaku operaciju  $p_j(x)$  koja se događa u rasporedu  $s$   $t_i$  je svjesna operacije  $p_j(x)$  ili postoji operacija otključavanja  $ou_i(y)$  u rasporedu  $s$  takva da vrijedi  $ou_i(y) < o_j(x)$ .

Dakle, ako transakcija  $t_i$  donira pravo rada na podatku transakciji  $t_j$ , tada transakcija  $t_i$  ne može otključati podatke, odnosno ne može se završiti dok transakcija  $t_j$  ne otključa podatak za koji joj je donirano pravo izmjene. Time se osigurava serijalizabilnost transakcija. Primjena koncepta altruističnoga zaključavanja na katastarske podatke je objašnjena u (Daniilidis, 2006), ali autor ne daje detalje o načinu primjene ili tehničke implementacije.

### Sortiranje temporalnih atributa

Sortiranje temporalnih atributa (eng. *Timestamp ordering (TO)*) osigurava serijalizabilnost dodjeljivanjem temporalnih atributa transakcijama umjesto zaključavanjem  $n$ -torki. Kod sortiranja temporalnih atributa svakoj se transakciji  $t_i$  dodjeljuje temporalni atribut  $ts(t_i)$  koja označava početak transakcije. Svaka operacija unutar transakcije preuzima temporalni atribut  $ts(t_i)$ . Konflikti se rješavaju prema redoslijedu temporalnih atributa. Glavno pravilo glasi: ako su  $p_i(x)$  i  $q_j(x)$ ,  $i \neq j$  konfliktne operacije, tada vrijedi da se  $p_i(x)$  obavlja

prije  $q_j(x)$  ako i samo ako vrijedi  $ts(t_i) < ts(t_j)$ . Operacija  $p_i$  se odbija ako dođe na red prekasno, odnosno ako već postoji konfliktna operacija  $q_j$  za koju vrijedi  $i \neq j$  i  $ts(t_j) < ts(t_i)$ . Ako se operacija odbije, čitava se transakcija odbija zbog svojstva atomičnosti.

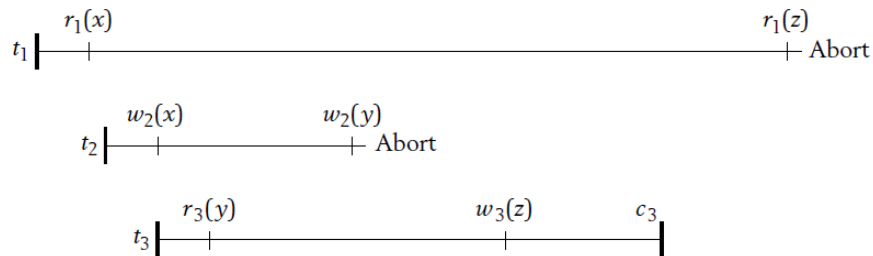
Kako bi se odredilo koja je transakcija došla na red prekasno, potrebno je bilježiti sljedeće temporalne attribute za svaki podatak  $x$ :

- $maxRTS(x)$ : vrijednost najnovijega temporalnog atributa operacije čitanja koja je pokrenuta,
- $maxWTS(x)$ : vrijednost najnovijega temporalnog atributa operacije pisanja pokrenuta.

Osnovni protokol sortiranja temporalnih atributa (eng. *Basic timestamp ordering (BTO)*) uspoređuje navedene dvije vrijednosti sa svakom transakcijom kako bi odredio prihvaćati se operacija:

- operacija čitanja  $r_i(x)$ : ako je  $ts(t_i) < maxWTS(x)$  tada se operacija odbacuje, u suprotnome ( $ts(t_i) \geq maxWTS(x)$ ) operacija čitanja  $r_i(x)$  se postavlja u red pokretanja i postavlja se  $maxRTS(x) = max(maxRTS(x), ts(t_i))$ ,
- operacija pisanja  $w_i(x)$ : ako je  $ts(t_i) < maxRTS(x)$  i ako je  $ts(t_i) < maxWTS(x)$  tada se operacija odbacuje, u suprotnome se  $w_i(x)$  postavlja u red pokretanja i postavlja se  $maxWTS(x) = max(maxWTS(x), ts(t_i))$ .

Slika 2.6 prikazuje primjer konkurentnih transakcija koje se provode prema BTO protokolu. Transakcije  $t_1$  i  $t_2$  se odbacuju, jer njihove operacije dolaze prekasno na provedbu, odnosno transakcija  $t_2$  se odbacuje zato što za operaciju pisanja  $w_2(y)$  vrijedi  $ts(t_2) < maxRTS(y)$ , a transakcija  $t_1$  se odbacuje jer za operaciju čitanja  $r_1(z)$  vrijedi  $ts(t_1) < maxWTS(z)$ .



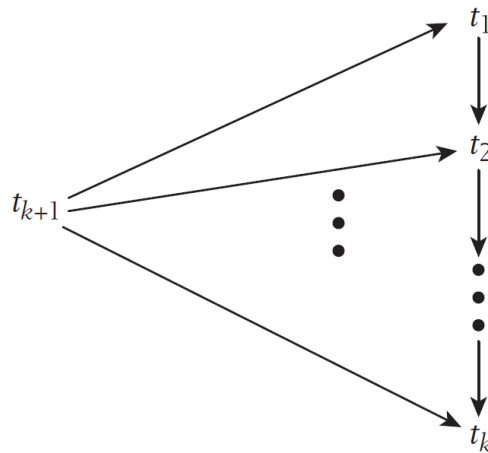
Slika 2.6: Provedba transakcija u skladu s BTO protokolom

Prednost je BTO protokola što osigurava serijalizabilnost kao i razne varijante 2PL protokola, a transakcije ne moraju čekati jedna drugu što znači da ne postoji mogućnost pojave zastoja. Nedostatak je što ne osigurava oporavak transakcije (eng. *recoverability*) zato što se dopušta čitanje podataka od transakcija koje nisu provedene. Ovaj nedostatak uklanja protokol strogo sortiranja temporalnih atributa (eng. *Strict timestamp ordering*

(STO)) koji zahtijeva da se operacije obavljaju samo na važećim  $n$ -torkama. Drugi je nedostatak BTO protokola mogućnost da ista transakcija bude više puta odbačena i ponovno pokrenuta zbog konkurentnih transakcija.

### Provjera grafa serijalizabilnosti

Provjera grafa serijalizabilnosti (eng. *Serializability graph tester (SGT)*) temelji se na činjenici da se povijest može prikazati grafom te da taj graf mora biti aciklički. SGT protokol održava graf serijalizabilnosti koji prikazuje transakcije i njihovu međusobnu zavisnost. Serijalizabilnost se održava na način da se osigurava da graf bude aciklički u svakome trenutku.



Slika 2.7: Graf serijalizabilnosti (Weikum i Vossen, 2001)

Slika 2.7 prikazuje povijest kao što je navedeno ispod:

$$s = r_{k+1}(x) \underbrace{w_1(x)w_1(y_1)c_1}_{\text{operacije } t_1} \underbrace{w_2(x)w_2(y_2)c_2 \dots}_{\text{operacije } t_2} \underbrace{w_k(x)w_k(y_k)c_k \dots}_{\text{operacije } t_k} \dots$$

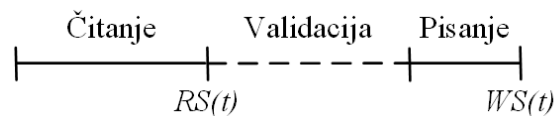
Za primjer uzmimo da je sljedeća operacija u transakciji  $t_{k+1}$  operacija pisanja  $w_{k+1}(z)$ . Transakcija  $t_{k+1}$  se može provesti ako vrijedi  $z \notin \{x, y_1, \dots, y_k\}$ , a u suprotnome se odbija. Da bi se graf serijalizabilnosti mogao testirati, potrebno je dohvatiti informacije o transakcijama  $t_i$ ,  $1 \leq i \leq k$  od kojih neke mogu biti već ranije provedene.

Iako je SGT protokol intuitivan i atraktivan s teoretskoga gledišta, nije praktičan za implementaciju. Najveći implementacijski izazov predstavlja pohranjivanje grafa serijalizabilnosti. Prostor za pohranu grafa serijalizabilnosti raste s kvadratom broja relevantnih transakcija od kojih neke ne moraju biti niti aktivne.

### 2.1.3. Optimistični protokoli za upravljanje konkurentnim transakcijama

Optimistični protokoli pretpostavljaju da se konflikti rijetko događaju. Primjerice, kod baze podataka koja sadrži katalog proizvoda i 99 % transakcija samo čita podatke o cijena i opisima proizvoda. Povremeno se ažuriraju cijene ili dodaju novi proizvodi, ali se to događa rijetko u odnosu na učestalost čitanja podataka. U takvome bi slučaju 2PL protokol bio previše restriktivan jer bi se  $n$ -torke nepotrebno zaključavale što znači da se smanjuje konkurentnost transakcija, odnosno nepotrebno se troše resursi baze podataka.

Optimistični protokoli odgađaju provjeru serijalizabilnosti transakcije do trenutka kada je transakcija spremna za provedbu. Zato optimistični protokoli uglavnom sadrže dvije ili tri faze transakcije (Slika 2.8) (Kung i Robinson, 1981). Kod transakcija koje čitaju podatke postoje dvije faze. Faza čitanja nikada ne može prouzročiti konflikt i ne provjerava se, ali prije vraćanja rezultata potrebno je provjeriti rezultat zbog mogućih konfliktnih situacija. Svaka transakcija radi promjene na privatnim kopijama koje druge transakcije ne vide. Transakcija koja je spremna za provedbu validira se, a označava se ispravnom ako je transakcija serijalizabilna, nakon čega slijedi faza pisanja u kojoj privatna kopija zamjenjuje izvornu  $n$ -torku. Ako je transakcija neispravna, odbija se. Nakon provedbe transakcije promjene su vidljive svim ostalim transakcijama.



Slika 2.8: Tri faze transakcije kod optimističnih protokola

U nastavku se navode dvije različite vrste validacije transakcije kod optimističnih protokola (Kung i Robinson, 1981):

- SUBP provjerava konfliktnost transakcije sa svim transakcijama koje su već provedene (eng. *Backward-oriented optimistic concurrency control (BOCC)*).
- SUBP provjerava konfliktnost transakcije sa svim transakcijama koje se izvode paralelno, ali su još uvijek u fazi čitanja (eng. *Forward-oriented optimistic concurrency control (FOCC)*).

Kod BOCC validacije, transakcija  $t_j$  se označava pozitivno validiranom ako za svaku već provedenu transakciju  $t_i$  vrijedi jedno od sljedećih pravila:

- $t_i$  je završila prije nego što je  $t_j$  započela
- $RS(t_j) \cap WS(t_i) = \emptyset$  i faza pisanja transakcije  $t_i$  završava prije nego je započela faza pisanja transakcije  $t_j$

Između čitanja i validacije transakcije može proći duže vrijeme što znači da redosljed operacija dviju transakcija, kao što je navedeno ispod, može rezultirati odbijanjem transakcije  $t_j$ :

$$\dots r_j(x) \dots w_i(x) \dots \dots \dots \text{validacija}(t_j)$$

Zbog toga FOCC validira transakciju  $t_j$  sa svim konkurentnim transakcijama  $t_i$  koje su u fazi čitanja. Završetak operacije čitanja transakcije  $t_i$  u trenutku  $n$  označimo s  $RS^n(t_i)$ . Tada se  $t_j$  prema FOCC validaciji označava ispravnom u vremenu  $n$  ako sljedeće pravilo vrijedi za sve transakcije  $t_i$  koje su još uvijek u fazi čitanja u trenutku  $n$ :

$$WS(t_j) \cap RS^n(t_i) = \emptyset$$

Drugim riječima, ovaj se uvjet može definirati na sljedeći način: transakcija  $t_j$  označava se ispravnom ako se skup operacija pisanja ( $WS(t_j)$ ) ne preklapa sa skupom operacija čitanja konkurentnih transakcija  $t_i$  u trenutku  $n$ . Ovaj je uvjet automatski zadovoljen kod transakcija koje samo čitaju podatke.

#### 2.1.4. Upravljanje konkurentnim transakcijama verzioniranjem

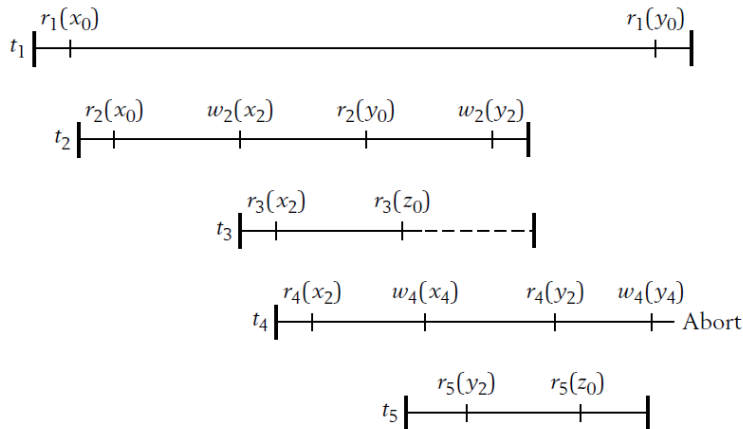
U ovome su odjeljku objašnjeni protokoli za upravljanje konkurentnim transakcijama verzioniranjem koji predstavljaju proširenja protokola navedenih ranije u ovome poglavlju kako bi bili prikladni za višestruke verzije. Verzioniranje se uvodi kao jedan od mogućih načina za povećanje konkurentnosti transakcija i krajnji korisnik nema dojam da se radi o više verzija, jer se verzije stvaraju automatski u okviru operacija unutar transakcije od strane sustava za upravljanje bazom podataka.

##### MVTO protokol

Protokol za sortiranje temporalnih atributa kod višestrukih verzija (eng. *Multiversion timestamp ordering (MVTO)*) obrađuje operacije po principu prvi-unutra-prvi-van (eng. *First-in-first-out (FIFO)*). Operacije nad  $n$ -torkama se transformiraju u operacije nad verzijama  $n$ -torki i obrađuju se na način da rezultat izgleda kao da su operacije napravljene na jednoj verziji  $n$ -torke. Svaka verzija nosi temporalni atribut  $ts(t_i)$  transakcije  $t_i$  u okviru koje je stvorena. MVTO protokol djeluje u skladu sa sljedećim pravilima:

- Operacija  $r_i(x)$  transformira se u operaciju  $r_i(x_k)$ , gdje  $x_k$  označava verziju  $n$ -torke  $x$  koja nosi najnoviji temporalni atribut manji od  $ts(t_i)$  te je zapisan od strane transakcije  $t_k$ ,  $k \neq i$ .
- Operacija  $w_i(x)$  obrađuje se na sljedeći način:  
Ako je operacija  $r_j(x_k)$  već stavljena u red pokretanja i vrijedi  $ts(t_k) < ts(t_i) < ts(t_j)$ , tada se  $w_i(x)$  odbija i  $t_i$  se otkazuje, a u suprotnome se  $w_i(x)$  transformira u  $w_i(x_i)$  i završava izvođenje.

- Provedba se transakcije  $c_i$  odgađa dok se ne provedu sve transakcije  $t_j$  koje su kreirale nove verzije podataka koje čita transakcija  $t_i$ . Ovaj dio protokola nije obavezan i pridržavanje ovoga pravila osigurava oporavak transakcije, odnosno izbjegavanje raznih anomalija poput čitanja neprovedenih podataka.



Slika 2.9: MVTO protokol (Weikum i Vossen, 2001)

Slika 2.9 prikazuje transakcije koje se istovremeno provode nad istim  $n$ -torkama. Korištenje MVTO protokola za upravljanje konkurentnim transakcijama predstavlja poboljšanje u odnosu na BTO protokol. Transakcija  $t_1$  provodi se paralelno s transakcijom  $t_2$ , jer čita verziju  $y_0$ , a transakcija  $t_2$  stvara novu verziju  $y_2$ . Transakcija  $t_3$  mora čekati da završi transakcija  $t_2$ , jer je  $t_3$  pročitala verziju  $x_2$ . Transakcija  $t_4$  mora biti odbačena, jer stvara novu verziju  $y_4$ , a transakcija  $t_2$  je stvorila ranije verziju  $y_2$ .

### MV2PL protokol

Protokol zaključavanja u dvije faze kod višestrukih verzija (eng. *Multiversion two-phase locking (MV2PL)*) temelji se na SS2PL protokolu. Općenito se razlikuju:

- provedene verzije predstavljaju sve podatke koji su zapisani u transakcijama koje su već provedene,
- trenutne verzije predstavljaju verzije podataka koje su nastale u transakciji koja je zadnja provedena nad njima,
- neprovedene verzije predstavljaju sve ostale verzije (stvorene od aktivnih transakcija).

Zadnja se operacija unutar transakcije tretira različito od ostalih, jer prethodi ili jest operacija provedbe transakcije. Svaka se operacija u transakciji tretira na sljedeći način:

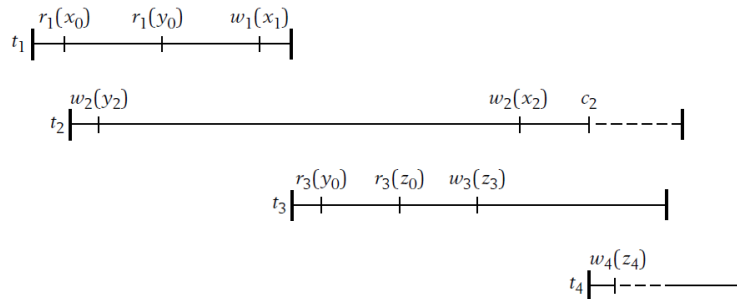
- Ako operacija nije posljednja u transakciji:

$r(x)$  se provodi odmah na posljednju provedenu verziju podatka  $x$  ili na neprovedenoj verziji  $x$  (ovisno o protokolu);

$w(x)$  se provodi samo u slučaju kada transakcija koja je posljednja zapisala  $x$  završena, odnosno kada nema drugih neprovedenih verzija.

- Ako je operacija posljednja u transakciji  $t_i$ , odgađa se dok sljedeće vrste transakcija nisu provedene:
  - sve transakcije  $t_j$  koje čitaju trenutnu verziju podatka zapisanu od transakcije  $t_i$ ,
  - sve transakcije  $t_j$  od kojih transakcije  $t_i$  čitaju neku verziju.

Zbog ograničenih resursa sustava za upravljanje bazama podataka koji ne mogu podržati beskonačni broj verzija, razvijen je protokol zaključavanja u dvije faze temeljen na dvije verzije (eng. *Two-version 2PL (2V2PL)*) koji predstavlja specijalizaciju MV2PL protokola. Dvije su verzije koje se čuvaju za vrijeme trajanja transakcije: vrijednost koju je podatak imao prije transakcije i ona koju će imati nakon transakcije. Zbog ovakvog ograničenja transakcije koje čitaju podatke imaju više koristi od 2V2PL protokola. Transakcije čitaju uvijek trenutnu verziju, odnosno zadnju provedenu verziju. Ipak, kod 2V2PL postoje tri različite vrste zaključavanja (čitanje, pisanje, provedba) podataka koje omogućavaju veću konkurentnost transakcija u odnosu na klasični 2PL protokol.



Slika 2.10: 2V2PL protokol (Weikum i Vossen, 2001)

Slika 2.10 prikazuje primjer provedbe konkurentnih transakcija prema 2V2PL protokolu. Transakcija  $t_3$  čita vrijednost  $y$  ( $y$  zaključan za čitanje) te transakcija  $t_2$  mora čekati kako bi se završila transakcija  $t_3$ . Transakcija  $t_4$  mora pričekati završetak transakcije  $t_3$  zbog ograničenja protokola o postojanju najviše dvije verzije istoga podatka.

### 2.1.5. Izolacija transakcije

Izolacija transakcije svojstvo je koje osigurava da promjene koje transakcija napravi nisu vidljive drugim transakcijama. Izolacija se osigurava protokolima za upravljanje konkurentnim transakcijama. Od svih se ACID svojstava, izolacija najčešće opušta s ciljem povećanja konkurentnosti transakcija. Zbog toga SQL norma (ISO, 2016) definira više razina izolacije transakcije koje su ustvari kontrolirane devijacije (S)S2PL protokola:



- čitanje neprovedenih  $n$ -torki (eng. *read uncommitted*),
- čitanje provedenih  $n$ -torki (eng. *read committed*),
- serijalizabilnost (eng. *conflict serializability*).

Navedene se tri razine izolacije razlikuju po trajanju zaključanosti čitanja i pisanja. Čitanje neprovedenih  $n$ -torki ne zahtijeva zaključavanje čitanja i može se koristiti s ciljem pregledavanja ili statističkih procjena gdje konzistentni pogled na podatke nije potreban (primjerice, promjena cijena nekoliko proizvoda ima mali utjecaj na računanje prosječne vrijednosti cijene nekoliko tisuća proizvoda). Kod ove razine izolacije dugo trajanje zaključanosti pisanja ne može osigurati ispravnost transakcije, jer transakcija ovisi o čitanju podatka koji može biti neispravan što dovodi do problema čitanja neprovedenih podataka.

Čitanje provedenih  $n$ -torki sprječava ovakve probleme do neke mjere, posebice eliminira problem čitanja neprovedenih  $n$ -torki. Ovu razinu karakterizira kratko trajanje zaključanosti čitanja i duga zaključanost pisanja. Ova razina izolacije vrlo dobro reducira konflikte, posebno između dugih transakcija čitanja i kratkih transakcija pisanja. Zbog toga se često koristi u sustavima za upravljanje bazama podataka. Ova je razina izolacije zadana (eng. *default*) u PostgreSQL sustavu za upravljanje bazama podataka<sup>1</sup> kao pretpostavljena.

Treća se razina izolacije prema SQL standardu dijeli na dvije podrazine, potpunu i opuštenu serijalizabilnost. Opuštena se serijalizabilnost još naziva ponovljeno čitanje (eng. *repeatable read*). Jedina anomalija koju ova razina izolacije dopušta jesu fantomski redovi koja se događa ako se dva upita izvrše istovremeno i vrate različiti broj redova (zato što su u međuvremenu redovi dodani ili brisani). Druga podrazina jest potpuna serijalizabilnost koja predstavlja najvišu razinu izolacije i ne dopušta uvođenje nekonzistentnosti u podatke u obliku ranije navedenih anomalija u podacima.

Kod protokola za upravljanje konkurentnim transakcijama verzioniranjem ranije navedene izolacije nisu zadovoljavajuće, jer je definicija konflikta drugačija. Primjeri su razina izolacije koje pretpostavljaju protokole temeljene na verzioniranju čitanje provedenih višestrukih verzija (eng. *Multiversion read committed*) te izolacija pogleda (eng. *Snapshot isolation*). Kod čitanja provedenih višestrukih verzija, transakcija čita najnoviju provedenu verziju podatka. Ako transakcija zapisuje promjene na dohvaćenome podatku, on se zaključava za pisanje do trenutka završetka transakcije. Kod izolacije pogleda svaka operacija čita najnoviju verziju podatka od početka transakcije.

U sustavima za upravljanje bazama podataka na razini transakcije se može podesiti razina izolacije kao što je prikazano dolje na primjeru PostgreSQL SUBP (Ispis koda 2.1).

Ispis koda 2.1: Postavljanje razine izolacije na razini transakcije

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED ;
SELECT * FROM cp_parcel WHERE area > 100 ;
```

---

<sup>1</sup><https://www.postgresql.org/docs/10/static/transaction-iso.html>

## 2.2. Napredni transakcijski modeli

Prethodno objašnjeni osnovni transakcijski model pretpostavlja kratko trajanje transakcije te ima ograničenja koja onemogućuju obavljanje kompleksnih operacija unutar jedne transakcije (Elmagarmid, 1991):

- transakcije ne smiju biti dugoga trajanja,
- transakcije ne mogu biti ugniježdene,
- transakcije ne smiju biti djelomično provedene,
- transakcije ne podržavaju suradnju na aktivnostima,
- transakcije ne podržavaju lokalnu autonomiju,
- transakcije ne podržavaju kontrolu korisnika.

Zbog toga se pojavljuju brojni napredni transakcijski modeli koji nadograđuju osnovni transakcijski model kroz neki od sljedećih segmenata:

- Podrška za transakcije dugoga trajanja (eng. *long-lived transactions*),
- Podrška za transakcije s djelomičnim provođenjem (eng. *open-ended activities*),
- Podrška za ugniježdivanje transakcija (eng. *nested transactions*),
- Podrška za suradnju na aktivnostima,
- Podrška za lokalnu autonomiju,
- Podrška za višekorisnički rad na transakciji.

Chrysanthis i Ramamritham (1992) prezentiraju formalni okvir koji opisuje svojstva proširenih transakcijskih modela kako bi se pomoću njih mogao iskazati bilo koji transakcijski model ili pomoću takvoga formalnog okvira definirati novi (Chrysanthis i Ramamritham, 1994). U radu koriste pojam prošireni transakcijski model, dok Elmagarmid (1991) koristi izraz napredni transakcijski model. Pojam naprednih baza podataka je formiran kako bi se naglasila razlika u odnosu na koncept osnovnoga transakcijskog modela (Barghouti i Kaiser, 1991). Neki su od naprednih transakcijskih modela navedeni u nastavku:

- ugniježdene transakcije (eng. *Nested transactions*) (Moss, 1981),
- SAGA - transakcije dugoga trajanja (Garcia-Molina i Salem, 1987),
- dijeljene transakcije (eng. *Split-transactions*) (Pu i dr., 1988),
- kooperativne transakcije (eng. *Cooperative transaction hierarchies*) (Nodine i Zdonik, 1992),
- interaktivni transakcijski model (eng. *Interactive transaction model*) (Lee i dr., 1991),
- transakcije na više razina (eng. *Multi-level transactions*) (Weikum i Schek, 1991),
- odgodive i nepovezane transakcije (eng. *Deferred and Decoupled Transactions*) (Dayal i dr., 1991),
- politransakcije (eng. *Polytransactions*) (Sheth i dr., 1992),
- ConTract model (Wächter i Reuter, 1992),

- semantičke transakcije (eng. *S-Transactions*) (Holtkamp, 1990),
- fleksibilne transakcije (eng. *Flexible Transactions*) (Kaiser, 1990),
- Tool Kit (Unland i Schlageter, 1991).

Ugniježdene transakcije (Moss, 1981) nastale su skoro u isto vrijeme kada je prezentiran osnovni transakcijski model (Gray, 1981). Ugniježdene transakcije omogućavaju grupiranje transakcija, odnosno operacija transakcije može biti sama transakcija kao što je prikazano u primjeru ispod (Ispis koda 2.2).

### Ispis koda 2.2: Ugniježdena transakcija

```
begin (T1)
  read(x)
  write(y)
  begin (T11)
    read(z)
    write(y)
  end (T11)
  write(z:=x+y)
end (T1)
```

Transakcija najviše razine (u primjeru iznad to je transakcija T1) mora zadržati ACID svojstva, a transakcije niže razine moraju zadržati svojstva atomičnosti i izolacije dok konzistentnost nije obavezna. Promjene postaju vidljive kada se transakcija najviše razine provede. Ugniježdene transakcije mogu sadržavati bilo koji broj podtransakcija s kojima tvori stablo transakcija. Glavna je prednost ovoga modela modularnost, odnosno podtransakcije se mogu složiti tako da transakcija najviše razine obavi kompleksnu operaciju. Uvjeti se odbacivanja čitave transakcije mogu definirati na razini podtransakcije, a podtransakcije se mogu provoditi paralelno (eng. *intra-transaction parallelism*). Otvorene ugniježdene transakcije (Weikum i Schek, 1992) opuštaju svojstvo izolacije tako da dopuštaju da rezultat provedene podtransakcije bude vidljiv drugim konkurentnim podtransakcijama.

SAGA je napredni transakcijski model koji podržava transakcije dugoga trajanja te predstavlja skup relativno nezavisnih podtransakcija  $T_1, T_2, \dots, T_n$  za koje su vezane kompenzirajuće transakcije  $C_1, C_2, \dots, C_{n-1}$ . Sustav jamči da će se provesti skup transakcija  $T_1, T_2, \dots, T_n$  ili  $T_1, T_2, \dots, T_j, C_j, \dots, C_2, C_1$  ( $j < n$ ), odnosno SAGA posjeduje svojstvo atomičnosti. Podtransakcije mogu biti međusobno ovisne na bilo koji način, a pojedina se podtransakcija može provesti bez čekanja na druge transakcije. U slučaju pogreške, SAGA može provesti podtransakcije koje se nisu provele ako je moguće, a u suprotnome se poništavaju promjene svih podtransakcija pokretanjem kompenzirajućih transakcija. SAGA mora imati svojstva atomičnosti i trajnosti. Konzistentnost je ugrožena, jer je moguće da podtransakcije ne gledaju isto konzistentno stanje. Izolacija je reducirana na razinu podtransakcije.

Dijeljene transakcije omogućavaju dijeljenje transakcije na dvije serijalizabilne transakcije kako bi se rasporedili resursi između transakcija (Pu i dr., 1988). Glavna je svrha

dijeljenja transakcije mogućnost provedbe jedne podtransakcije kako bi se otkrili rezultati nadređenoj transakciji. Dijeljene transakcije definiraju uvjete kako bi se osigurala konzistentnost i serijalizabilnost transakcije. Primjerice, ako se transakcija  $T$  dijeli na transakcije  $A$  i  $B$ , uvjeti glase:

- $WS_A \cap WS_B = WriteLast_B$  - osigurava da se  $n$ -torke u skupu  $WS_{last}(B)$  ažuriraju zadnji u transakciji  $B$  što osigurava da  $A$  neće prepisati rezultat od  $B$ ,
- $RS_A \cap WS_B = \emptyset$  - osigurava da  $A$  neće čitati  $n$ -torke od  $B$
- $RS_B \cap WS_A = ShareSet$  - definira da  $B$  smije čitati od  $A$ .

Svi navedeni uvjeti osiguravaju da će  $A$  biti provedena prije  $B$ . Ako su  $WriteLast_B$  i  $ShareSet$  prazni skupovi, tada se  $A$  i  $B$  mogu provesti neovisno jedna o drugoj, a u suprotnome se  $A$  mora provesti prije  $B$ , jer  $B$  ovisi o rezultatu transakcije  $A$ . Prednost je ovoga pristupa reducirana izolacija, odnosno otključavanje resursa koji su rezultat provedene transakcije  $A$ . Ovo ujedno znači da su rezultati vidljivi odmah po provedbi podtransakcije. Druga je prednost što provedba transakcije  $A$  nije ugrožena prekidom podtransakcije  $B$ . Model je predviđen za korištenje kod dugih transakcija s nesigurnim završetkom poput Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM) projekata ili razvoja softvera.

Kooperativne transakcije (Nodine i Zdonik, 1992) opuštaju serijalizabilnost kako bi podržale aktivnosti koje zahtijevaju suradnju poput CAD transakcija (Bancilhon i dr., 1985). Umjesto serijalizabilnosti kod ovoga se modela uvjeti ispravnosti trebaju definirati od strane aplikacije. Grupa transakcija uključuje više kooperativnih transakcija koje predstavljaju određeni zadatak. Uvjeti su ispravnosti definirani protokolima. Unutarnji protokol definira pravila ponašanja svake pojedine transakcije, dok vanjski protokol definira način interakcije između transakcija. Interaktivni transakcijski model (Lee i dr., 1991) razvijen je kako bi se podržali kooperativni zadaci u multimedijalnome telekomunikacijskom okruženju.

Transakcije na više razina (Weikum i Schek, 1992) opuštaju serijalizabilnost s ciljem povećanja konkurentnosti transakcija. Model predstavlja varijantu ugniježđenih transakcija s fiksno definiranom razinom ugniježdivanja. Različite razine u transakcijskome stablu mogu primijeniti različite protokole za upravljanje konkurentnim transakcijama.

Odgodive i nepovezane transakcije (Dayal i dr., 1991) predstavljaju još jednu varijantu ugniježđenih transakcija koje su namijenjene za primjenu u distribuiranim bazama podataka, jer podržavaju paralelizam unutar transakcija (transakcije se ne moraju provoditi sekvencijalno). Provedba se odgodive podtransakcije može odgoditi do samoga kraja transakcije, a podtransakcija može kreirati sasvim novu transakciju koja se naziva nepovezana transakcija. Autori definiraju pojam duge aktivnosti (eng. *long running activity*) koja se sastoji od niza transakcija.

Politransakcije (Sheth i dr., 1992) su napredni transakcijski model koji omogućava ažuriranje podataka koji su međusobno zavisni i pohranjeni u distribuiranim bazama podataka. Međusobna se zavisnost i konzistentnost podataka pohranjuje kao trojka  $\{D, C, A\}$

u shemu zavisnosti između baza podataka (eng. *interdatabase dependency schema (IDS)*).  $D$  označava zavisnu transakciju za određenu transakciju,  $C$  definira kada se zavisna transakcija treba dogoditi, a  $A$  je zavisna transakcija. Politransakcija  $T^+$  za transakciju  $T$  definira se na sljedeći način:

- $T$  se uzima kao transakcija najviše razine u  $T^+$
- provjera IDS-a, generiranje zavisnih transakcija od  $T$  koje se uzimaju kao podtransakcije od  $T$ ,
- za svaku se podtransakciju  $T_i$  dohvaćanju zavisne transakcije koje se tretiraju kao podtransakcije od  $T_i$ ,
- postupak se ponavlja dok se ne dođe do transakcija koje nemaju zavisne transakcije.

Kako bi se održala konzistentnost, sve podtransakcije od  $T^+$  trebaju biti provedene, osim ako se ne radi o nepovezanim transakcijama koje se mogu provesti kasnije (nakon što se  $T$  provede).

ConTract model (Wächter i Reuter, 1992) omogućava grupiranje transakcija u aktivnosti (prema autorima aktivnost i ConTract su sinonimi) koje su u stanju obaviti kompleksne operacije. Redoslijed je transakcija opisan pomoću skripte (eng. *script*), a svaka se transakcija naziva korak (eng. *step*) i predstavlja osnovnu jedinicu rada unutar skripte. Skripta opisuje tijek rada aktivnosti, a može se sastojati od jednostavnih programskih elemenata poput petlji, uvjeta (primjerice, if-then-else), paralelizma (primjerice, for-each petlja). Model omogućava grupiranje i ugnježdavanje transakcija te definiciju ovisnosti između transakcija. ConTract model osigurava oporavak-naprijed pomoću skripti za oporavak koje vraćaju stanje transakcije u stanje prije pada sustava.

Semantičke transakcije (S-transakcije) (Holtkamp, 1990) predstavljaju napredni transakcijski model razvijen kako bi se podržale autonomne operacije u bankarstvu. Globalna je transakcija sastavljena od lokalnih transakcija. Model podržava autonomiju lokalne transakcije, a izolacija globalne transakcije nije podržana pa se oporavak temelji na kompenzirajućim transakcijama.

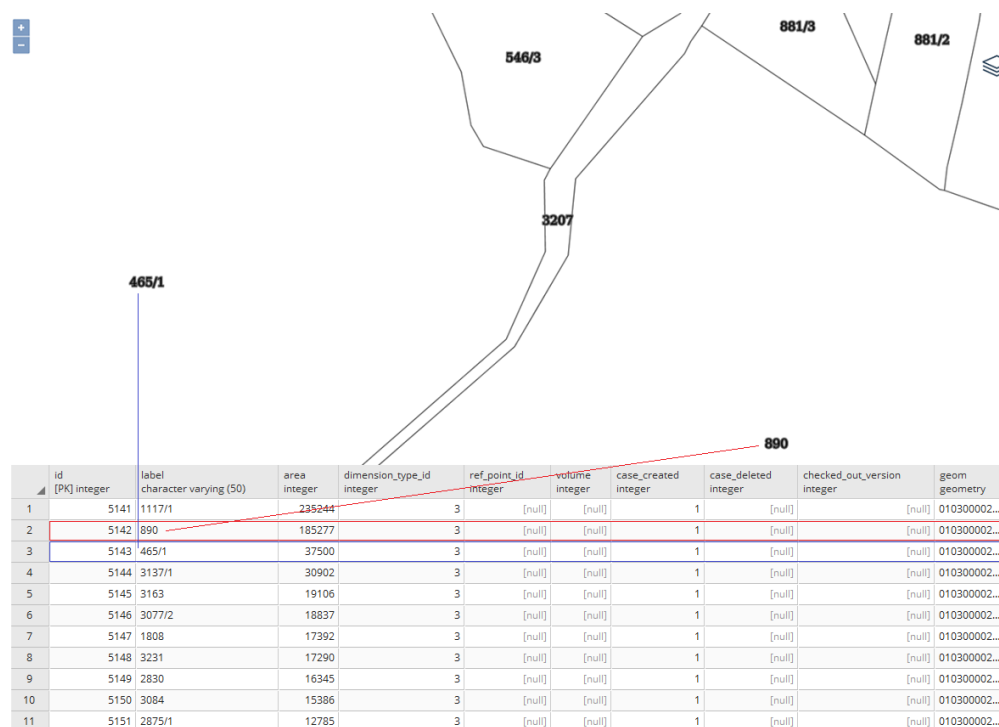
Fleksibilne transakcije (Kaiser, 1990) omogućavaju definiranje transakcija koje mogu kontrolirati korisnici. Transakcija započinje kada korisnik pokrene naredbu *BeginTransaction*. Nakon toga korisnik može pokrenuti bilo koji broj operacija (čitanja i pisanja). Transakcije imaju nesiguran završetak (eng. *open-ended*), odnosno korisnik ne mora prethodno definirati kojim će  $n$ -torkama pristupiti i koje će  $n$ -torke mijenjati. Transakcija završava kada korisnik pokrene naredbu *CommitTransaction* ili *AbortTransaction*. Fleksibilne transakcije podržavaju suradnju transakcija:

- serijalizabilnost provedbe transakcija: transakcije provedene konkurentno imaju serijalizabilni rezultat,
- definicija domene suradnje: domena suradnje skup je transakcija koje rješavaju zajednički zadatak. Transakcija ne mora biti serijalizabilna s ostalim transakcijama unutar domene, jer korisnici u domeni rješavaju nekonzistentnosti dok transakcija mora biti serijalizabilna sa svim ostalim transakcijama izvan domene.

Tool Kit model pokušava riješiti problem različitih zahtjeva različitih primjena transakcijskih modela. Primjerice, u bankarskome je sustavu izolacija bitna i obavezna, dok kod kooperativnih transakcija nije prihvatljiva. Model dopušta definiciju podtransakcija koje poštuju strogu izolaciju kao i definiciju kooperativnih podtransakcija u okviru iste hijerarhije transakcija.

## 2.3. Upravljanje konkurentnim transakcijama nad prostornim podacima

Dosad objašnjeni protokoli za upravljanje konkurentnim transakcijama kod osnovnoga transakcijskog modela kao i napredni transakcijski modeli osiguravaju konzistentnost na razini  $n$ -torke što nije dovoljno da se osigura ispravnost prostornih podataka. Prostorni se podaci u današnje vrijeme intenzivno pohranjuju u objektno-relacijske baze koje omogućavaju pohranjivanje prostornih podataka različitih struktura.



Slika 2.11: Prostorni i tablični prikaz podataka

Prostorni podaci imaju implicitne prostorne odnose koji nisu eksplicitno vidljivi (Slika 2.11), odnosno dva susjedna retka ( $n$ -torke) u tablici nisu dva susjedna poligona u prostoru i obrnuto. Kod neprostornih podataka promjena jedne  $n$ -torke može utjecati na drugu ako se identifikator  $n$ -torke koristi kao strani ključ druge  $n$ -torke. U tome je slučaju uvjet ispravnosti osiguranje referencijalnog integriteta, odnosno da se  $n$ -toraka čiji se

identifikator koristi kao strani ključ ne može brisati bez da se prethodno izbrišu vezane  $n$ -torke. Kod prostornih je podataka situacija kompleksnija, jer promjena na jednome prostornom objektu može utjecati na drugu  $n$ -torku (primjerice, kod dodavanja čvora na rub poligona). Isto se tako na prostorne podatke mogu se postaviti dodatni uvjeti koji definiraju njihovu veličinu i oblik kao što je slučaj kod katastarskih čestica i uvjeta koje im definiraju prostorni planovi.

Topološke podatkovne strukture pružaju prednosti u odnosu na strukture temeljene na poligonima. Primjerice, u radu u kojemu zagovaraju implementaciju podrške za održavanje topoloških podatkovnih struktura u relacijskim SUBP-ima Van Oosterom i dr. (2002) navode sljedeće prednosti topoloških struktura:

- izbjegava se redundancija (topološka je struktura kompaktnija od poligonalne),
- lakše održavanje konzistentnosti nakon uređivanja podataka,
- veća učinkovitost kod vizualizacije, jer se čita manje podataka s diska,
- prirodni model podataka za određene primjene (primjerice, prilikom izmjere prikupljaju se podaci o bridu, zajedno s atributima granice katastarske čestice),
- veća učinkovitost za određene vrste upita (primjerice, pronađi susjedne poligone).

Autori funkcionalnosti koje pružaju strukture temeljene na poligonima implementiraju pomoću funkcija unutar baze podataka (konkretno, autori koriste Oracle bazu podataka i proceduralni jezik PL/SQL za implementaciju funkcija). Pregled podataka zajedno s geometrijskim objektima omogućavaju kroz SQL poglede. Na taj je način ostvarena podrška za najbolje od topološke strukture (nema redundancije) i geometrijske strukture (jednostavnost analize i prikaza geometrijskih objekata).

Iako je topološka struktura kompaktnija i pruža bolje performanse kod vizualizacije, situacija je drugačija kod ažuriranja podataka temeljenih na topološkoj strukturi. Hoel i dr. (2003) uz problem učinkovitosti tipičnih upita (poput računanja površine poligona) navode i ostale nedostatke topološke strukture vezane uz postupak ažuriranja i osiguravanja ispravnosti:

- održavanje semantičkog integriteta u modelu,
- performanse i kompleksnost kod ažuriranja (moraju se ažurirati sve veze unutar topološke strukture).

Topološka struktura uvelike ovisi o vezama između pojedinih elemenata topološke strukture (čvor-brid, brid-poligon, poligon-objekt). Referencijalni integritet ne osigurava ispravnost topološke strukture. Uz referencijalni integritet mora postojati dodatni sustav provjere semantičkog integriteta u modelu. Implementacija takvoga sustava može biti u obliku okidača ili pohranjenih procedura. Međutim, takvim se dodatnim provjerama uvodi dodatna kompleksnost te se smanjuju performanse. Zbog rasporostranjenosti topoloških elemenata na više tablica te veza između njih kod jednostavnih geometrijskih operacija (primjerice, pomicanje poligona u određenome smjeru) potrebno je napraviti izmjene u više tablica što čini proces ažuriranja topološke strukture kompleksnim.

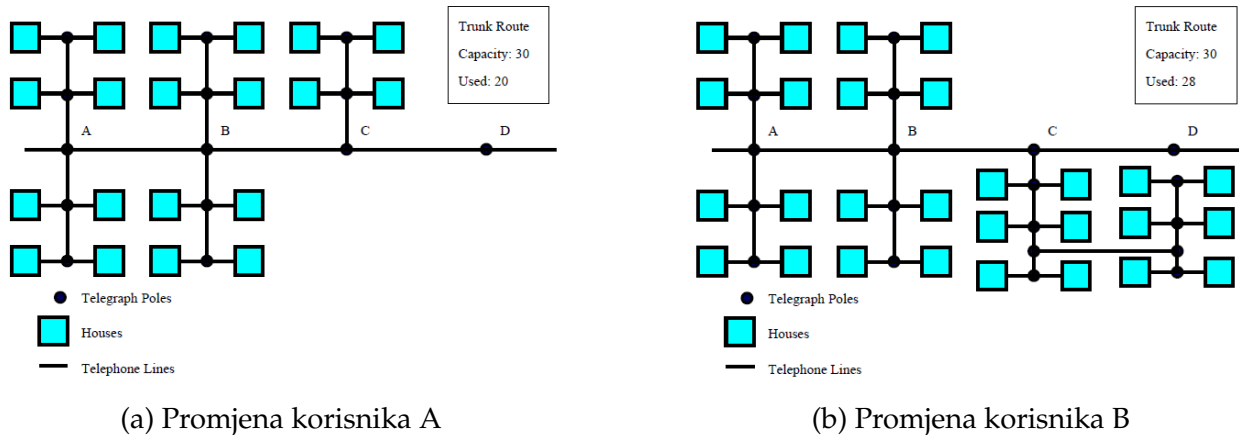
U današnje vrijeme većina sustava za upravljanje prostornim bazama podataka pruža podršku za održavanje topoloških podatkovnih struktura (Oracle Spatial, PostgreSQL+PostGIS) no to se, prema saznanjima autora, u praksi rijetko koristi. Razlog je tome upravo visoka razina kompleksnosti implementiranih mehanizama, odnosno mehanizama (korisnička sučelja, razmjenski formati, vanjski GIS alati) koji bi, u suradnji s tom implementacijom trebali djelovati. Čak i kada je implementacija topološkoga modela zahtijevana implementatori se češće odlučuju za vlastita, prilagođena rješenja. Hoel i dr. (2003) predlažu jednostavniji način pohranjivanja i validacije prostornih podataka temeljenih na poligonalnoj strukturi te uvode semantičke uvjete ispravnosti koji kontroliraju topološku ispravnost prostornih podataka (primjerice, katastarske čestice se ne smiju preklapati). Ovo je posebno naglašeno ako se razmišlja o odvajanju postupka pripreme promjena od provedbe promjena (transakcija). Razmjenski će model podataka u slučaju primjene topološke strukture biti kompleksan ili se pribjegava pojednostavljivanju (samo razmjena linija i točaka), jer je prilikom uvoza podataka potrebno kreirati veze između topoloških elemenata unutar baze podataka. Stoga se za uspostavu potpunoga topološkoga modela implementiraju naknadni mehanizmi unutar sustava koji će formirati trajne veze između topoloških elemenata iz onih privremenih koje su pohranjene u pripremljenoj promjeni u razmjenskome formatu. To opet postaje problem ako promjena treba više puta ići između osobe koja priprema promjenu i sustava u kojemu se svakoga puta treba raditi konverzija između konačnih topoloških elemenata i privremenih pohranjenih u razmjenskome formatu.

Situacija je s poligonalnim strukturama drugačija. Većina sustava za upravljanje prostornim bazama podataka (Oracle, PostgreSQL+PostGIS) podržava pohranjivanje raznih geometrijskih struktura pa tako i onih temeljenih na poligonima. Poligoni se pohranjuju kao objekti, a SUBP pruža veliki broj ugrađenih funkcija koje omogućavaju rad s poligonima na jednostavan način kao i s bilo kojim drugim jednostavnim tipom podatka. Razmjenski se formati poput GML-a temelje na paradigmi objektno-orijentiranoga modeliranja gdje su prostorni podaci pohranjeni kao skup objekata sa svojim atributima (Galić, 2006). U objektno-relacijskoj bazi podataka položaj je atribut određenog entiteta i lakše je tu strukturu zapisati u objektnu strukturu razmjenskoga formata od topološke strukture. Kod poligonalne strukture korisnik preuzima čitave objekte (zajedno s njihovim geometrijama) i skup novih verzija objekata učitava u sustav. Prilikom učitavanja ne rade se konverzije niti ažuriranje referenci prema strukturama nižih razina, jer ih nema. Zbog toga je proces ažuriranja podataka ili učitavanja gotove promjene u sustav jednostavniji kod poligonalne strukture te se u ovom radu koristi upravo prostorna sastavnica temeljena na poligonima.

Kuo i dr. (1998) prezentiraju napredni transakcijski model Constraint based Long Transaction (COLT) koji podržava duge transakcije nad prostornim podacima. Model omogućava provedbu transakcija koje ne moraju biti serijalizabilne, a koje osiguravaju ispravnost prostornih podataka definiranjem uvjeta ispravnosti. Uvjeti ispravnosti definiraju se na razini cijele baze podataka. Ako cijela baza podataka ne udovoljava određenom uvjetu,



nije ga moguće koristiti. U radu se nudi primjer rada na prostornim podacima telekomunikacijske mreže i istovremeno dva korisnika od kojih svaki radi promjene na različitim dijelovima mreže (Slika 2.12).

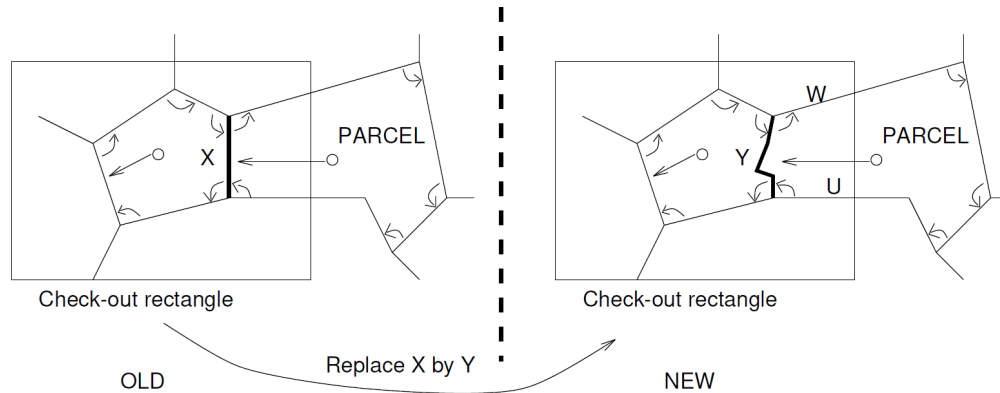


Slika 2.12: Konkurentne promjene dvaju korisnika (Kuo i dr., 1998)

Korisnik prije rada preuzima prostorne podatke lokalno (eng. *check-out*) i nakon što napravi izmjene, pokreće provedbu transakcije. Tijekom provedbe transakcije, promjene se na podacima integriraju u globalni skup podataka (eng. *check-in*). Ako postoje zavisne transakcije kao što je slučaj na slici 2.12, tada se transakcija ne odbacuje već se dopušta revizija transakcije u okviru koje se promjene iz konkurentnih transakcija integriraju. Provedba je transakcije moguća tek kada sve konkurentne transakcije postanu ispravne.

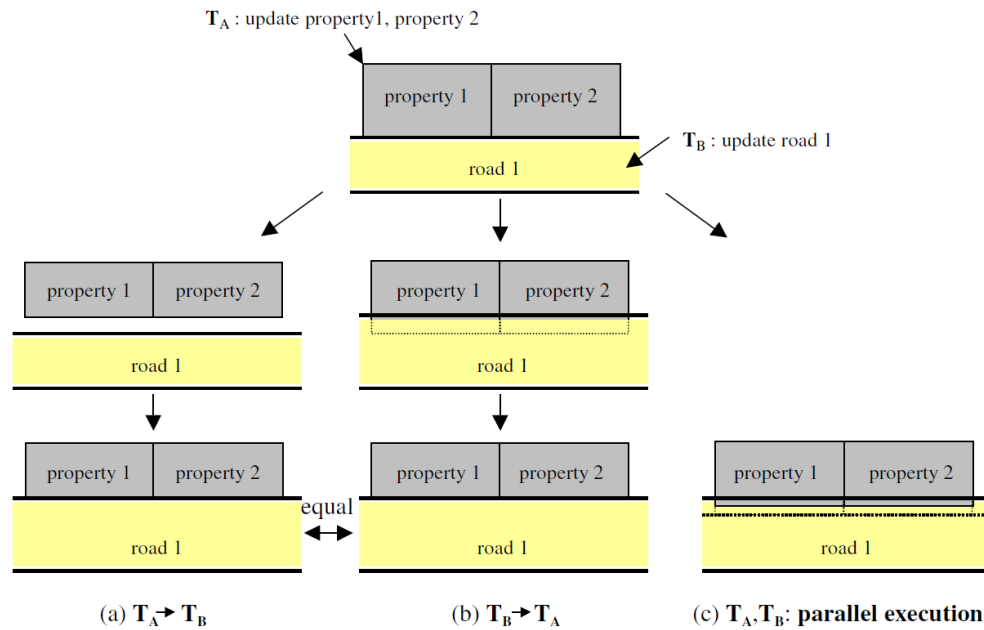
Van Oosterom (1997) se bavi održavanjem katastarskih podataka (katastarske čestice) čija se prostorna sastavnica temelji na strukturi zvanoj CHAIN-method (Lemmen i van Oosterom, 1995), a struktura se bridova temelji na strukturi krilatoga brida. U radu su razmotrena glavna pitanja vezana uz transakcije nad prostornim podacima: prostorno definiranje radnoga područja i zaključavanje objekata u njemu radi osiguravanja konzistentnosti podataka te modeliranja temporalne sastavnice. Radno područje definira korisnik pravokutnikom i svi se objekti, koji se u potpunosti nalaze unutar radnoga područja zaključavaju. Radna se područja ne smiju preklapati. Bridovi koji djelomično ulaze u radno područje djelomično su zaključani tako da se koordinate krajnjih čvorova brida ne smiju mijenjati (Slika 2.13). Podrška je za duge transakcije omogućena za bridove i katastarske čestice dodjeljivanjem trenutaka stvaranja (*tmin*) i brisanja (*tmax*) (Hunter i Williamson, 1990) što omogućava da transakcije čitanja podataka nisu blokirane od strane transakcija pisanja budući da se izmjene rade na privatnim verzijama. Korisnik prije uređivanja podataka preuzima objekte (eng. *check-out*), radi izmjene te provedbom transakcije (eng. *check-in*) novi podaci postaju važeći, a preuzeti se arhiviraju. Budući da se arhiviraju i bridovi i katastarske čestice, kod slučaja je sa slike 2.13, prije provedbe transakcije potrebno provjeriti jesu li nastale nove verzije katastarskih čestica u konkurentnim transakcijama za vrijeme pripremanja transakcije. Ako postoje konkurentne promjene na zajedničkoj katas-

tarskoj čestici, podaci se preuzimaju iz konkurentne transakcije te spajaju s promjenama u predmetnoj transakciji i tek je tada transakcija spremna za provedbu. Ovom se provjerom osigurava serijalizabilnost transakcija.



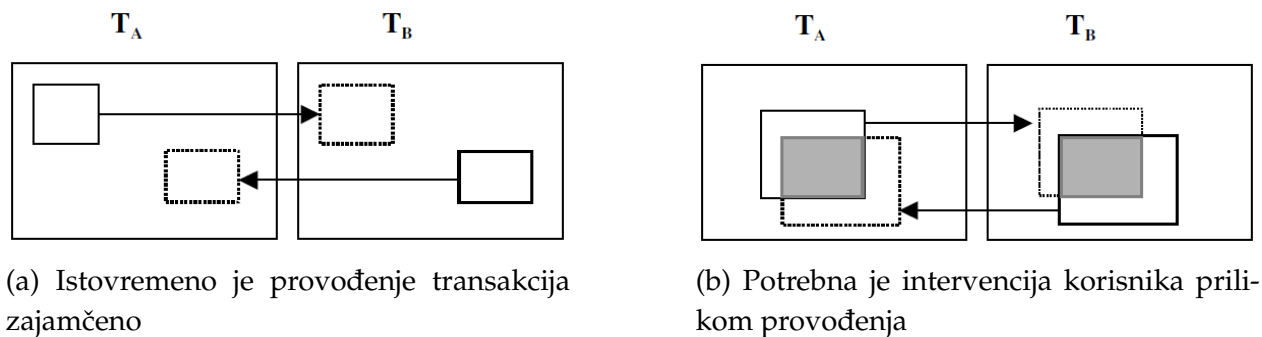
Slika 2.13: Problematično ažuriranje (van Oosterom, 1997)

Choi i dr. (1999) se bave osiguravanjem konzistentnosti prostornih podataka kod integracije promjena iz različitih transakcija u distribuiranim bazama podataka. U radu definiraju radno područje koje je definirano pravokutnikom. Radno se područje dijeli između distribuiranih baza podataka kako bi transakcije mogle ispitivati postoje li konkurentne transakcije. Autori predstavljaju primjer problema integriranja promjena iz dvije konkurentne transakcije (Slika 2.14). Ako transakcija  $T_a$  promijeni (smanji) objekte 1 i 2, a transakcija  $T_b$  proširi cestu, te se dvije transakcije mogu provesti istovremeno. Problem koji se tu nameće jest da transakcija  $T_a$  mora znati za  $T_b$ , inače će provedbom  $T_a$  nastati pogreška (rupa) u podacima. Nadalje, ako jedna od transakcija obustavi promjene, opet je moguće da će se pojaviti pogreška u podacima. Ako se radna područja preklapaju, ispituje se prostorni odnos objekata koji sudjeluju u transakciji. Ako objekti imaju prostorni odnos DISJOINT, tada se transakcije mogu provesti istovremeno, a u suprotnome je potrebna intervencija korisnika koji procjenjuje mogu li se transakcije provesti istovremeno.



Slika 2.14: Primjer konfliktne situacije (Choi i dr., 1999)

Autori zaključuju da se tradicionalni optimistični pristupi teško mogu primijeniti na kontrolu repliciranja prostornih objekata kod distribuiranih baza podataka te definiraju koncepte zonskoga zaključavanja (eng. *region locking*) i zaključavanja pisanja temeljenoga na prostornom odnosu (eng. *Spatial Relationship-Bound Write locking*). Primjenom je tih koncepata moguće provesti više transakcija istovremeno ako radna područja transakcija ne dijele zajednički prostorni odnos (odnosno imaju odnos DISJOINT). Autori tvrde da se transakcije mogu provesti i u slučaju kada radna područja imaju zajednički odnos primjenom proširenoga protokola provedbe u dvije faze, nazvanoga protokol provedbe u dvije faze temeljen na prostornim odnosima (eng. *Spatial Relationship-based 2PC protocol*). U tome je slučaju potrebna suradnja između konkurentnih transakcija kako bi se promjene iz tih transakcija ispravno provele, odnosno potrebna je intervencija korisnika (Slika 2.15).



Slika 2.15: Upravljanje konkurentnim transakcijama (Choi i dr., 1999)

Matijević (2006) se bavi modeliranjem promjena nad prostornim podacima u katas-

tru. Prostorna se sastavnica temelji na eksplicitno pohranjenoj topološkoj polu-brid strukturi. U radu su definirani uvjeti ispravnosti geometrijskih i topoloških promjena nad katastarskim podacima te operatori koji su potrebni za upravljanje topološkom polu-brid strukturom. Ispravnost ravninskih grafova, koji su rezultat prostornih operacija, testira se pomoću Eulerove karakteristike topoloških petlji koje ulaze i izlaze iz promjene. Rad se bavi formalizacijom promjena, a ne transakcijama. Međutim, takva se definicija promjene može primijeniti u okviru navedenih transakcijskih modela koji omogućavaju definiranje proizvoljnih uvjeta ispravnosti. Utjecaj promjene na poligonu na susjedne poligone uzrokovan topološkim i geometrijskim promjenama samo je spomenut. Matijević i dr. (2008) dodatno su razradili i nadogradili koncept uvođenjem indeksiranja prostornih podataka čime su značajno poboljšane performanse ispitivanja ispravnosti ravninske particije.



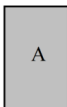

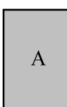
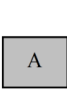
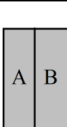
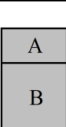
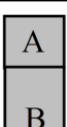
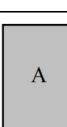
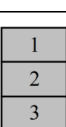

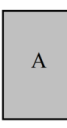
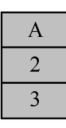
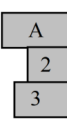
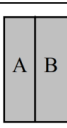


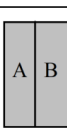
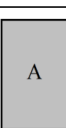
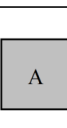
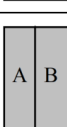
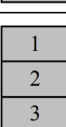

Vranić i dr. (2015) se također bave transakcijama nad katastarskim česticama. Transakciju definiraju kao prijelaz katastarskih podataka iz jednoga konzistentnog stanja u drugo prema konceptu prijelaznoga neto učinka transakcije objašnjenog u Widom i Finkelstein (1990). Prema tome se konceptu pohranjuje samo konačni učinak transakcije koji se kontrolira. Primjerice, ažuriranje nakon kojega slijedi brisanje, rezultira brisanjem objekta. Zbog toga je faza pripremanja transakcije odvojena od faze provjere transakcije. Autori su na temelju postojećih istraživanja napravili sintezu vrsta transakcija nad katastarskim česticama te definirali uvjete ispravnosti kako bi iste mogle biti predane vanjskim dionicima na izvođenje (Tablica 2.1).

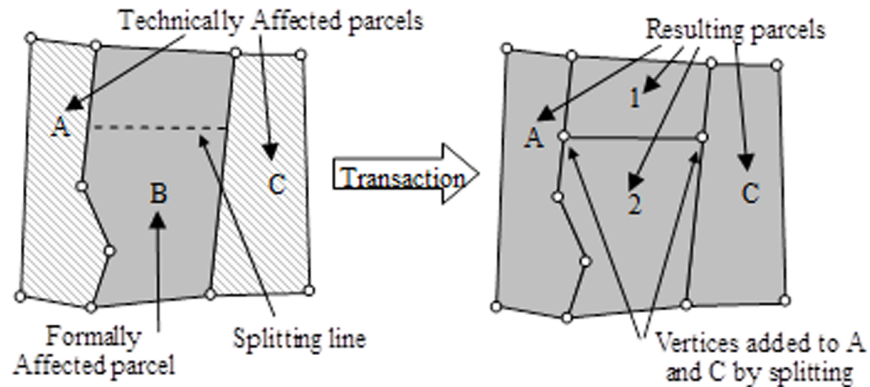
Vranić i dr. (2015) definiraju pojam formalno zahvaćenoga područja koje predstavlja radno područje transakcije, a definirano je geometrijskom unijom katastarskih čestica koje sudjeluju u transakciji. Prostorna je sastavnica katastarskih čestica predstavljena geometrijom u skladu s normom ISO 19125 (ISO, 2004a,b). Kod takvoga je modela svaki brid pohranjen dvaput osim ako se ne radi o bridu na rubu skupa prostornih podataka. Zbog takve je redundancije moguće je uvođenje pogrešaka poput preklapanja ili rupa između poligona. Stoga postoji mogućnost da se zahvaćeno područje tijekom transakcije izmijeni ako se napravi izmjena na granici formalno zahvaćenoga područja. Da bi se podržala mogućnost ispravljanja takvih pogrešaka, većina prethodnih navedenih vrsta transakcija ima dvije varijante:

- Nepromjenjivo zahvaćeno područje,
- Promjenjivo zahvaćeno područje.

Kod promjenjivo zahvaćenoga područja moguće je dodavati, brisati ili mijenjati položaj čvorova na rubu formalno zahvaćenoga područja (Slika 2.16). Da bi bilo moguće dodati čvorove i na susjednim poligonima, potrebno ih je tehnički dodati u transakciju. Tehničkim se dodavanjem čestica u transakciju one zaključavaju što znači da druga transakcija ne može raditi ikakve promjene na tim česticama.

Tablica 2.1: Vrste transakcija (Vranić i dr., 2015)

Br. zahvaćenih čestica	Br. rezultirajućih čestica	Zahvaćeno područje	Zahv. područje nepromijenjeno	Zahv. područje promijenjeno	Transakcija
0	1..m		n/a		Registracija geometrije
1..n	0		n/a		Brisanje geometrije
1	1		n/a		Ispravljanje geometrije
n	n				Višestruko ispravljanje geometrije
1	m				Dioba
1	m				Oduzimanje
n	1				Spajanje
n	1				Pripajanje
n	m				Preraspodjela



Slika 2.16: Dioba koja zahtijeva promjene na susjednim česticama (Vranić i dr., 2015)

Ispitivanjem su topološkog odnosa zahvaćenih i rezultirajućih katastarskih čestica definirani uvjeti ispravnosti koji osiguravaju sigurno izvođenje transakcije nad katastarskim česticama i navedeni su u nastavku:

- vrsta transakcije,
- preklapanje verzija katastarskih čestica,
- ravninska particija,
- opuštena ravninska particija,
- preklapanje katastarskih čestica i
- preklapanje zahvaćenog i rezultirajućega područja.

Definirane vrste transakcija i uvjeti ispravnosti omogućavaju da se deklarirana vrsta transakcije provede te da transakcija prevede katastarske podatke iz jednoga konzistentnog stanja u drugo prema ACID svojstvima. Međutim, autori u obzir ne uzimaju konkurentne procese. Da bi se zadržala ispravnost prostorne sastavnice i izolacija transakcija, potrebno je zaključati sve formalno i tehnički zahvaćene katastarske čestice. Na ovaj način postoji mogućnost da se zaključa veći broj katastarskih čestica, pogotovo ako je izdužena čestica (primjerice, cesta, vodotok i slično) predmet transakcije ili je samo tehnički zahvaćena transakcijom.

## 2.4. Rekapitulacija

Relacijske, odnosno objektno-relacijske baze podataka upravljaju podacima kroz transakcije prema osnovnome transakcijskom modelu u skladu s ACID svojstvima. Osnovni transakcijski model podrazumijeva kratko trajanje transakcije, a operacije su unutar transakcije jednostavne operacije čitanja i zapisivanja podataka što za naprednije primjene nije dovoljno. Zbog toga su razvijeni brojni napredni transakcijski modeli koji nadograđuju određeni aspekt osnovnoga transakcijskog modela. Većina naprednih transakcijskih modela podržava dugo trajanje transakcije dok suradnju na transakcijama podržavaju samo

neki. Međutim, iako transakcijski sustavi osiguravaju ispravnost transakcija, njihova povezanost sa SUBP-om dodatno ograničava fleksibilnost, jer funkcionalnosti SUBP-a ne podržavaju heterogenu informatičku infrastrukturu (primjerice web servise).

Nijedan od naprednih transakcijskih modela ne može biti direktno primijenjen na kontroliranje ispravnosti prostorne sastavnice katastarskih čestica bez znatnih izmjena. Radovi koji se bave kontroliranjem ispravnosti prostornih, odnosno katastarskih podataka uglavnom se temelje na ideji prostorne definicije radnoga područja transakcije. Način definicije radnoga područja kao i kriteriji za kontroliranje ispravnosti prostorne sastavnice ovise o pristupu kao i o strukturi prostorne sastavnice. Van Oosterom (1997) se bavi podacima o katastarskim česticama čija se prostorna sastavnica temelji na topološkoj strukturi, dok ostali radovi (Choi i dr., 1999; Vranić i dr., 2015) koriste strukturu temeljenu na poligonima prema normi ISO 19125 (ISO, 2004a). Opisani pristupi uglavnom ograničavaju pristup većemu skupu podataka zbog osiguravanja izolacije transakcija u slučaju konkurentnoga pristupa, dok se promjena uglavnom događa na manjem skupu. U fokusu je tih istraživanja osiguravanje ispravnosti prostornih podataka dok je upravljanje konkurentnim transakcijama izvan fokusa. S obzirom na duže trajanje transakcija u sustavima za upravljanje zemljištem kao i na činjenicu da se prostornim definiranjem obuhvata transakcije mogu zaključati veliki skupovi podataka, primjena kvalitetnoga protokola za upravljanje konkurentnim transakcijama znatno može utjecati na konkurentnost, odnosno učinkovitost transakcija.

### 3. Sustavi za upravljanje tijekom rada

Tijek rada (eng. *Workflow (WF)*) računalno je poboljšanje ili automatizacija poslovnoga procesa, djelomično ili u cijelosti. Sustav koji u potpunosti definira, upravlja i provodi tijekove rada kroz provedbu zadataka čiji je redoslijed provedbe određen strukturom tijeka rada naziva se sustav za upravljanje tijekom rada (eng. WFMS). (WFMC, 1995). Sustavi za upravljanje tijekom rada nastaju zbog potrebe da se eliminiraju nedostaci transakcijskih modela i poveća fleksibilnost upravljanja poslovnim procesima. Koncept je WFMS-a rezultat doprinosa iz nekoliko interdisciplinarnih područja poput programskog inženjerstva, upravljanja procesima, baza podataka (Rusinkiewicz i Sheth, 1995) i distribuiranih sustava (Georgakopoulos i Hornick, 1994; Georgakopoulos i dr., 1994), a razvoj je počeo sredinom 1990-ih godina.

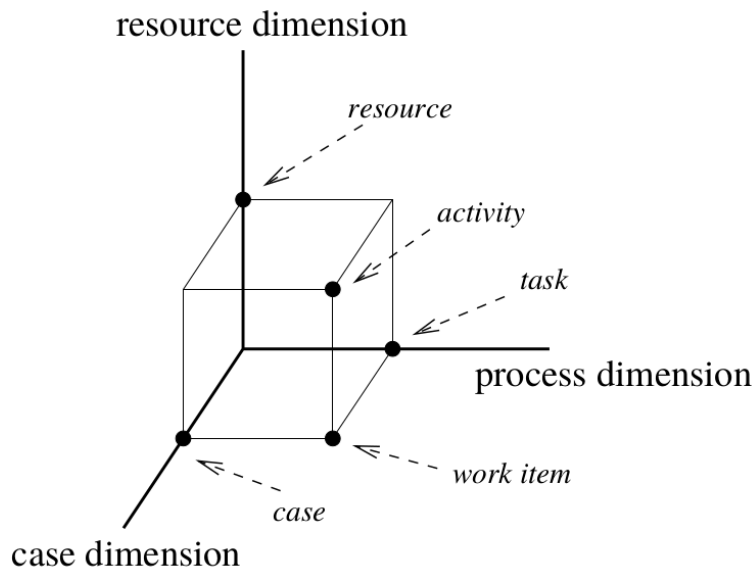
WFMS-i omogućavaju suradnju različitih dionika na procesu te pružaju podršku za zadatke koje mogu pokretati dionici ili se mogu pokretati automatski. Zadatak može predstavljati računalni programi, web servis ili skup SQL naredbi. Kao u počecima korištenja baza podataka, tako su se i WFMS-i u početku najviše koristili u uredskim okruženjima poput bankarstva, osiguranja, administracije (WFMC, 1995), no koncept se WFMS-a s vremenom počeo primjenjivati u širokom rasponu djelatnosti poput industrije i proizvodnje (Leymann i Roller, 2000) ili upravljanja prostornim podacima (Alonso i Hagen, 1997; Weske i dr., 1998; Yue i dr., 2015; Du i Cheng, 2017). U posljednje je vrijeme fokus istraživanja vezanih uz WFMS promijenjen na poboljšavanje WFMS-a (Van Der Aalst, 2000), pronalaženjem zajedničkih komponenti koje se koriste (Van der Aalst i ter Hofstede, 2002; Van der Aalst i dr., 2003) s ciljem njihove standardizacije kako bi se olakšala njihova implementacija u notacijama za modeliranje tijeka rada (Van der Aalst, 1998; Van der Aalst i ter Hofstede, 2005). Postoji i trend stvaranja WFMS-a koji nadilaze pojedine organizacije te omogućavaju interakciju s dionicima izvan organizacije (Van der Aalst, 2000; Meng i dr., 2002) te korištenja dostupnih web servisa i senzora (Yue i dr., 2015). Razvojem tehnologije Blockchain pojavili su se primjeri njezine implementacije u WFMS u financijskome sektoru (Fridgen i dr., 2018), ali koncept je primjenjiv na druga područja pa tako i na sustave za upravljanje zemljištem (Anand i dr., 2016; Enemark i McLaren, 2017; Anand, 2017; Lemmen i dr., 2017).



### 3.1. Terminologija

Udruženje za upravljanje tijekom rada (eng. *Workflow Management Coalition (WFMC)*) globalna je organizacija pojedinaca i organizacija koje koriste, razvijaju i istražuju upravljanje tijekom rada te upravljanje poslovnim procesima (eng. *Business Process Management (BPM)*). WFMC razvija standarde, educira tržište o vezanim temama te je jedina organizacija usmjerena na razvoj standarda vezanih isključivo za poslovne procese. WFMC je 1999. godine objavio rječnik pojmova (WFMC, 1999) vezanih uz modeliranje tijeka rada, a nastavku su navedene definicije koje se koriste u ovom radu.

Glavni je predmet sustava za upravljanje tijekom rada **poslovni proces** ili samo **proces**. Poslovni proces skup je jedne ili više povezanih procedura, odnosno aktivnosti koje ostvaruju određeni cilj. Trajanje procesa počinje izvođenjem prve aktivnosti i završava završetkom posljednje aktivnosti. U periodu između početka prve i završetka posljednje aktivnosti proces je aktivan. Primjer procesa može biti kupnja avionske karte, reklamacija proizvoda ili dioba katastarske čestice. Određeni proces još se naziva **slučaj** (eng. *case*). Upravljanje tijekom rada se temelji na slučajevima (eng. *case-based*), odnosno svaki se posao obavlja kroz određeni slučaj. Cilj je upravljanja tijekom rada obrađivati slučajeve uspješno i učinkovito. Definicija procesa određuje redoslijed i način provedbe procesa. **Zadatak** predstavlja skup operacija koje predstavljaju logičku cjelinu u okviru procesa. Zadatak se može pokrenuti na više načina, primjerice automatski, intervencijom dionika, nakon isteka određenoga vremena ili u određeno vrijeme. WFMS upravlja slučajevima koji se stvaraju u skladu s definicijom procesa. Redoslijed zadataka je određen definicijom procesa. WFMS u slučaju automatskoga zadatka poziva aplikaciju (eng. *invoked application*) ili dodjeljuje zadatak dioniku sustava ako je potrebna njegova intervencija. Zadatak koji se provodi u okviru određenoga slučaja naziva se **radna stavka** (eng. *work item*), a radna stavka koju izvršava određeni dionik, odnosno općenitije resurs naziva se **aktivnost** (eng. *activity*). Između zadataka mogu se definirati uvjeti ovisnosti koji mogu predstavljati preduvjete ili postuvjete određenoga zadatka. Preduvjet definira može li se zadatak uopće pokrenuti, a postuvjetom se provjerava ispravnost rezultata koji je određeni zadatak vratio. **Resurs** je općeniti pojam koji označava stroj (printer, web servis) ili osobu koja sudjeluje u radu na slučajevima. Resursi se prema karakteristikama grupiraju u klase resursa (eng. *resource class*). U jednoj klasi može postojati više resursa, a određeni se resurs može nalaziti u više klasa. Ukoliko se klase temelje na svojstvima, odnosno funkcionalnim preduvjetima, nazivaju se uloge (eng. *role*). Ukoliko se podjela u klase temelji na organizacijskoj strukturi, klasa se resursa naziva organizacijska jedinica (eng. *organizational unit*). Svaki zadatak može biti dodijeljen različitom resursu što omogućava raspodjelu posla i rasterećenje resursa što posredno može dovesti do povećanja učinkovitosti.

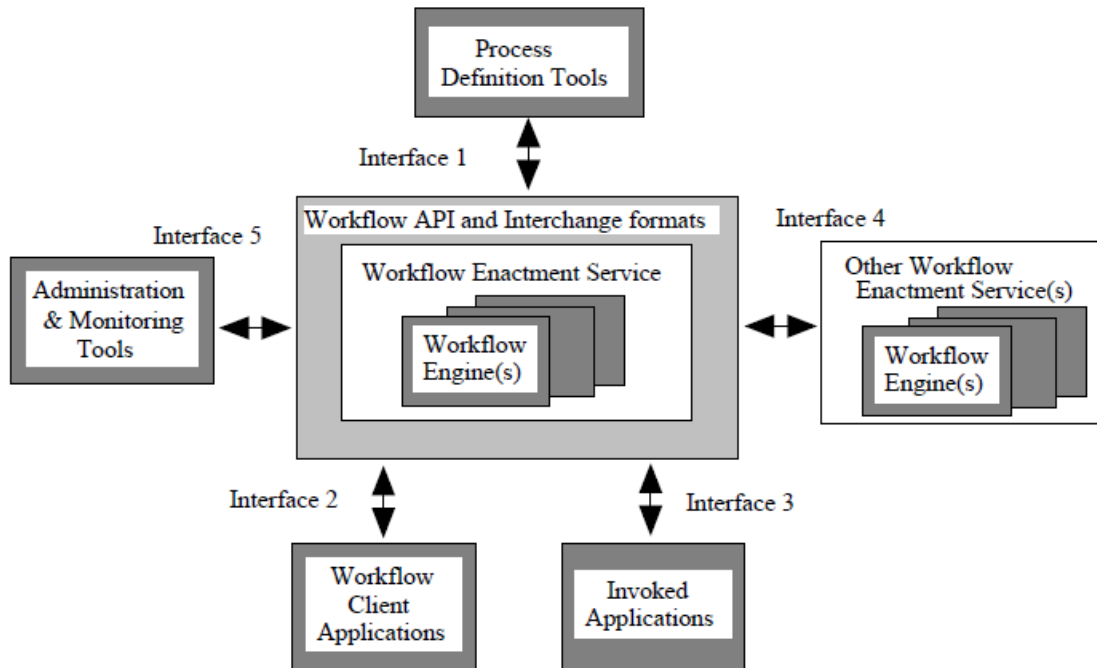


Slika 3.1: Trodimenzionalni pogled na tijek rada (Van der Aalst, 1998)

Slika 3.1 prikazuje tri dimenzije tijeka rada: dimenzija slučaja, dimenzija procesa i dimenzija resursa. Dimenzija slučaja označava da se svaki slučaj obrađuje posebno. S gledišta tijeka rada, slučajevi direktno ne utječu jedan na drugoga, ali postoji ovisnost zbog dijeljenja resursa i podataka. Dimenzija procesa definira zadatke i navigaciju između tih zadataka. Dimenzija resursa određuje grupiranje resursa u uloge i organizacijske jedinice. U takvom koordinatnom sustavu može se predložiti tijek rada točkama. Svaka točka može predstavljati radnu stavku (slučaj+zadatak) ili aktivnost (slučaj+zadatak+resurs). Upravljanje tijekom rada objedinjuje tri dimenzije (slučaj, zadatak i organizacija) prikazane na slici 3.1 (Van der Aalst, 1998). Treća je dimenzija, odnosno upravljanje resursima izvan opsega ovoga rada te će biti obrađena koliko je to nužno za obrazlaganje prvih dviju dimenzija.

### 3.2. Referentni model sustava za upravljanje tijekom rada

Referentni model WFMS-a definira zajedničke karakteristike, komponente te opću strukturu WFMS-a (WFMC, 1995). Referentni model definira opću strukturu WFMS-a pomoću komponenti koje nisu međusobno ovisne već komuniciraju kroz zajednička sučelja (eng. *Workflow application programming interface (WAPI)*) što omogućava fleksibilnost, jer komponente mogu komunicirati međusobno na više načina i razina (Slika 3.2). Takva je slabo povezana arhitektura WFMS-a posebno pogodna za implementaciju pomoću tehnologija poput web servisa.



Slika 3.2: Referentni model sustava za upravljanje tijekom rada (WFMC, 1995)

Dva su osnovna modula svakoga WFMS-a: modul za definiciju (eng. *Process Definition Tools*) i modul za interpretaciju procesa (eng. *Workflow Enactment Service*) (Slika 3.2). WFMS može sadržavati i modul za praćenje procesa koji omogućuje analizu i poboljšanje poslovnih procesa u smislu paradigme reinženjerstva poslovnih procesa (eng. *Business Process Re-engineering (BPR)*). Dva WFMS-a mogu međusobno komunicirati kroz sučelje 4 i na taj je način podržana uspostava WFMS-a distribuiranog između više organizacija (Van der Aalst, 2000; Meng i dr., 2002; Fekete i dr., 2003). Aktivnosti koje u okviru tijeka rada obavljaju razne operacije mogu biti web servisi, računalni programi ili procedure na bazi podataka. Vanjski ili unutarnji servisi/programi mogu se pozivati preko sučelja 3. Način je i trenutak njihova pozivanja određen logikom tijeka rada, a logika pojedine aktivnosti nije poznata sustavu za upravljanje tijekom rada.

Modul za definiciju procesa služi za opisivanje procesa u obliku razumljiv računalima. Taj se oblik može temeljiti na formalnome jeziku za opisivanje procesa, objektnome modelu, relacijskome modelu ili jednostavnijim sustavima poput raznih skripti. Modul za definiciju procesa može biti dio sustava za upravljanje tijekom rada ili dio opširnijeg alata za analizu poslovnih procesa. Postoje različiti grafički jezici za modeliranje i prikaz poslovnih procesa, a WFMC je definirao format XML Data Processing Language (XDPL) temeljen na XML-u (engl. eXtensible Markup Language) koji omogućava razmjenu definicija procesa (WFMC, 2012).

Sama definicija procesa sadrži sve potrebne informacije o procesu kako bi se omogućilo izvođenje takvoga procesa u modulu za interpretaciju procesa. Definicija procesa obuhvaća informacije o uvjetima koji moraju biti ispunjeni da bi proces počeo ili završio,

zadatke od kojih je proces sastavljen te pravila navigacije između njih.

Definicija procesa može uključivati i podatke o organizacijskoj shemi i pravima korisnika unutar organizacije. Na taj su način određene aktivnosti i objekti unutar procesa dodijeljeni određenoj ulozi korisnika umjesto određenome korisniku. Modul za interpretaciju procesa tada ima odgovornost dodijeliti određenu aktivnost korisniku koji ima pravo, odnosno kojemu je dodijeljena određena uloga.

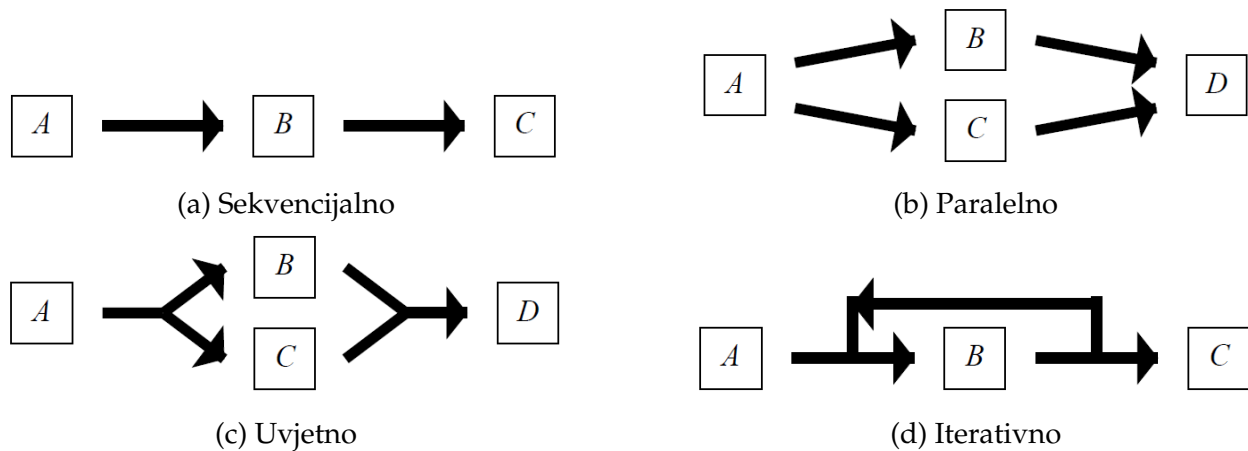
Modul za interpretaciju procesa interpretira opis procesa, kontrolira instanciranje procesa i aktivnosti te zadavanje zadataka korisniku i pozivanje programskih alata ukoliko je to potrebno. Modul za interpretaciju procesa servis je koji se sastoji od jednog ili više upravitelja tijeka rada (eng. *workflow engine*) koji kreiraju, upravljaju i provode instance tijeka rada. BPEL (eng. Business Process Execution Language) je jezik (temelji se na XML-u) koji se može koristiti za prosljeđivanje uputa upravitelju tijeka rada na koji način i kojim se redoslijedom pozivaju zadaci (web servisi). Komunikacija se između različitih sustava može ostvariti putem različitih protokola poput SOAP-a (eng. *Simple Object Access Protocol*). Dostupnim tehnologijama, koje se temelje na web servisima, moguće je povećati učinkovitost upravljanja procesima. Primjerice, sustav za upravljanje zemljištem može pomoću web servisa na provjeru poslati podatke u sustav prostornoga planiranja, koji pak može povratnu informaciju vratiti isto putem web servisa.

### 3.3. Komponente tijeka rada

Svaki tijek rada ima svoj početak i završetak, odnosno početni i završni zadatak. Između ta dva zadatka može biti proizvoljan broj zadataka koji se mogu provoditi kako bi se obavio poslovni proces. Način njihove povezanosti ovisi o uvjetima između zadataka i logici tijeka rada (Alonso i dr., 1997). WFMC (1999) definira četiri općenite vrste usmjeravanja tijeka rada (Slika 3.3):

- sekvencijalno (eng. *sequential*). Provedba jednoga zadatka slijedi provedbu drugoga. Na slici 3.3a zadatak *B* se provodi nakon *A*, a prije *C*.
- paralelno (eng. *parallel*). Na slici 3.3b zadaci *B* i *C* se provode istovremeno ili bilo kojim redoslijedom budući da ne ovise jedan o drugome. Sastoji se od dva dijela: *AND – split* (tijek se rada razdvaja na dva paralelna tijeka) i *AND – join* (tijekovi se rada spajaju u jedan). *D* može početi nakon što *B* i *C* završe.
- uvjetno (eng. *conditional*). Na slici 3.3c provodi se zadatak *B* ili *C*. Za modeliranje se ove vrste koriste dva dijela: *OR – split* i *OR – join*. Nakon što se zadatak *A* provede, radi se izbor prema uvjetima između *B* i *C*. *D* može započeti nakon što se provede *B* ili *C*. Ukoliko se izbor obavlja prije, takvo se grananje naziva ekskluzivno ili (*XOR*), a ukoliko se izbor određene grane odgađa ili odvija kasnije tada se radi o običnome grananju (*OR*).
- iterativno (eng. *iteration*). Ponekad je određeni zadatak potrebno provesti nekoliko puta. Na slici 3.3d prikazan je zadatak *B* koji se provodi jednom ili više puta dok se

određeni uvjet ne ispuni. Ovakvi se tijekovi rada uglavnom izbjegavaju. Iteracije bi trebalo obaviti unutar pojedinoga zadatka.



Slika 3.3: Vrste usmjeravanja tijeka rada (Van der Aalst, 1998)

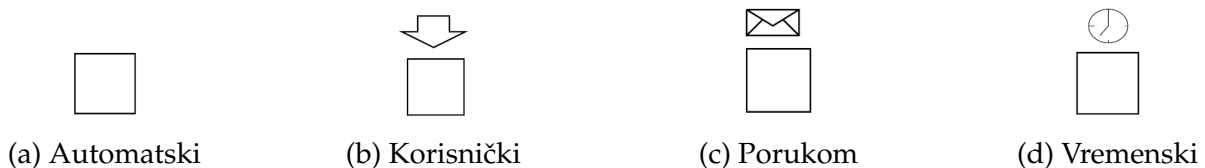
Slični su uzorci navedeni u Eder i Liebhart (1994); Alonso i dr. (1996). Van der Aalst i dr. (2003) prepoznaju 20 uzoraka koji se najčešće koriste kod modeliranja tijeka rada te ih uspoređuju s postojećim aplikacijskim rješenjima kako bi utvrdili njihovu zastupljenost. U nastavku su navedene grupe tih uzoraka:

- Osnovni uzorci kontrole tijeka rada (Slika 3.3).
- Napredni uzorci grananja i sinkronizacije. Nadograđuju osnovne uzorke kontrole tijeka rada s elementima poput višestrukoga grananja, fleksibilnih vrsta spajanja procesa (Synchronizing merge).
- Strukturni uzorci. Uzorci koji omogućavaju manje strogu strukturu tijeka rada (Arbitrary Cycles, Implicit Termination).
- Uzorci koji uključuju višestruke instance. Omogućavaju da se dijelovi procesa instanciraju višestruko. Primjerice, kod obrade štete kod osiguranja moguće je uzeti izjave od više svjedoka.
- Uzorci temeljeni na stanjima. Klasični WFMS fokusiran je na aktivnosti, a ne na stanja. To ograničava modeliranje tijeka rada kod modeliranja uzoraka temeljenih na stanjima (Deferred Choice, Interleaved Parallel Routing, Milestone).
- Uzorci otkazivanja koji uključuju otkazivanje pojedine aktivnosti ili čitavoga procesa.

Iz analize autori zaključuju da nijedno aplikacijsko rješenje ne podržava svih 20 uzoraka. Ustvari mnogi alati podržavaju relativno mali podskup naprednih uzoraka. Osnovne uzorke podržava većina aplikacijskih rješenja uzetih u obzir. White (2004) koristi 20 prepoznatih uzoraka u Van der Aalst i dr. (2003) te za svaki od njih ispituje može li se modelirati pomoću notacije za modeliranje poslovnih procesa (eng. *Business Process Management Notation (BPMN)*) i UML dijagrama aktivnosti (eng. *UML Activity Diagram (UML AD)*). Autor zaključuje kako većina uzoraka može biti prikazana korištenjem bilo kojega jezika.

Osim komponenti koje omogućavaju usmjeravanje tijeka rada, na različite je načine potrebno uvesti i koncept pokretanja/okidanja (eng. *triggering*) pojedinoga zadatka. Primjerice, ako je određen zadatak dodijeljen djelatniku, on mora biti dostupan kako bi ga proveo. Ako je djelatnik bolestan, zadatak se ne može provesti. WFMS ne može prisiliti djelatnika da provede dodijeljeni zadatak. Zato je potrebno razdvojiti pojmove omogućavanja i pokretanja zadatka. Okidač (eng. *trigger*) je vanjski uvjet koji vodi do pokretanja omogućenoga zadatka. Van der Aalst (1998) definira četiri vrste zadataka prema načinu pokretanja (Slika 3.4):

- automatski (Slika 3.4a): zadatak se pokreće čim se omogući. Koristi se kod pozivanja aplikacija koja ne zahtijeva intervenciju korisnika.
- korisnički (Slika 3.4b): zadatak se pokreće intervencijom korisnika, odnosno korisnik pokreće omogućeni zadatak.
- porukom (Slika 3.4c): vanjski događaj (poruka) provodi omogućeni zadatak (primjerice, e-mail, web servis).
- vremenski (Slika 3.4d): omogućeni se zadatak provodi u definirano vrijeme ili ako je u određenome statusu više od predviđenoga perioda (primjerice, ako korisnik ne provede zadatak za 10 sati).



Slika 3.4: Načini pokretanja zadataka (Van der Aalst, 1998)

Pomoću definiranih načina pokretanja zadataka mogu se definirati alternativni smjеровi tijeka rada koji će uzeti u obzir dostupnost potrebnih resursa i podataka kako ne bi došlo do zastoja u procesu. Primjerice, pomoću OR dijeljenja moguće je definirati dva alternativna tijeka rada: pripremanje promjene od strane katastarskoga službenika ili učitavanje pripremljene promjene od strane ovlaštenoga mjernika, primjerice u razmjenskome formatu Geography Markup Language (GML) (Galić, 2006). Odvajanjem se faze omogućavanja i pokretanja zadatka te korištenjem uzoraka za upravljanje tijekom rada mogu spriječiti situacije koje se mogu dogoditi kada je određeni resurs nedostupan (primjerice, ako je određeni djelatnik bolestan). U takvim se slučajevima može postaviti vremensko ograničenje unutar kojega zadatak mora biti prihvaćen inače se može dodijeliti nekome drugome. Time se sprječavaju mogući zastoji i podiže učinkovitost i iskoristivost resursa.

### 3.4. Modeliranje i grafički prikaz procesa

Procesi su dominantni element WFMS-a i potrebno je koristiti kvalitetan način za modeliranje i analiziranje procesa. U nastavku se analiziraju tri načina modeliranja i prikaza

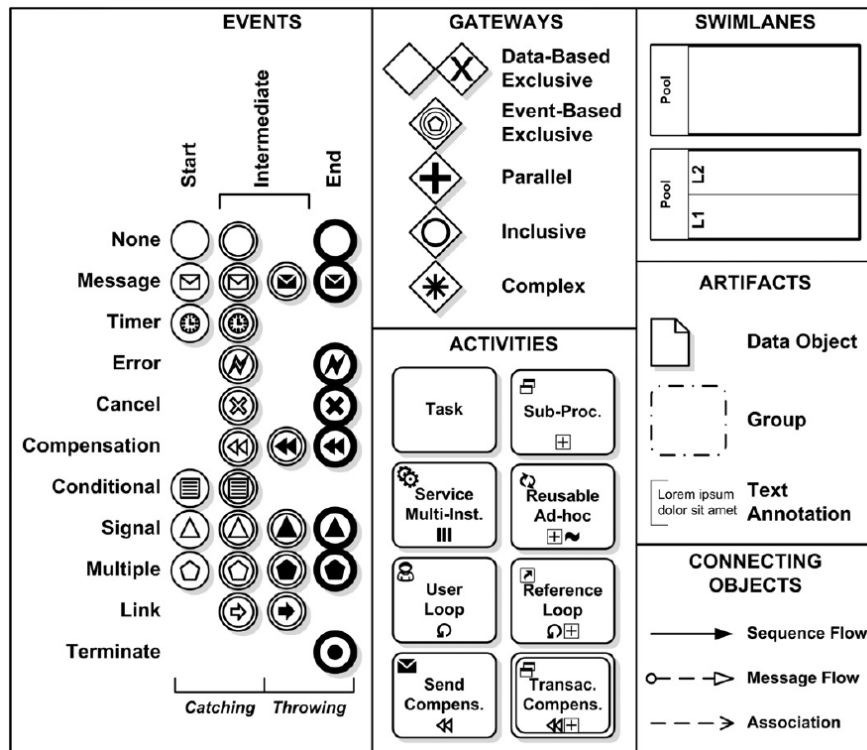
poslovnih procesa koji su standardizirani od strane međunarodne organizacije za norme ISO.

### 3.4.1. Notacija za modeliranje poslovnih procesa

Notacija za modeliranje poslovnih procesa (*BPMN*) (OMG, 2013) omogućava standardno zapisivanje poslovnih procesa na grafički i intuitivan način različitim dionicima. BPMN je prihvaćen i kao međunarodna norma ISO 19510 (ISO, 2013).

Osnovni su elementi BPMN dijagrama (Slika 3.5):

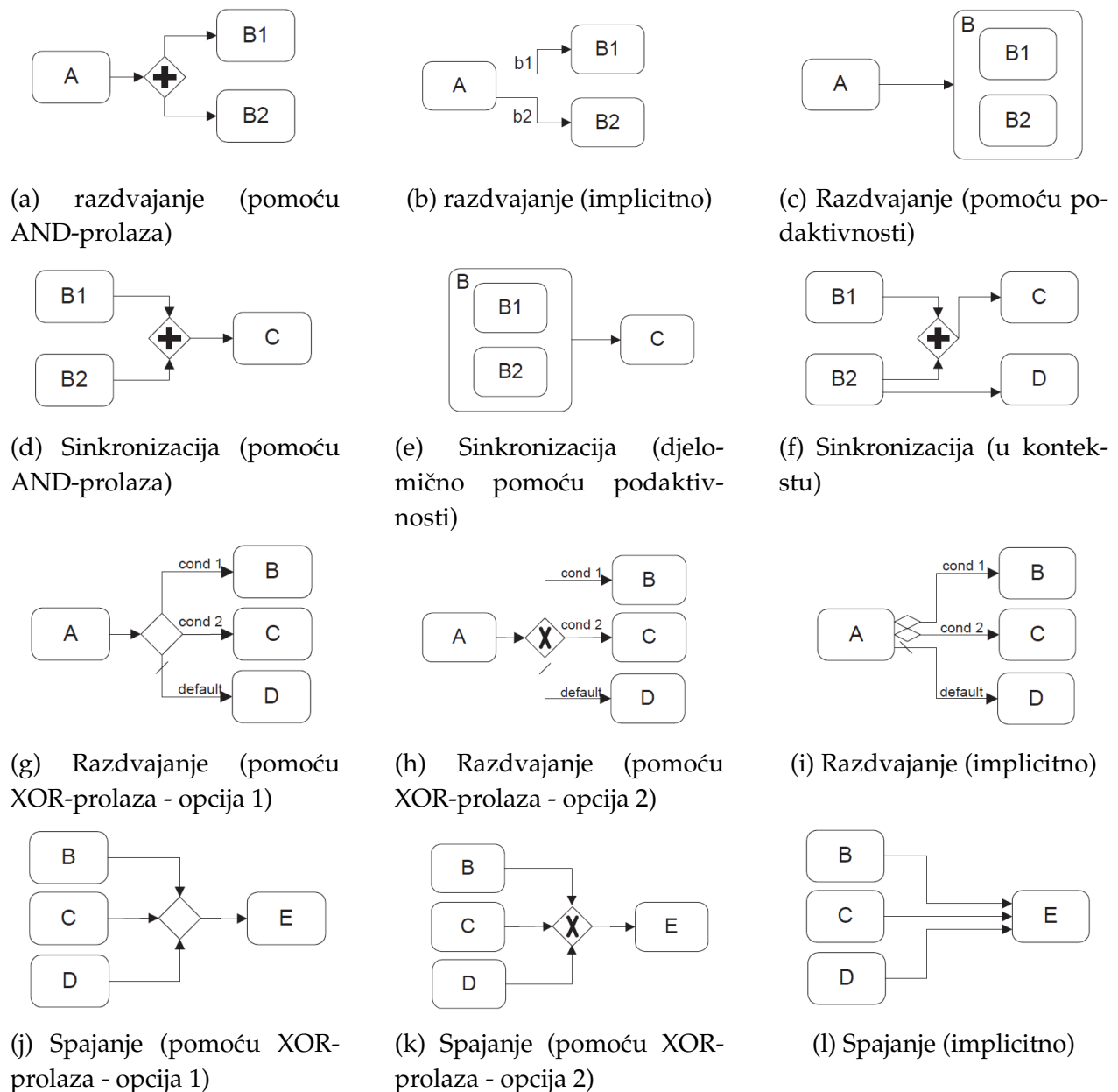
- događaji (eng. *events*),
- aktivnosti (eng. *activities*),
- pristupnici (eng. *gateways*),
- particije (eng. *swimlanes*),
- pomoćni objekti (eng. *artifacts*),
- objekti spajanja (eng. *connecting objects*).



Slika 3.5: Osnovni elementi BPMN-a (Chinosi i Trombetta, 2012)

Navedeni se elementi mogu povezivati na različite načine. Uobičajeni uzorci, koji se koriste kod modeliranja tijeka rada, navedeni su u Van der Aalst i ter Hofstede (2002). Slika 3.6 prikazuje kako se osnovni uzorci tijeka rada mogu prikazati pomoću BPMN-a.

Sa slike 3.6 može se primijetiti kako postoji više načina prikaza određenog uzorka tijeka rada.



Slika 3.6: BPMN vrste usmjeravanja tijeka rada (Van der Aalst, 1998)

### 3.4.2. UML dijagrami aktivnosti

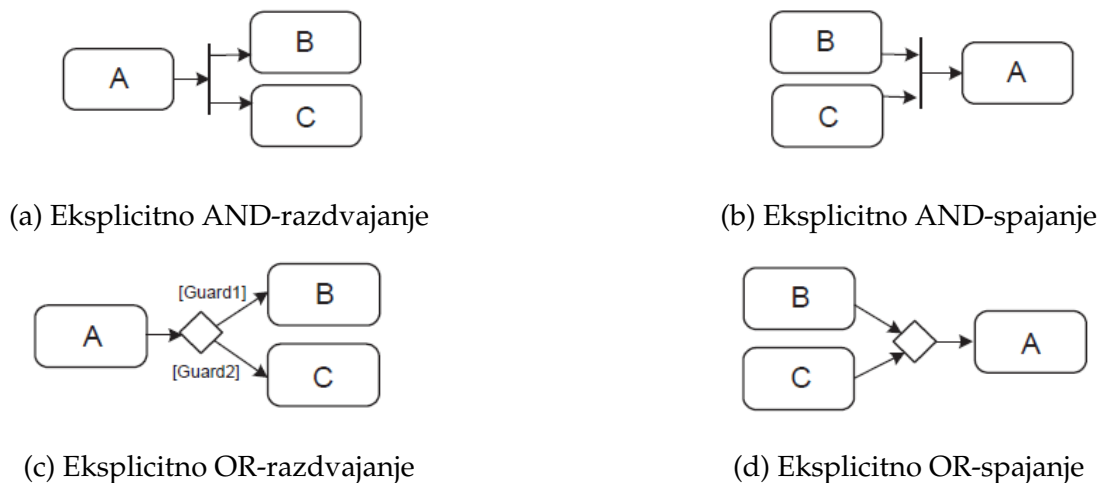
UML općenito pruža širok raspon notacija za opisivanje različitih pogleda računalnih struktura i ponašanja (primjerice, dijagrami klasa, slučajeva korištenja, objektni dijagrami) (OMG, 2017; ISO, 2012b). Za modeliranje poslovnih procesa mogu se koristiti UML dijagrami aktivnosti (*UML AD*) iako im to nije primarna svrha. UML AD mogu se koristiti



za modeliranje aktivnosti po slučajevima korištenja ili aktivnosti u korisničkim sučeljima (primjerice, redoslijed obavljanja operacija u računalnim programima). Za usporedbu, BPMN se koristi isključivo za modeliranje poslovnih procesa.

Osnovna jedinica rada koju UML AD propisuju je akcija (eng. *action*) koja uzima skup ulaznih podataka te ih transformira u skup izlaznih podataka. Ulazni i/ili izlazni skupovi podataka mogu biti prazni, odnosno moguće su akcije koje ne zahtijevaju rad s podacima. UML AD propisuju više od 40 različitih tipova akcija (primjerice, *Accept Event*, *Send signal*) koje omogućavaju slične funkcionalnosti kao i BPMN što potvrđuje White (2004). Aktivnost (eng. *Activity*) se sastoji od akcija ili drugih aktivnosti (dakle, moguće je ugnježđivanje aktivnosti). UML AD se sastoje od čvorova povezanih linijama. Čvorovi mogu predstavljati aktivnosti, akcije, podatkovne objekte ili kontrolne čvorove (čvorovi u kojima se tijek rada razdvaja ili spaja).

UML AD podržavaju osnovne uzorke tijeka rada (Slika 3.7), a pružaju ograničenu podršku za modeliranje nekoliko naprednijih uzoraka tijeka rada (*Synchronising merge*, *Multiple instance without a priori Runtime Knowledge*, *Interleaved Parallel Execution*, *Milestone*) (Wohed i dr., 2005). Russell i dr. (2006) proširuju istraživanje te ispituju prikladnost UML-a za modeliranje poslovnih procesa i uzoraka vezanih uz druge dimenzije tijeka rada (podaci i resursi).



Slika 3.7: Vrste usmjeravanja tijeka rada pomoću UML AD-a (Wohed i dr., 2005)

### 3.4.3. Petrijeve mreže

Petrijeve mreže (eng. *Petri net*) mogu se koristiti u raznim primjenama pa tako i za modeliranje poslovnih procesa, a uveo ih je uveo Carl Adam Petri u 1960-ima (Petri, 1962, 1966). Petrijeve mreže imaju snažnu matematičku osnovu, a grafička ih priroda čini jednostavnima za razumijevanje. Teoretski temelji BPMN i UML AD notacija potječu iz Petrijevih mreža.

Petrijeve mreže usmjereni su grafovi koji se sastoje od tri komponente: **mjesta** (eng. *place*), **prijelaza** (eng. *transition*) i **luka** (eng. *arc*). Mjesta se prikazuju kao krugovi i predstavljaju moguća stanja ili uvjete sustava. Prijelazi su prikazani pravokutnicima i opisuju zadatke (operacije) koji mogu promijeniti stanje sustava. Veze između mjesta i prijelaza ostvaruju se lukovima koji su jedina veza između mjesta i prijelaza u bilo kojemu smjeru, a veza između čvorova istoga tipa (mjesto-mjesto, prijelaz-prijelaz) nije moguća.

Petrijeve mreže matematički su definirane kao skup  $(P, T, F)$ , gdje je:

- $P$  konačan skup mjesta,
- $T$  konačan skup prijelaza ( $P \cap T = \emptyset$ ),
- $F \subseteq (P \times T) \cup (T \times P)$  skup lukova (povezanost tijeka (eng. *flow relation*)).

Mjesto  $p$  zove se **ulazno mjesto** prijelaza  $t$  ako postoji usmjereni luk od  $p$  prema  $t$ . Mjesto  $p$  naziva se **izlazno mjesto** prijelaza  $t$  ako postoji usmjereni luk od  $t$  prema  $p$ . U bilo kojemu trenutku mjesto može sadržavati više **oznaka** (eng. *token*) koje se označavaju crnom točkom unutar mjesta. **Stanje** predstavlja distribuciju oznaka po mjestima, odnosno  $M \in P \rightarrow \mathbb{N}$ . Primjerice, stanje procesa  $0p_1 + 2p_2 + 4p_3 + 0p_4$  znači da mjesto  $p_2$  ima dvije oznake, mjesto  $p_3$  četiri oznake dok ostala mjesta nemaju povezanih oznaka. Broj se oznaka mijenja tijekom izvođenja mreže budući da su prijelazi aktivne komponente, odnosno mijenjaju stanje Petrijeve mreže prema sljedećim pravilima aktivacije:

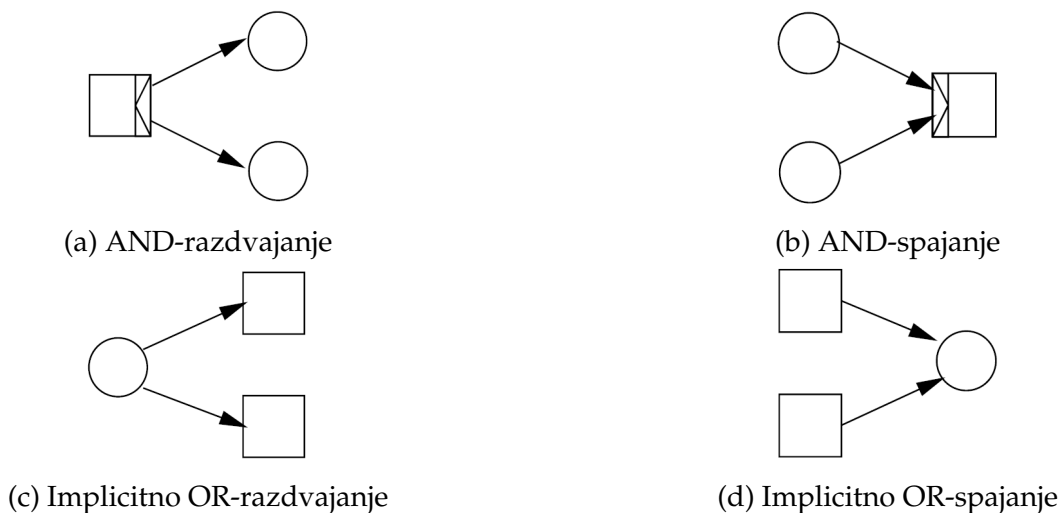
- prijelaz  $t$  se označava omogućenim za aktivaciju ako je svako ulazno mjesto  $p$  od  $t$  sadrži barem jednu oznaku,
- omogućeni se prijelaz  $t$  može aktivirati. Ako se  $t$  aktivira, konzumira jednu oznaku iz svakog ulaznoga mjesta  $p$  od  $t$  te stvara jednu oznaku na svakom izlaznome mjestu  $p$  od  $t$ .

Klasične Petrijeve mreže omogućavaju modeliranje stanja, događaja, uvjeta, sinkronizacije, paralelizma izbora i iteracije. S vremenom su proširivane kako bi se omogućile kompleksnije strukture (hijerarhijski procesi, vrijeme, podaci):

- Proširenje bojama za podršku modeliranja podataka. Bojom se označavaju karakteristike oznaka kako bi ih se moglo razlikovati. Moguće je definirati uvjete koji uzimaju boju oznake u obzir.
- Proširenje s podrškom za vrijeme. Vremenska je komponenta sustava bitna jer je potrebno modelirati trajanja i zastoje. Vrijeme može biti povezano s oznakama, mjestima i prijelazima.
- Podrška za hijerarhiju kako bi se mogli modelirati kompleksni modeli. Da bi se smanjila kompleksnost pojedinih modela, osmišljen je pojam podmreže (eng. *subnet*). Na najvišoj se razini daje opis procesa bez uzimanja detalja u obzir. Na se nižim razinama može detaljnije definirati tijek rada.

Petrijeve mreže proširene bojom, vremenom i hijerarhijom nazivaju se Petrijeve mreže visoke razine (eng. *high-level Petri net*), a prihvaćene su kao međunarodna norma ISO 15909 (ISO, 2004c). Prvi dio norme (ISO, 2004c) definira sintaksu i semantiku Petrijevih mreža, dok drugi dio (ISO, 2004d) definira razmjenski format (*Petri Net Markup Language (PNML)*) pomoću kojega je moguće razmjenjivati definicije poslovnih procesa ili općenitije, Petrijevih mreža. Modeliranje se procesa pomoću Petrijevih mreža izvodi u skladu s ranije navedenom terminologijom (WFMC, 1999): proces je definiran kao niz povezanih *zadataka* koji se modeliraju kao prijelazi, a uvjeti se modeliraju kao mjesta. Skup prijelaza i mjesta povezanih lukovima čini graf Petrijevih mreža, odnosno proces je iskazan grafom Petrijevih mreža. Svaki slučaj (eng. *case*) može biti u određenome stanju što se modelira pomoću oznaka (eng. *token*).

Petrijeve mreže koje se koriste za modeliranje procesa u sustavima za upravljanje tijekom rada nazivaju se mreže tijeka rada (ili kraće WF mreže). WF mreža zadovoljava dva preduvjeta. WF mreža ima jedno ulazno mjesto (*i*) i jedno izlazno mjesto (*o*). Kod WF mreža svi zadaci i uvjeti između ulaznog i izlaznoga mjesta moraju biti vezani s barem dva luka (na element ispred i na element iza), odnosno svaki prijelaz treba doprinijeti rješavanju slučaja (Van der Aalst, 1998).



Slika 3.8: Vrste usmjeravanja tijeka rada kod Petrijevih mreža (Van der Aalst, 1998)

Van Der Aalst (1996) navodi prednosti korištenja Petrijevih mreža s ciljem modeliranja tijeka rada:

- *formalna semantika unatoč grafičkoj prirodi.* Petrijeve mreže grafički su jezik, ali semantika je matematički definirana.
- *Temeljen na stanjima umjesto događajima.* Za razliku od drugih notacija za modeliranje procesa, Petrijeve mreže omogućavaju modeliranje stanja slučaja. Ostale se tehnike modeliranja procesa (poput UML AD) temelje na aktivnostima što znači da se zadaci modeliraju eksplicitno, a stanja se između pojedinih zadataka zanemaruju.

- *Obilje tehnika analize.* Petrijeve mreže se mogu analizirati pomoću mnogo dostupnih tehnika. Korištenjem tih tehnika može se ispitivati ispravnost definicije procesa (Murata, 1989).

Međutim, isto tako Petrijeve mreže imaju ograničenja kod modeliranja procesa (Van der Aalst i dr., 2003), odnosno modeliranja određenih naprednih uzoraka:

- uzorci koji uključuju višestruke instance procesa,
- napredni uzorci sinkronizacije,
- uzorci otkazivanja.

Navedena ograničenja ne znače da Petrijeve mreže visoke razine ne mogu izraziti takve uzorke. Međutim, potrebno je uložiti znatan trud u modeliranje takvih procesa te su grafovi Petrijevih mreža u takvim slučajevima kompleksni. Zbog toga Van der Aalst i ter Hofstede (2002, 2005) predlažu novi jezik za modeliranje poslovnih procesa Yet Another Workflow Language (YAWL) koji također ima formalnu osnovu temeljenu na Petrijevim mrežama, ali pruža širi skup elemenata za modeliranje procesa. Zbog formalne osnove YAWL-a provedena su istraživanja iz područja implementacije (Van Der Aalst i dr., 2004) i transformacije BPMN dijagrama u YAWL mreže (Decker i dr., 2008; Börger, 2012) s ciljem provjere ispravnosti poslovnih procesa.

#### 3.4.4. Osvrt na jezike za modeliranje procesa

BPMN i UML AD najčešće su korišteni jezici za modeliranje i prikaz poslovnih procesa. Iako veliki broj istraživanja potvrđuje da je BPMN prikladniji za modeliranje poslovnih procesa od UML AD-a (White, 2004; Russell i dr., 2006; Wohed i dr., 2006), Birkmeier i dr. (2010) ipak tvrde da su UML AD barem jednako prikladni kao i BPMN za modeliranje poslovnih procesa.

Međutim, nedostatak je BPMN-a i UML AD-a mogućnost ispitivanja formalne ispravnosti procesa. Zato Ramadan i dr. (2011) prikazuju formalizaciju BPMN-a pomoću obojenih Petrijevih mreža s ciljem omogućavanja automatske validacije poslovnih procesa. Druga je moguća primjena takve formalizacije omogućavanje simulacije poslovnih procesa s ciljem otkrivanja i otklanjanja pogrešaka u tijeku rada. Slična su istraživanja provedena u Decker i dr. (2008); Börger (2012) gdje su autori transformirali poslovne procese iz BPMN-a notacije u YAWL.

Jedan je od ciljeva ovoga doktorskog rada omogućavanje definiranja uvjeta ispravnosti prostorne sastavnice katastarskih čestica na razini vrste procesa te omogućavanje definiranja alternativnih tijekova rada. Za tu su svrhu dovoljni osnovni uzorci tijeka rada te su WF mreže odabrane zbog svoje jednostavnosti, solidne matematičke osnove te mogućnosti provjere formalne ispravnosti.

### 3.5. Transakcijski sustavi za upravljanje tijekom rada

Integracijom su se transakcijskih modela u WFMS pokušali smanjiti ili eliminirati nedostaci WFMS-a. Jedno je od prvih istraživanja vezanih uz modeliranje tijeka rada prikazano u Rusinkiewicz i Sheth (1995). U radu je objašnjena primjena naprednoga transakcijskog modela ConTract (Wächter i Reuter, 1992) koji omogućava definiciju tijeka rada kroz pisanje programske skripte. Georgakopoulos i Hornick (1994) predstavljaju okvir TSME koji služi za specifikaciju naprednih transakcijskih modela koji su primjenjivi u distribuiranim sustavima. Koncept je dalje razvijen u Georgakopoulos i dr. (1994). Navedeni radovi objašnjavaju dvije važne komponente WFMS-a, duge aktivnosti i distribuirane sustave te definiraju smjer za razvoj WFMS-a.

Alonso i dr. (1996) navode da su ciljevi WFMS-a sustava slični onima koje imaju napredni transakcijski modeli, međutim, napredni su transakcijski modeli koncentrirani na osiguravanje konzistentnosti podataka kroz transakcije te su vezani uz SUBP čime su ograničeni na korištenje samo funkcionalnosti koje pruža SUBP. Većina naprednih transakcijskih modela nije niti implementirana. Primjer je implementacije objašnjen u Barga i Pu (1995), ali takva implementacija nije dalje razrađena niti je ušla u širu upotrebu. Worah i Sheth (1997) tvrde da napredni transakcijski modeli ne mogu biti osnova za uspostavu sustava za upravljanje tijekom rada.

Osnovni i napredni transakcijski modeli podržavaju ACID svojstva. Iako postoje istraživanja vezana uz prilagođavanje svojstva atomičnosti (Derks i dr., 2001) i izolacije (Guabtni i dr., 2006) kod WFMS-a, takvi koncepti nisu dalje razrađeni niti implementirani. Eder i Liebhart (1994) navode da su ACID svojstva i serijalizabilnost kod upravljanja konkurentnim transakcijama previše ograničavajući za aktivnosti u WFMS-u te navode primjere kako bi to potkrijepili:

- *Dugo trajanje aktivnosti:* transakcije kod baza podataka podrazumijevaju kratko trajanje, dok aktivnosti kod WFMS-a uglavnom imaju duže trajanje, utječu na mnoge objekte te imaju kompleksan tijek rada. Provedba ACID transakcije dugoga trajanja može imati znatan negativan utjecaj na ostale transakcije.
- *Suradnja i konkurentnost kroz semantičku serijalizabilnost:* Aktivnosti mogu uključivati više korisnika (zajednički rad na istome skupu podataka) te oni mogu pristupati podacima, mijenjati ih, radi čega je potreban sustav sinkronizacije podataka. Serijalizabilnost kao kriterij ispravnosti konkurentnoga pristupa u takvim je slučajevima previše ograničavajuća. Konzistentnost se ipak može osigurati definiranjem uvjeta ispravnosti između aktivnosti. Primjer se definiranja semantičke serijalizabilnosti može pronaći u Breitbart i dr. (1993).
- *Korisnički definirana atomičnost u slučaju pada sustava:* Atomičnost (princip: sve-ili-ništa) je ograničavajuća i opsežna za aktivnosti u WFMS-u. Primjerice, nedopustivo je da se čitava aktivnost ili tijek rada otkáže jer se otkazala manja podaktivnost koja ne utječe na ostale podaktivnosti. Takve se situacije mogu riješiti uvođenjem točaka

spremanja koje će omogućiti da se tijek rada nastavi od zadnjega konzistentnog stanja tijeka rada, odnosno podrškom za oporavak u smjeru naprijed (eng. *forward recovery*). Uvođenjem se kompenzirajućih aktivnosti može osigurati i oporavak u smjeru natrag (eng. *backward recovery*) kojim se stanje podataka vraća u ono stanje koje je vrijedilo prije početka procesa.

- *Suradnja i konkurentnost kroz opuštenu izolaciju*: Ako podaktivnosti mogu čitati podatke druge aktivnosti koja nije provedena, moguća je veća razina konkurentnosti i suradnje. Međutim, ako se glavna aktivnost otkáže, potrebno je kompenzirati sve aktivnosti koje su provedene, a čitale su podatke neprovedenih aktivnosti. Ako kompenzirajuće aktivnosti postoje, tada opuštanje svojstva izolacije povećava konkurentnost. Međutim, nije moguće sve aktivnosti kompenzirati (primjerice, slanje obavijesti poštom naručitelju procesa). Da bi konzistentnost bila zajamčena, potrebno je osigurati da će se kompenzacija aktivnosti moći obaviti.

Kao moguće rješenje navedenih nedostataka autori predlažu korištenje transakcijskoga modela u WFMS-u za pouzdano upravljanje dugim, konkurentnim poslovnim procesima. Kako bi to bilo moguće, oni predlažu WFMS sa sljedećim karakteristikama: opuštena atomičnost (definiranje pravila atomičnosti), opuštena serijalizabilnost (korištenje semantičke serijalizabilnosti), opuštena izolacija (omogućavanje pristupa neprovedenim podacima).

Međutim, transakcijski se WFMS s takvim svojstvima ne može primijeniti na sustave za upravljanje zemljištem jer ispravnost procesa ima najveću važnost, a atomičnost i izolacija su svojstva koja pomažu da se ona osigura. Na razini katastarskih čestica opuštena atomičnost ili izolacija mogu ugroziti ispravnost katastarskih podataka. Omogućavanje djelomične provedbe procesa nad prostornom sastavnicom katastarskih čestica može uvesti neispravnosti u podatke. Zbog toga je potrebno osigurati ACID svojstva u transakcijskome WFMS-u koji se primjenjuje u sustavu za upravljanje zemljištem uz istovremeno povećanje fleksibilnosti procesa.

Kamath i Ramamritham (1996) navode općenite nedostatke WFMS-a vezane uz osiguravanje ispravnosti podataka u slučaju istovremenoga pristupa:

- *Koordinirani procesi*: primjerice, kod automatizacije uredskih poslova. Iako je poredak aktivnosti fiksna i WFMS to podržava, konkurentan pristup podacima može uvesti nekonzistentnosti. Atomičnost procesa može ograničiti konkurentnost procesa, odnosno WFMS pruža slabu podršku za konkurentan pristup.
- *kooperativni procesi*: CAD/CAM primjeri su ovakve vrste procesa gdje više korisnika radi izmjene na istim objektima. Poredak se aktivnosti određuje dinamički u trenutku izvođenja procesa. Potrebno je definirati uvjete ispravnosti i ovisnosti između aktivnosti kako bi se osigurala konzistentnost i oporavak procesa.
- *Integracija netransakcijskih aplikacija* poput aplikacije za uređivanje teksta ili tablica u transakcijskome WFMS-u problematična je, jer WFMS ne može takve aktivnosti kontrolirati (budući da nisu transakcije). Moguće je rješenje sprječavanje istovremenoga

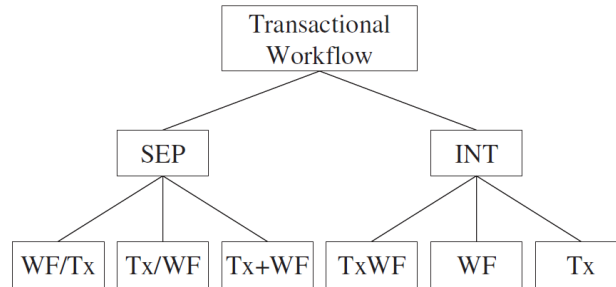
pozivanja aplikacije nad istim podacima.

- *Procesi u mobilnom okruženju*, odnosno podrška za rad bez veze na bazu podataka (eng. *offline mode*). Prijedlog je rješenja prikazan u Alonso i dr. (1995).
- *Upravljanje padovima sustava*. WFMS ne pružaju dobru podršku za upravljanje padovima. Dok neki sustavi omogućavaju oporavak u smjeru naprijed, za oporavak u smjeru natrag potrebno je uvesti kompenzacijske aktivnosti ili aktivnosti prema ACID svojstvima.
- *Dinamička promjena definicije procesa*, odnosno promjene tijeka rada nakon što proces započne.

Navedeni su nedostaci riješeni kod baza podataka i mogu se ublažiti ili eliminirati primjenom transakcijskoga WFMS-a, koji funkcionira prema ACID svojstvima.

Primjer implementacije transakcijskoga tijeka rada može se pronaći u Schick i dr. (2011). Autori su implementirali višeslojni sustav koji se temelji na jeziku za modeliranje YAWL. Sustav za osiguravanje fleksibilnosti koristi konfiguracijske datoteke temeljene na XML-u. Transakcijski se koncepti temelje na otvorenim ugniježđenim transakcijama, a upravljanje konkurentnim procesima na višestrukim verzijama (Muro i dr., 1984). Model definira osnovne transakcije koje moraju osigurati serijalizabilnost i oporavak. Ugniježdene transakcije omogućavaju sastavljanje transakcija od raznih podtransakcija. Moguće je definirati skup potencijalnih transakcija gdje samo jedna od transakcija mora biti uspješna. Moguće je i definirati skup (sferu) transakcija za koji vrijede određeni uvjeti ispravnosti. Još je jedan bitan koncept: uvođenje nebitne transakcije (eng. *non-vital*) čije otkazivanje ne utječe na uspjeh cjelokupnoga procesa što se može primijeniti na sustave za upravljanje zemljištem. Primjerice, jedan od nositelja prava na susjednoj katastarskoj čestici ne može biti obaviješten, jer mu je nepoznata adresa prebivališta. Ako se na takav proces primjeni svojstvo atomičnosti, tada bi neuspješna provedba toga zadatka uzrokovala otkazivanje čitavoga procesa. Ako se takav zadatak definira kao nebitan, proces se može provesti čak i u slučaju da se taj nebitni zadatak ne provede uspješno. Predloženi sustav omogućava znatno fleksibilniju strukturu od naprednih transakcijskih modela, ali sustav se previše bazira na transakcijske modele što smanjuje mogućnosti u smislu korištenja netransakcijskih aktivnosti. Isto tako, u radu nije obrađen način raspodjele pojedinih transakcija korisnicima, odnosno podržava li i na koji način višekorisnički rad.

Kroz vrijeme su provedena razna istraživanja i modeli integracije transakcijskih koncepata i WFMS-a. Stoga su Grefen i Vonk (2006) napravili sistematizaciju mogućih načina njihove integracije u odnosu na konceptualnu i arhitekturnu (implementacijsku) razinu (Slika 3.9). Autori definiraju osnovnu podjelu na integrirane modele (INT) kod kojih postoji jedinstveni model za definiciju tijeka rada i transakcijskih koncepata te odvojene modele (SEP) kod kojih se koriste dvije odvojene specifikacije, jedna za tijek rada i druga za transakcijske koncepte.



Slika 3.9: Sistematizacija transakcijskih WFMS-a (Grefen i Vonk, 2006)

Kod modela *WF/Tx* tijek je rada apstraktniji od transakcija, odnosno transakcije su sastavni dijelovi zadataka. Tijek rada upravlja redoslijedom i načinom pozivanja pojedinih zadataka, odnosno transakcija. Na slici 3.10a s desne je strane prikazana definicija transakcije koja čita vrijednosti iz forme za unos podataka i zatim zapisuje promjene. Ovisno o uspjehu operacije zapisivanja transakcija se provodi ili odbacuje. Tijek rada definira strukturu zadatka koja se sastoji od transakcije i forme za unos podataka. Tijek rada osigurava da se podaci proslijede transakciji, ali nije upoznat s unutarnjom strukturom transakcije.

Kod modela *Tx/WF* transakcije su apstraktnije, odnosno tijek rada definira strukturu procesa, a transakcijski model definira transakcijska svojstva tijeka rada. Na slici 3.10b prikazan je tijek rada koji se sastoji od četiri sekvencijalna zadatka. Tijek je rada integriran u transakcijski model te je na čitavi tijek rada primijenjeno svojstvo atomičnosti. Jedan je od nedostataka ovoga pristupa što kod kompleksnih i dugih procesa direktna primjena ACID svojstava može značajno smanjiti konkurentnost procesa.

Model *Tx+WF* sadrže dvije odvojene specifikacije na istoj razini apstrakcije. Tijek rada i transakcijski model podmodeli su implicitnoga, slabo povezanoga modela procesa. Na slici 3.10c s desne je strane prikazana definicija tijeka rada koja se sastoji od tri sekvencijalna zadatka. U definiciji je tijeka rada navedena veza na specifikaciju transakcijskoga modela. Transakcijskim je modelom definirana i ovisnost u drugome smjeru prema tijeku rada te je definirano konzistentno stanje (točka spremanja) nakon provedenoga prvog zadatka. Definicijom je transakcije određeno da se učinak pojedine transakcije može poništiti pozivanjem kompenzacijskih zadataka. Prednost je ovoga modela što se definicija transakcijskih svojstava može promijeniti neovisno od definicije tijeka rada.



WFDL	TxDL
BEGIN TASK $task_1$	BEGIN TRANSACTION
BUSINESS TRANSACTION	READ $form_1.field_1$
USES FORM $form_1$	READ $form_1.field_2$
END TASK	USE $form_1$
	WRITE $form_1.field_1$
	WRITE $form_1.field_2$
	IF $status\_ok$
	THEN COMMIT TRANSACTION
	ELSE ABORT TRANSACTION
	END TRANSACTION

(a) Model WF/Tx

TxDL	WFDL
BEGIN TRANSACTION $tr_1$	BEGIN WORKFLOW $wf_1$
EXECUTE ATOMIC	TASK $task_1 task_2 task_3 task_4$
IMPLEMENTATION $wf_1$	SEQUENCE $task_1 task_2$
END TRANSACTION	SEQUENCE $task_1 task_3$
	SEQUENCE $task_2 task_4$
	SEQUENCE $task_3 task_4$
	END WORKFLOW

(b) Model Tx/WF

TxDL	WFDL
BEGIN TRANSACTION $tr_1$	BEGIN WORKFLOW $wf_1$
REFERS WORKFLOW $wf_1$	REFERS TRANSACTION $tr_1$
COMP $ctask_1 task_1$	TASK $task_1 task_2 task_3$
COMP $ctask_2 task_2$	SEQUENCE $task_1 task_2$
SAFEPOINT $task_1$	SEQUENCE $task_2 task_3$
END TRANSACTION	END WORKFLOW

(c) Model Tx+WF

TxWFDL
BEGIN WORKFLOW $wf_1$
TASK $task_1$ COMP $ctask_1$
TASK $task_2$ COMP $ctask_2$
TASK $task_3$ COMP $none$
SEQUENCE $task_1 task_2$
SEQUENCE $task_2 task_3$
SAFEPOINT $task_1$
END WORKFLOW

(d) Model TxWF

WFDL
BEGIN WORKFLOW $wf_1$
TASK $task_1 task_2 task_3$ # regular tasks
TASK $ctask_1 ctask_2 ctask_3$ # compensating tasks
SPLIT $or_1 or_2 or_3$
SEQUENCE $task_1 or_1$ # start regular control flow
SEQUENCE $or_1 task_2$
SEQUENCE $task_2 or_2$
SEQUENCE $or_2 task_3$
SEQUENCE $or_1 ctask_1$ # start compensation control flow
SEQUENCE $or_2 ctask_2$
SEQUENCE $or_3 ctask_3$
SEQUENCE $ctask_3 ctask_2$
SEQUENCE $ctask_2 ctask_1$
SEQUENCE $ctask_1 task_1$ # start regular control flow
END WORKFLOW

(e) Model WF

TxDL
BEGIN TRANSACTION $tr_1$
SUBTRANSACTION $s_1$
$action_1$
$action_2$
END SUBTRANSACTION
SUBTRANSACTION $s_2$
$action_3$
$action_4$
END SUBTRANSACTION
PARALLEL $s_1 s_2$
END TRANSACTION

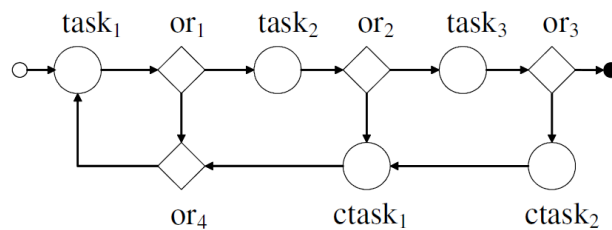
(f) Model Tx

Slika 3.10: Modeli integracije koncepta transakcije i WFMS (Grefen i Vonk, 2006)

Model  $TxWF$  predstavlja hibridni model koji sadrži koncepte transakcije i tijeka rada. Model predstavlja jedinstvenu specifikaciju procesa spajanjem modela  $Tx$  i  $WF$  (Slika 3.10d).

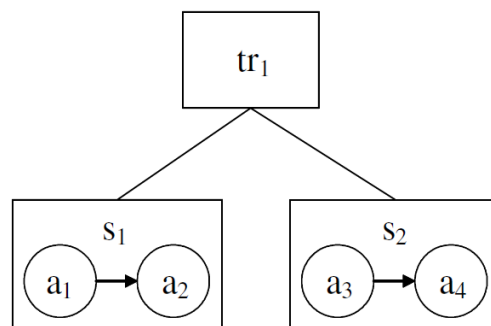
Model  $WF$  koristi jedinstvenu definiciju tijeka rada, a transakcijski se koncepti izražavaju pomoću elemenata tijeka rada (primjerice, pomoću uzoraka za kontroliranje tijeka rada sa slike 3.3 i kompenzirajućih zadataka). Kompenzacija se procesa može implementirati pomoću elemenata tijeka rada, razdvajanjem tijeka rada na alternativne tijekove (OR-dijeljenje) koji će nastupiti ako se provjerom utvrdi je li potrebno otkazivanje i poništavanje procesa. Kod ovoga je modela bitno naglasiti da se provjera je li zadatak uspješno obavljen događa tek nakon što zadatak završi, a pogreška se vjerojatno dogodila prilikom izvođenja zadatka. Konzistentnost se može osigurati primjenom ACID svojstva na svaki zadatak koja će onemogućiti djelomičnu provedbu zadatka. Uvođenjem se alternativnih

tijekova rada mogu eliminirati razlozi za odbacivanje procesa. Na slici 3.11 prikazan je primjer procesa definiranoga na slici 3.10e te način poništavanja promjena određenoga zadatka pomoću alternativnih tijekomova rada i kompenzirajućih zadataka. Uvođenjem točaka spremanja može se omogućiti da proces nastavi od zadnjega konzistentnog stanja u slučaju pogreške ili pada sustava, odnosno nije potrebno svaki put nakon pada sustava početi proces ispočetka.



Slika 3.11: Kompenzacija zadataka prema modelu WF (Grefen i Vonk, 2006)

Model  $Tx$  koristi jedinstveni transakcijski model, a koncepti tijeka rada izražavaju se kroz transakcije. Kod  $Tx$  modela semantika je tijeka rada implicitno izražena specifikacijom transakcije. Na slici 3.12 prikazana je transakcija koja se sastoji od dvije podtransakcije koje se mogu provesti istovremeno, a specifikacija je same transakcije prikazana na slici 3.10f. Napredni transakcijski modeli mogu sadržavati naprednije uzorke upravljanja tijekom rada. Što su napredniji uzorci upravljanja tijekom rada, to se  $Tx$  model približava modelu  $TxWF$ . Struktura  $Tx$  modela uglavnom je u obliku blokova, dok je struktura  $WF$  i  $TxWF$  više u obliku grafova.



Slika 3.12: Primjer transakcije s jednostavnom strukturom (Grefen i Vonk, 2006)

Postojeći komercijalni WFMS-i pružaju najbolju podršku za model  $WF/Tx$  iako je podrška za transakcijska svojstva ograničena i moguća jedino na razini pojedinih zadataka. Uz ovaj, model  $WF$  također je prihvatljiv za primjenu u postojećim komercijalnim sustavima za upravljanje tijekom rada jer ne zahtijeva definiranje novih elemenata niti uvođenje dodatnih transakcijskih elemenata. Transakcijski se elementi implementiraju pomoću elemenata tijeka rada. U posljednje se vrijeme koristi model  $Tx+WF$  koji također pruža

ograničenu transakcijsku podršku u okruženjima koja koriste web servise. Primjer je implementacije jezik Business Process Execution Language (BPEL) koji služi za definiciju načina i redoslijeda pozivanja web servisa. Iako BPEL ne podržava transakcije, razne ih implementacije temeljene na BPEL-u podržavaju (poput Oracle, SAP, IBM). Model Contract (Rusinkiewicz i Sheth, 1995) i TSME (Georgakopoulos i Hornick, 1994) primjeri su *Tx* modela, dok je primjer *WF* modela objašnjen u Dehnert (2001).

Prednosti su odvojenih sustava podjela odgovornosti i modularnost, dok je glavni nedostatak integracija dvije odvojene specifikacije koje definiraju strukturu tijeka rada, odnosno transakcija. Prednost je integriranih modela jednostavna formalizacija i konzistentnost specifikacije dok je glavni nedostatak ograničena izražajnosti procesa, jer transakcijski koncepti ne podržavaju dovoljnu razinu fleksibilnosti kod modeliranja procesa (kod modela *Tx*) i obratno, elementi tijeka rada su ograničeni transakcijskim konceptima te su dijagrami kompliciraniji zbog osiguravanja transakcijskih svojstava (kod modela *WF*). U ovome se radu modeliranje procesa nad katastarskim česticama koristi za definiciju fleksibilnoga načina definiranja uvjeta ispravnosti različitih vrsta procesa te su za potrebe ovoga rada dovoljni osnovni uzorci tijeka rada. Zato se u ovom radu koristi model *WF* kojim je moguće izraziti osnovne uzorke tijeka rada bez kompliciranja ili smanjivanja izražajnosti dijagrama tijeka rada. Isto tako, model *WF* podržava i netransakcijske zadatke (primjerice, pozivanje aplikacije za uređivanje teksta) unutar istoga tijeka rada.

## 3.6. Prostorni sustavi za upravljanje tijekom rada

U domeni primjene WFMS-a na upravljanje ili rad s prostornim podacima u odnosu na domenu neprostornih podataka postoji relativno malo istraživanja. Istraživanja primjene WFMS-a na prostorne podatke započelo je primjenom u znanstvenim istraživanjima.

Alonso i Hagen (1997) proširuju ranije razvijeni općeniti sustav upravljanje tijekom rada OPERA s podrškom za prostorne podatke i nazivaju ga Geo-Opera. Sustav Geo-Opera nadograđuje koncept upravljanja tijekom rada kako bi se omogućila distribuirana i heterogena platforma za razvijanje i upravljanje kompleksnim geografskim modelima u znanstvenim istraživanjima. Razvijeni sustav omogućava fleksibilno upravljanje istraživanjima primjenom WFMS-a te sadržava funkcionalnosti modeliranja procesa, upravljanja iznimkama te izvođenje upita nad podacima i metapodacima. Sličan je pristup objašnjen i u Weske i dr. (1998). Kroz prikaz slučajeva korištenja autori pokazuju kako je moguće koristiti WFMS u provođenju znanstvenih istraživanja u geoznanostima gdje svaki podatak ima prostornu komponentu. U radu primjenjuju ranije razvijeni prototip WFMS-a za korištenje kod znanstvenih istraživanja opisan u Vossen i dr. (1996).

Chai i dr. (2008) primjenjuju koncept WFMS-a na učinkovito upravljanje prikupljanjem seizmičkih podataka iz dvadesetak gradova u Kini. Autori primjenjuju gotovo programsko rješenje i nadograđuju ga kako bi bolje odgovaralo njihovim potrebama. Naglasak stavljaju na upravljanje radom korisnika na prostornim podacima te njihovu koordinaciju.

Budući da je glavna svrha implementiranoga sustava koordinirano prikupljanje prostornih podataka i njihovo kontroliranje od strane korisnika, autori naglašavaju nedostatke automatske kontrole te naglašavaju važnost ljudske intervencije kod procesa kontroliranja prostornih podataka što je u suprotnosti s većinom istraživanja koja se bave definiranjem uvjeta ispravnosti s ciljem automatske kontrole (Choi i dr., 1999; van Oosterom, 1997; Vranić i dr., 2015). Yue i dr. (2015) primjenjuju koncept upravljanja tijekom rada na povezivanje raznih senzora i servisa za obradu prostornih podataka koji su u skladu s OGC OpenMI standardom (OGC, 2004). Još je jedan primjer implementacije prostornoga WFMS-a koji omogućava obradu prostornih podataka prikazan u Du i Cheng (2017).

Osim primjene WFMS-a na postupke rada s prostornim podacima, postoje i istraživanja u domeni sustava za upravljanje zemljištem, ali ona ne uključuju tehnološke niti formalne zahtjeve WFMS-a u takvim sustavima, već stavljaju težište na katastarske podatke i postupke, a WFMS je ponuđen kao moguće rješenje postavljenih problema. Projekt *COST Action G9 'Modelling Real Property Transactions'* (Zevenbergen i dr., 2007) nije vezan uz koncept WFMS-a, ali daje solidnu osnovu za modeliranje procesa u sustavima za upravljanje zemljištem. U okviru je projekta napravljena komparativna analiza ekonomske učinkovitosti transakcija u prometu nekretninama te su analizirane veze između troškova transakcija i nacionalnih praksi vezanih uz nekretnine. Bitni su rezultati ovoga projekta: definirana terminologija, kriteriji ispravnosti transakcija (pravni i tehnički), učinak transakcija (direktni i posredni) i definirane opće faze procesa nad nekretninama.

Ferlan i dr. (2007) razdvajaju pravni dio prometa nekretnina (oglašavanje, predugovor, ugovor i registracija) od stvaranja nove nekretnine koji predstavlja tehnički proces (primjerice, diobu zemljišta). Opće su faze procesa formiranja stvaranja nekretnine:

- Kontrola zemljišne politike. U ovoj se fazi kontrolira usklađenost formirane nekretnine s pravilima zemljišne politike, primjerice, ograničavanje veličine i oblika čestice ili vrste njezinoga korištenja.
- Priprema procesa. Uključuje definiciju novih granica koje moraju biti izmjerene. Moguće je stvaranje novih prava (primjerice, pravo prolaza) ili brisanje postojećih prava. Pravni i tehnički dio pripreme procesa mogu biti odvojeni potproces.
- Odluka katastra. Formiranje nezavisne pravne jedinice, odnosno nove nekretnine čin je koji mora biti zakonski definiran, odnosno mora se donijeti neki oblik odluke kojom se određuje vrijeme stupanja na snagu određene izmjene.
- Registracija. Nekretnine se registriraju kako bi prostorni obuhvat i sadržaj nove nekretnine bio poznat tržištu i tijelima javne vlasti.

Navratil i Frank (2004) analiziraju katastarske sustave (vrste, organizaciju, dokumente, dionike) te procese koji se odvijaju u katastarskim sustavima. Na najvišoj se razini razlikuju dvije vrste procesa: zapisivanje (eng. *inscription*) i dohvaćanje (eng. *retrieval*) podataka. Zapisivanje se može podijeliti u tri kategorije: prijenos ili stvaranje prava, uklanjanje prava i promjena podataka. Zapisivanje se sastoji od nekoliko koraka:

- Zaprimanje. Korisnik podnosi zahtjev za promjenu.

- Formalna provjera. Operator u katastru kontrolira potpunost podataka i u ovome bi se koraku trebao riješiti visoki postotak pogrešaka.
- Registracija. Operator u katastru dodaje dokumente, dodjeljuje im identifikatore, ali se označava tako da se jasno vidi da dokument nije prošao detaljnu provjeru.
- Detaljna provjera. Operator detaljno testira ispravnost. Ovisno o rezultatu provjere, autor može predane dokumente ispraviti i ponovno testirati ili dokumenti mogu biti uklonjeni iz registra.
- Zapisivnje promjene u bazu podataka. Ukoliko dokumenti utječu na dijelove baze podataka i zahtijevaju promjene, potrebno ih je zapisati u bazu podataka. Primjerice, brisanje prava postavlja status prava iz važećega na izbrisano, ali se dokument na temelju kojega je pravo bilo važeće, ne briše.

Hespanha (2012) faze procesa definirane u Ferlan i dr. (2007) dodatno razrađuje te ih primjenjuje na modeliranje procesa ažuriranja katastarskih podataka u Portugalu, ali integrira pravnu i tehničku komponentu u jedinstveni tijek rada:

- faza pripremanja,
- kontroliranje zemljišne politike,
- faza formalizacije,
- faza odluke i registracije.

Slično je istraživanje proveo Sari (2010), gdje analizira postupak održavanja katastarskih podataka u Indoneziji. Autor postavlja zaključak da trenutno stanje ne zadovoljava potrebe korisnika i predlaže novi dizajn tijeka rada održavanja katastarskih podataka temeljen na LADM-u. Autor se ne bavi tehnološkim aspektima, već na primjeru diobe čestica izrađuje dijagram aktivnosti te objašnjava koje LADM klase sudjeluju u procesu.

Phoung (2011) primjenjuje WFMS s ciljem povećanja transparentnosti postupaka u sustavima za upravljanje zemljištem. Zaključuje kako tradicionalni koncept transakcije koji se koristi u bazama podataka nije prikladan za konkurentne i duge procese, ali ne ulazi u tehnološke detalje. Nadalje, prepoznaje zahtjeve transparentnosti koji se postavljaju na postupke nad zemljištem:

- strukturalni: jasna podjela odgovornosti i uloga dionika,
- funkcionalni: javno dostupne informacije, otvoreni programi i procesi, provjere podataka te tehnološka sigurnost,
- transakcijski: ACID svojstva problematična su kod dugih transakcija. Izolacija se može osigurati verzioniranjem podataka.

WFMS pruža dovoljno fleksibilnosti kako bi podigao razinu transparentnosti postupaka te smanjio pojavu korupcije u katastru kao što je navedeno u Phoung (2011). Zahtjevi transparentnosti, iako ne donose tehnološka rješenja za uspostavu WFMS-a u sustavima za upravljanje zemljišta, daju smjernice za razvoj takvoga sustava te općenite funkcionalnosti koje bi taj sustav trebao podržavati.

### 3.7. WFMS i učinkovitost poslovnih procesa

Učinkovitost je širok pojam koji se često mjeri na određeni način. U tome je smislu navedena definicija u Anić (2006) kao sposobnost postizanja rezultata ili kapacitet proizvodnje, odnosno korisnoga djelovanja. Becker i zur Mühlen (1999) definiraju kriterije kojima se može mjeriti učinkovitost upravljanja poslovnim procesima od kojih su dva relevantna za ovaj rad: učinkovitost procesa i učinkovitost resursa. Prvi kriterij definira smanjivanje vremena trajanja procesa ili poštivanje zadanih rokova procesa, dok drugi kriterij definira optimalnu raspodjelu zadataka između dionika kako bi se potpomoglo ispunjenje prvoga kriterija. Smanjivanjem trajanja procesa povećava se njegova učinkovitost. Dumas i dr. (2013) detaljnije razrađuju učinkovitost procesa te obrađuju različite kriterije učinkovitosti procesa (financijske, učinkovitost pojedinoga zadatka, inovacijski i korisnički). Za ovaj su rad bitni samo kriteriji kojima se mjeri učinkovitost pojedinoga zadatka. U tu svrhu autori dijele svaki zadatak na dva dijela: vrijeme čekanja i vrijeme obrade. Omjer vremena obrade i ukupnoga trajanja zadatka definira kriterij učinkovitosti zadatka. Dakle, smanjivanjem se vremena čekanja na pojedinome zadatku povećava učinkovitost procesa.

Sustavi za upravljanje tijekom rada uvedeni su kako bi se omogućilo učinkovitije upravljanje poslovnim procesima (WFMC, 1995). Uz osnovne se komponente WFMS-a, rano pojavila ideja o poboljšavanju poslovnih procesa (BPR) koji podrazumijeva kontinuirani proces ispitivanja potreba korisnika, definiranja tijeka rada, analize učinkovitosti i poboljšavanja poslovnih procesa (Georgakopoulos i dr., 1995; Grover i dr., 1995). Van Osch i Lemmen (2004) opisuju trenutno stanje, potrebu i plan uvođenja sustava za upravljanje tijekom rada u nizozemskome katastru kako bi se podigla učinkovitost sustava te zadovoljstvo korisnika. Uvođenjem WFMS-a u nizozemski katastar autori očekuju povećanje učinkovitosti, pouzdanosti postupka održavanja podataka i fleksibilnosti kod upravljanja osobljem budući da će biti moguće raspoređivati poslove između pojedinih lokalnih ureda što će dovesti do rasterećenja osoblja unutar sustava.

Učinkovitost je WFMS-a u sustavu za upravljanje zemljištem u ovome radu sagledana s dva aspekta: učinkovitost upravljanja poslovnim procesima te učinkovitost upravljanja prostornim podacima. Kao što Van Osch i Lemmen (2004) navode, samom je primjenom WFMS-a u sustavu za upravljanje zemljištem moguće povećati učinkovitost zbog širokoga skupa funkcionalnosti WFMS-a. Različite su notacije za modeliranje i prikaz poslovnih procesa razvijene te omogućavaju standardizirani prikaz i razumijevanje poslovnih procesa. U skladu s notacijama razvijeni su razmjenski formati koji omogućavaju strukturiranu razmjenu definicije procesa (primjerice, XDPL ili PNML). Katastarski su podaci povezani s različitim skupovima podataka te druge organizacije često koriste, ali i između vlastita ograničenja na katastarske podatke. Korištenjem je web servisa i protokola za njihovu međusobnu komunikaciju (primjerice, Simple Object Access Protocol (SOAP)) moguće povećati učinkovitost komunikacije između različitih institucija i sustava. WFMS podržava sudjelovanje različitih dionika s različitim pravima prilikom izvođenja procesa. Kvalitetnom je raspodjelom zadataka moguće podijeliti opterećenje između resursa te na

taj način ubrzati provedbu procesa. Russell i dr. (2005) navode uobičajene uzorke koji se pojavljuju kod dodjeljivanja zadatka resursu. Razmjena podataka, mogućnost sudjelovanja različitih dionika na procesu te komunikacija putem web servisa funkcionalnosti su koje mogu utjecati na smanjenje trajanja procesa, odnosno na povećanje njegove učinkovitosti.

Iako prostorna sastavnica temeljena na poligonima sadrži redundantne podatke, ipak ima prednosti u odnosu na topološku strukturu u smislu razmjene podataka u razmjenskim formatima poput GML-a. Uvoz promjena pripremljenih izvan sustava je lakši s poligonima jer se uvoze svi potrebni podaci i nisu potrebne dodatne operacije kako bi se uspostavile različite veze između podataka kao što je to slučaj s topološkom strukturom. U kombinaciji s podrškom za upravljanje korisnicima s različitim pravima moguće je dio procesa predati vanjskim dionicima (primjerice, pripremanje se i provjera procesa može dodijeliti mjernicima, a pravo stvaranja novoga procesa nad katastarskim česticama njihovim vlasnicima). Na taj se način može dodatno rasteretiti sustav za upravljanje zemljištem. Isto se tako, automatizacijom kontrole ispravnosti prostorne sastavnice može smanjiti trajanje procesa, a primjenom je prikladnoga protokola za upravljanje konkurentnim procesima moguće smanjiti vrijeme čekanja jednoga procesa na podatke koji su zaključani u drugome procesu što kumulativno utječe na povećanje učinkovitosti procesa.

## 3.8. Rekapitulacija

Jedna je od glavnih razlika između WFMS-a i transakcijskih modela u području ispravnosti i pouzdanosti. Transakcijski su modeli nastali na temeljima formalne osnove relacijskoga modela baza podataka dok su sustavi za upravljanje tijekom rada nastajali kao potreba tržišta da nadomjesti nedostatke transakcijskih modela. Zato su u početku mnogi WFMS-i nastali bez sustavne analize potreba korisnika i formalne osnove. U početku su WFMS-i implementirani za male organizacije, no kada su korisnici prepoznali njihov potencijal, pokušali su ih primijeniti na većemu broju korisnika što WFMS-i nisu mogli podnijeti zbog ograničenja arhitekture (jedna baza podataka umjesto distribuiranih baza podataka, heterogeni sustavi i slaba podrška za međusobnu komunikaciju). I dalje je jedan od nedostataka WFMS-a nedovoljna podrška za oporavak i upravljanje padovima. Istraživanja vezana uz transakcijske WFMS-e bave se rješavanjem tih problema. Grefen i Vonk (2006) sistematiziraju moguće načine integracije WFMS-a i transakcijskih koncepta. U ovome je radu to istraživanje iskorišteno za prepoznavanje općih svojstava njihove integracije.

Dosadašnja se istraživanja u domeni prostornih WFMS-a bave primjenom koncepta WFMS-a na specijalizirane primjene WFMS-a na određene slučajeve korištenja. Glavni su ciljevi takvih primjena omogućavanje suradnje višestrukih dionika te povezivanje jednostavnijih operacija nad prostornim podacima kako bi se obavile kompleksne transformacije nad podacima. Istraživanja iz područja sustava za upravljanje zemljištem stavljaju težište

na katastarske podatke i procese, a WFMS je predstavljen kao moguće rješenje postavljenoga problema. Nijedan od analiziranih radova ne uključuje primjenu koncepta transakcijskoga WFMS-a koji bi općenito mogao biti primijenjen na kontroliranje ispravnosti prostornih ili katastarskih podataka. Primjenom je transakcijskoga WFMS-a moguće osigurati ispravnost katastarskih podataka koja je neophodna u sustavima za upravljanje zemljištem. Moguće je i povećati njegovu učinkovitost smanjenjem trajanja procesa omogućavanjem različitim dionicima sudjelovanje na procesu ili podrškom za komunikaciju putem web servisa. Vrijeme čekanja na početak obavljanja zadatka zbog zaključanosti podataka u drugim procesima može se smanjiti primjenom prikladnoga protokola za upravljanje konkurentnim procesima.



## DIO II:

# KONCEPTUALNI MODEL I KORIŠTENJE TESTNOGA SUSTAVA

*U ovome je dijelu rada predstavljen konceptualni model koji omogućava modeliranje procesa nad katastarskim podacima pomoću koncepta transakcijskoga WFMS-a što osigurava konzistentnu transformaciju podataka iz jednog u drugo stanje. Poseban je naglasak i osvrt stavljen na prostornu sastavnicu katastarskih čestica. Primjenom je različitih protokola za upravljanje konkurentnim procesima kao i uvođenjem međnih točaka u model podataka moguće ostvariti različite razine konkurentnosti procesa. Isto je tako pomoću testnoga sustava kroz konkretne primjere provjerena ispravnost hipoteza postavljenih na početku istraživanja.*

## 4. Primjena WFMS-a na katastarske podatke

Prostorna sastavnica katastarskih podataka definira prostorni opseg raznih interesa (eng. *Rights, Restriction, Responsibility (RRR)*) koji se koriste kao osnova u raznim primjenama poput gospodarenja zemljištem ili kupoprodaje nekretnina te je zbog toga važna njihova ispravnost. Katastarski se podaci svakodnevno mijenjaju kroz različite poslovne procese koji moraju biti učinkoviti i jamčiti ispravnost podataka u svakome trenutku. Nadalje, razvoj tehnologije i sve veće potrebe za katastarskim podacima zahtijevaju fleksibilnost poslovnih procesa i njihove česte promjene u skladu s razvojem tehnologije i potrebama korisnika.

Podacima se u bazama podataka upravlja kroz transakcije u skladu s ACID svojstvima čime je osigurana ispravnost podataka. Transakcijski su modeli previše vezani uz bazu podataka i ne pružaju dovoljnu fleksibilnost kod modeliranja poslovnih procesa. Stoga su istraživanja krenula prema razvijanju sustava za upravljanje tijekom rada (WFMS). Glavna je prednost WFMS-a fleksibilnost, a glavni su nedostaci slaba podrška za osiguravanje ispravnosti i oporavka procesa. Zbog toga su u ovom radu integrirani transakcijski koncepti u WFMS, a proces osigurava ispravnost podataka u skladu s ACID svojstvima. ACID svojstva procesa postižu se pomoću elemenata tijekom rada prema modelu *WF (Workflow)* (Grefen i Vonk, 2006) koji podržava transakcijske i netransakcijske zadatke (poput pozivanja aplikacije za uređivanje teksta s ciljem izrade računa ili pozivanja web servisa). Time je osigurana fleksibilnost, ispravnost i oporavak procesa.

Za modeliranje se procesa koriste WF mreže (Van der Aalst, 1998), formalni grafički jezik, zbog solidne matematičke osnove, jednostavnosti i mogućnosti provjere ispravnosti dijagrama. Kako bi se osiguralo upravljanje konkurentnim procesima u ovome su radu protokoli za upravljanje konkurentnim transakcijama prilagođeni prostornim podacima. Kod pesimističnoga se tijekom rada mogu koristiti protokoli zaključavanja u dvije faze ili altruistično zaključavanje, a kod optimističnoga tijekom rada BOCC i FOCC protokoli.

S obzirom da priprema promjene na katastarskim podacima može potrajati te se provjerava jedino konačni učinak procesa, faze pripreme promjene i provjeravanja ispravnosti procesa razdvajaju se u skladu s prijelaznim neto učinkom procesa (Widom i Finkelstein, 1990). Osnovni su koncepti, vrste procesa, opseg procesa i uvjeti integriteta preuzeti iz (Vranić i dr., 2015) te su učinjene manje izmjene i proširenja modela kako bi se integrirali u model razvijen u ovom radu.

Osnovne su komponente i moduli WFMS-a definirani referentnim modelom WFMS-a

(WFMC, 1995). U ovome je radu razvijen dijagram klasa za modeliranje procesa koji omogućava pohranjivanje definicije tijeka rada u skup povezanih objekata. Verzioniranjem se tih objekata omogućuje brisanje pojedinih dijelova ili čitavoga tijeka rada i nakon što određeni proces započne prema tome tijeku rada. Da bi se podržala mogućnost oporavka procesa, za svaki zadatak moguće je definirati kompenzirajući zadatak koji će poništiti promjene.

U odjeljku 4.1 navedeni su pojmovi i koncepti koji se koriste za izradu modela te se definira pojam konkurentnoga procesa nad katastarskim česticama predstavljenih poligonima. U odjeljku 4.2 objašnjen je općeniti konceptualni model WFMS-a koji omogućava modeliranje procesa nad katastarskim podacima. U odjeljku 4.3 općeniti je konceptualni model WFMS-a proširen kako bi se omogućila podrška za modeliranje procesa nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Model podataka WFMS-a je moguće povezati s katastarskim česticama, a u odjeljku 4.4 su objašnjeni različiti pristupi modeliranju procesa koji osiguravaju različite razine konkurentnosti procesa. Uvođenjem međnih točaka u model podataka moguće je dodatno poboljšati konkurentnost procesa. Stoga je u odjeljku 4.5 objašnjen način prepoznavanja promjena na poligonima na razini točaka koji omogućava altruistični pristup modeliranju procesa kojim je moguće razmjenjivati podatke između konkurentnih procesa. Na kraju (odjeljak 4.6) je analizirana mogućnost integracije WFMS-a razvijenog u ovom radu s LADM-om.

### 4.1. Pregled ulaznih postavki

U ovome je radu prostorna sastavnica katastarskih čestica predstavljena poligonima u skladu s normom International Standardization Organization (ISO) 19125. To znači da je granica između susjednih katastarskih čestica dvostruko pohranjena. Zbog toga se mogu pojaviti pogreške poput rupa između poligona ili preklapanja, ukoliko nisu zadana pravila očuvanja ispravnosti prostorne sastavnice. Sustavi temeljeni na konceptima klasičnoga srednjoeuropskog katastra kao glavni uvjet integriteta prostorne sastavnice obično koriste ispravnost ravninske particije.

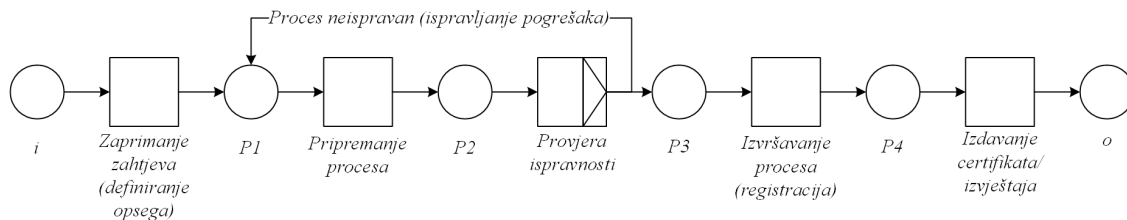
Procesi nad katastarskim podacima mogu trajati duže vrijeme te se često koristi verzioniranje katastarskih čestica kako procesi koji obavljaju promjene na podacima ne bi blokirali procese koji samo čitaju podatke (primjerice, izdavanje izvoda). Na taj način procesi pripremaju promjene na privatnim kopijama podataka, a ostali procesi čitaju posljednju važeću verziju. U ovome je radu primijenjen koncept prijelaznoga neto učinka procesa (Widom i Finkelstein, 1990) kojim se provjerava samo konačan učinak skupa operacija u procesu što znači da stanje privatne kopije može biti u privremenome nekonzistentnom stanju, ali konačni učinak tih operacija mora biti u skladu s uvjetima integriteta. Ispravnost je procesa osigurana uvjetima integriteta, a svojstvo atomičnosti osigurava da se proces provede u potpunosti ako je u skladu s uvjetima integriteta ili da se odbaci i ponište sve promjene pomoću kompenzirajućih zadataka. Izolacija se kod čitanja po-

dataka postiže verzioniranjem, jer procesi koji čitaju podatke nemaju pristup privatnim kopijama ostalih procesa, a za procese koji obavljaju promjene na podacima izolacija se postiže prostornom definicijom zahvaćenoga područja procesa i ograničavanjem istovremenoga pristupa česticama unutar njega. Jednom kada se proces provede, svojstvo je trajnosti prepušteno bazi podataka.

U (Zevenbergen i dr., 2007) prepoznate su opće faze procesa u katastru, a ovdje se navode faze koje se koriste kod modeliranja procesa nad prostornom sastavnicom katastarskih čestica (Slika 4.1):

- zaprimanje zahtjeva za promjenom nad podacima,
- priprema tehničkoga dijela promjene nad podacima,
- provjera ispravnosti promjene,
- zapisivanje promjena,
- izdavanje certifikata/izvještaja o promjeni.

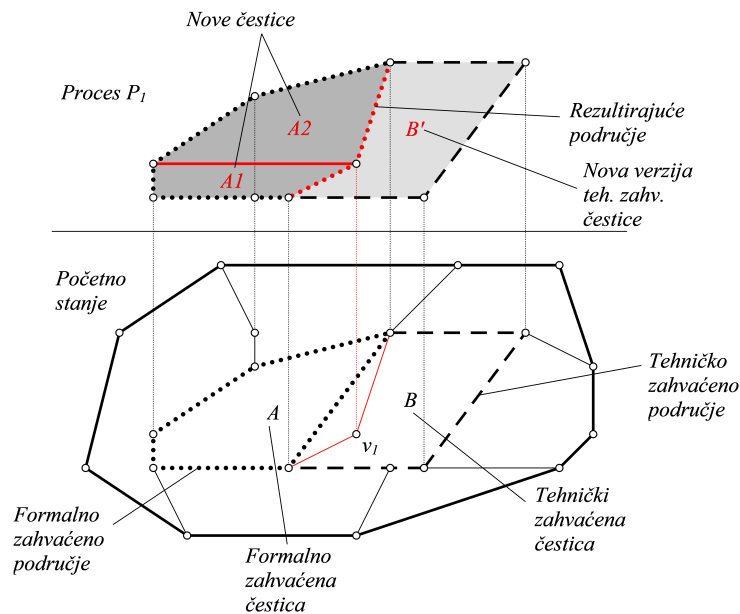
Procesi mogu uključivati i ostale faze koje mogu postaviti dodatna ograničenja na katastarske čestice vezane uz položaj, oblik ili veličinu čestice.



Slika 4.1: Osnovne aktivnosti procesa u katastru

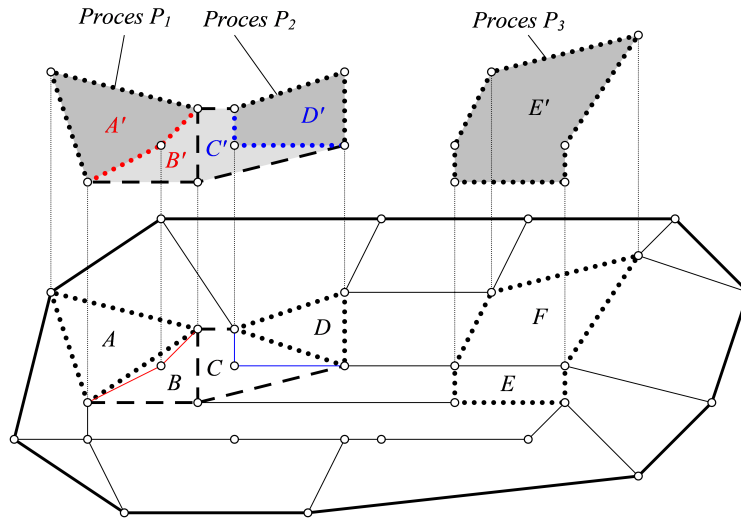
U ovome se radu kao osnova za ispitivanje ispravnosti procesa koriste uvjeti integriteta definirani u (Vranić i dr., 2015) jer ispitivanje ispravnosti na razini  $n$ -torke koji se koriste kod transakcija nije dovoljno za osiguravanje ispravnosti prostorne sastavnice katastarskih čestica. Autori su definirali i vrste procesa te uvjete integriteta prostorne sastavnice katastarskih čestica temeljene na poligonima. Zbog mogućih posrednih promjena na susjednim objektima, ukoliko su zadana pravila očuvanja topološke ispravnosti potrebno je prostorno definirati zahvaćeno područje procesa. Autori definiraju **formalno zahvaćeno područje** procesa nad prostornom sastavnicom katastarskih čestica pomoću geometrijske unije formalno zahvaćenih katastarskih čestica. **Formalno zahvaćene katastarske čestice** autori su definirali kao skup čestica koje sudjeluju u procesu i nad kojima je moguće obavljati promjene. Ako se rade promjene na granici formalno zahvaćenoga područja, radi se o varijabilnome formalno zahvaćenome području, odnosno granice se formalno zahvaćenoga područja mijenjaju u okviru procesa. U tome se slučaju uvodi pojam **tehnički zahvaćenih katastarskih čestica** koje se dodaju u proces s ciljem osiguravanja tehničke ispravnosti zbog utjecaja posrednih promjena na granici varijabilnoga formalno zahvaćenoga područja (Slika 2.16). Nad tehnički zahvaćenim katastarskim česticama moguće je

obavljati jedino promjenu korekcije geometrije na zajedničkim granicama sa zahvaćenim područjem, odnosno identitet se tehnički zahvaćenih čestica ne smije promijeniti niti uništiti (diobom ili spajanjem). Geometrijsku uniju tehnički zahvaćenih katastarskih čestica nazivaju **tehničko zahvaćeno područje**. Autori se nisu bavili konkurentnim procesima, te je za osiguravanje ispravnosti u slučaju konkurentnih procesa potrebno onemogućiti drugim procesima pristup formalnim i tehničkim katastarskim česticama. U ovome se radu definira **zahvaćeno područje** kao radno područje procesa i ono obuhvaća geometrijsku uniju formalno i tehnički zahvaćenoga područja (Slika 4.2). Ukoliko proces ne sadrži tehnički zahvaćene čestice, odnosno nema posrednih promjena na susjednim česticama, tada je formalno zahvaćeno područje jednako zahvaćenome području. Zahvaćeno područje ima još jednu svrhu, a to je da se definira koji je skup podataka neophodno dohvatiti za ispitivanje konzistentnosti prostorne sastavnice. Dakle, ne treba se provjeravati kompletni skup podataka već samo podskup koji je zahvaćen procesom što može povećati učinkovitost testova za kontroliranje ispravnosti prostorne sastavnice.



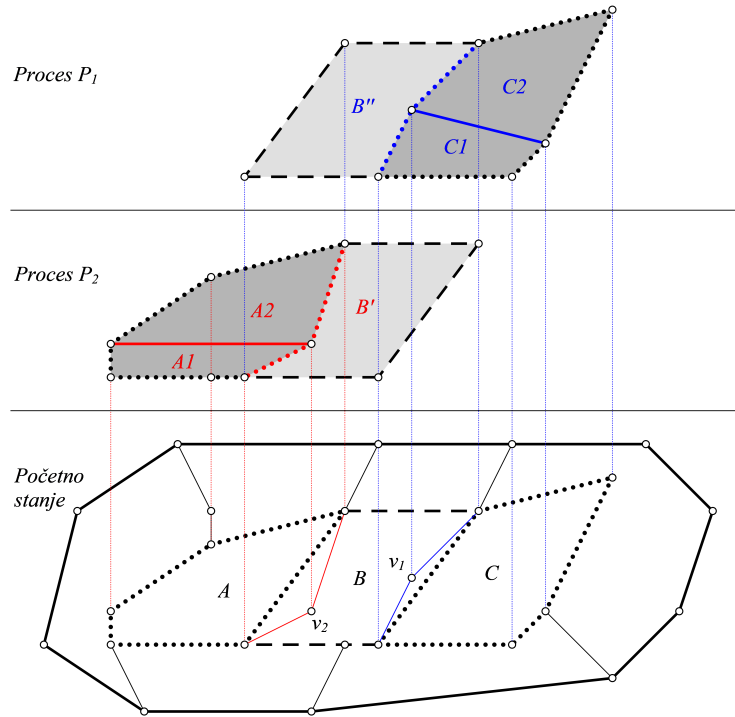
Slika 4.2: Definicija pojmova (temelji se na (Vranić i dr., 2015))

U nastavku se pojam definicije konkurentnoga procesa prilagođava procesima nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Koncept konkurentnoga procesa temelji se na konceptu konkurentne transakcije koji se koristi kod baza podataka. Konkurentnost se procesa određuje usporedbom prostornog odnosa zahvaćenih područja aktivnih procesa.



Slika 4.3: Procesi koji nisu konkurentni

Slika 4.3 prikazuje tri procesa koji nisu konkurentni. U procesu  $P_1$  mijenja se granica između čestica  $A$  (formalno zahvaćena čestica) i  $B$  (tehnički zahvaćena čestica). U procesu  $P_2$  mijenja se granica između čestica  $D$  (formalno zahvaćena čestica) i  $C$  (tehnički zahvaćena čestica). U procesu  $P_3$  spajaju se čestice  $E$  i  $F$  (obje su formalno zahvaćene čestice), a tehnički zahvaćenih čestica nema budući da se ne mijenja granica formalno zahvaćenoga područja. U primjerima na slici 4.3 zahvaćena se područja procesa ne preklapaju. Procesi  $P_1$ ,  $P_2$  i  $P_3$  pristupaju različitim skupovima podataka što se može utvrditi testiranjem topološkog odnosa zahvaćenih područja konkurentnih procesa.

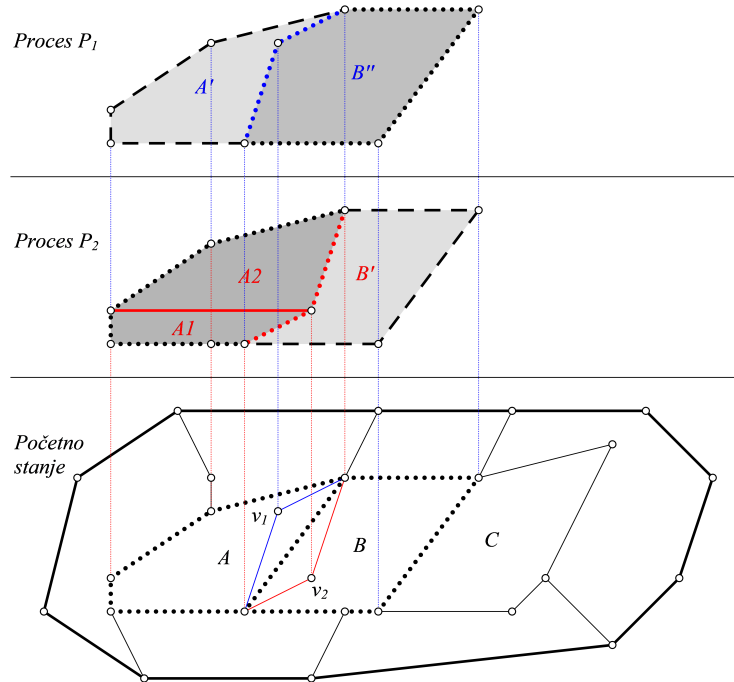


Slika 4.4: Konkurentni procesi

Slika 4.4 prikazuje dva procesa čija se formalno zahvaćena područja ne preklapaju, ali oba procesa rade izmjene na katastarskoj čestici  $B$  što znači da im se tehnički zahvaćena područja preklapaju. Proces  $P_2$  diobom čestice  $A$  dodaje čvor  $v_2$ , a proces  $P_1$  diobom čestice  $C$  dodaje čvor  $v_1$ . Zbog svojstva izolacije proces nema pristup promjenama koje ostali procesi istovremeno rade na istim podacima. Konkretno, proces  $P_1$  nije svjestan dodanoga čvora  $v_2$ , kao što proces  $P_2$  nije svjestan dodanoga čvora  $v_1$ . Proces mora poštivati svojstvo izolacije zbog ispravnoga temporalnog poretka verzija katastarskih čestica. Istovremenom će se provedbom procesa  $P_1$  i  $P_2$  pojaviti problem izgubljene izmjene i ostat će pohranjena promjena procesa koji se posljednji provede. Proces  $P_1$  i  $P_2$  dijele istu tehničku zahvaćenu česticu, odnosno pristupaju istim podacima. U nastavku je navedena definicija konkurentnoga procesa prilagođena katastarskim česticama temeljenima na poligonima (Definicija 4.1).

**Definicija 4.1 (Konkurentni proces na katastarskim česticama)** *Konkurentni proces je svaki aktivni proces čije se zahvaćeno područje preklapa sa zahvaćenim područjem aktivnoga procesa.*

Poseban je slučaj kada se formalno zahvaćena područja dva procesa dodiruju prikazan na slici 4.5. U takvome je slučaju čestica  $A$  formalno zahvaćena u procesu  $P_2$  i istovremeno tehnički zahvaćena u procesu  $P_1$ . Istovremenom provedbom procesa  $P_1$  i  $P_2$  nije moguće osigurati ispravnost procesa budući da čestica  $A$  nestaje u procesu  $P_2$ . Provedbom procesa  $P_1$  nastaje nova verzija čestice  $A$  što dovodi do situacije da na istome mjestu postoji više čestica istovremeno.



Slika 4.5: Konfliktni konkurentni procesi

Iz prostornog odnosa formalno zahvaćenih područja konkurentnih procesa slijedi definicija konfliktnog konkurentnog procesa nad katastarskim česticama temeljenima na poligonima (Definicija 4.2).

**Definicija 4.2 (Konfliktni konkurentni proces na katastarskim česticama)** *Konfliktni konkurentni proces je svaki aktivni proces kod kojeg se formalno zahvaćena područja dodiruju ili preklapaju.*

U definiciji 4.2 navedeno je da se radi o konfliktnome konkurentnom procesu i u slučaju kada se formalno zahvaćena područja preklapaju. Takav se slučaj može dogoditi ako je ista katastarska čestica dodana kao formalno zahvaćena u više procesa. Budući da svaki od konfliktnih konkurentnih procesa može mijenjati identitet formalno zahvaćene čestice, može se pojaviti problem duplih čestica na istome području kao i u slučaju kada se formalno zahvaćena područja dodiruju. Kod konkurentnih se procesa može pojaviti problem izgubljene vrijednosti. Zbog osiguravanja ispravnosti i oporavka procesi se u WFMS-u trebaju provoditi prema serijalizabilnom rasporedu.

## 4.2. Dijagram klasa modela procesa

U ovome je radu razvijen dijagram klasa u kojemu su elementi WF mreža organizirani u skup klasa koristeći paradigmu objektno orijentiranoga modeliranja, a dijagram klasa je podijeljen na dva modula u skladu s referentnim modelom WFMS-a. Razvijeni model

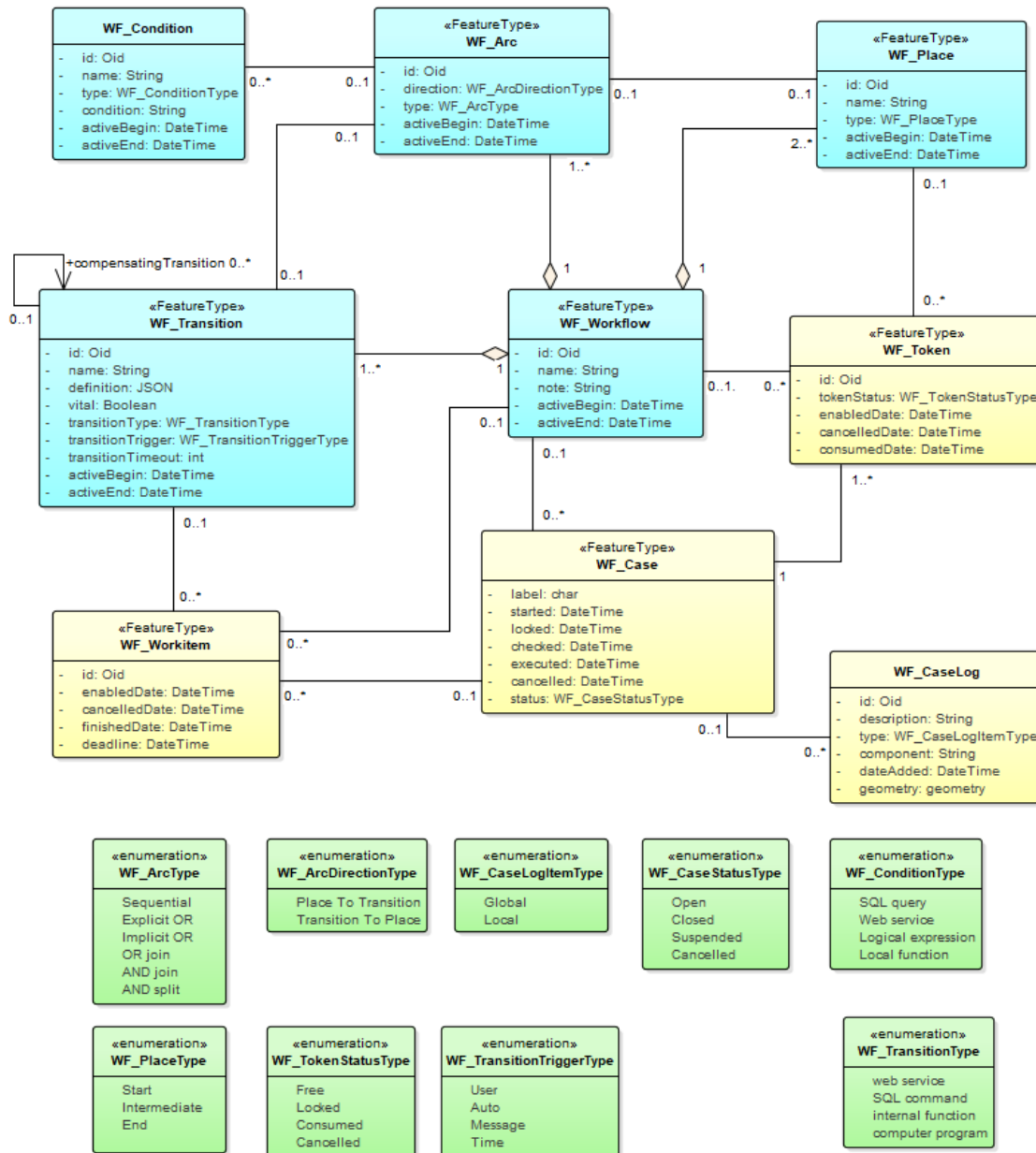


predstavlja općeniti konceptualni model koji uz standardne WFMS zahtjeve treba omogućiti sljedeće:

- opisivanje procesa WF mrežama,
- podrška za transakcijska svojstva atomičnosti i izolacije,
- definiranje uvjeta integriteta na razini prijelaza, odnosno zadatka,
- ograničavanje pristupa podacima na razini objekta,
- oporavak procesa pomoću kompenzirajućih zadataka (oporavak-natrag) stvaranjem reflektivne veze sa zadatka na kompenzirajuće zadatke,
- pohranjivanje geometrijske komponente o pogreškama nastalih tijekom provjere ispravnosti prostorne sastavnice.

Definicija je procesa predstavljena skupom međusobno povezanih klasa, a metode klase služe za interpretaciju i upravljanje procesima. U ovome su radu detaljno razrađeni moduli za definiciju i interpretaciju procesa (Slika 4.6) dok su ostali moduli razrađeni u minimalnom opsegu koji je potreban za modeliranje procesa za provjeru ispravnosti prostorne sastavnice katastarskih čestica i njihov je detaljan opis izvan opsega ovog rada. Plavom su bojom prikazane klase modula za definiciju procesa, a žutom bojom klase modula za interpretaciju procesa.

Tijek se rada (klasa *WF\_Workflow*) sastoji od niza mjesta (klasa *WF\_Place*) i prijelaza (klasa *WF\_Transition*) međusobno povezanih lukovima (klasa *WF\_Arc*). Svakome se luku može dodijeliti proizvoljan broj uvjeta (klasa *WF\_Condition*) što omogućava modeliranje preduvjeta i postuvjeta pojedinih prijelaza. Svaka klasa modula za definiciju procesa sadrži temporalne atribute *activeBegin* i *activeEnd* kako bi se omogućilo arhiviranje definicije procesa i nakon što nastanu slučajevi prema određenoj vrsti procesa, odnosno kako bi se tijekom rada mogao mijenjati ili brisati nakon što se pokrenu slučajevi. Ukoliko se slučaj pokrene prema određenome tijeku rada, taj tijek rada više nije moguće obrisati zbog osiguranja integriteta podataka, ali se može arhivirati korištenjem navedenih temporalnih atributa. Na ovaj je način ostvarena podrška za pohranjivanje novih verzija poslovnih procesa koji su rezultat kontinuiranoga poboljšavanja poslovnih procesa (BPR).



Slika 4.6: UML dijagram klasa modula za definiciju i interpretaciju procesa

Klasa *WF\_Place*, osim identifikatora i naziva, sadrži atribut tipa (*WF\_PlaceType*) koji označava vrstu mjesta: početno, završno ili međumjesto. Kao što je ranije objašnjeno, svaki tijek rada mora imati barem dva mjesta (početno i završno), ali može ih imati i više. Klasa *WF\_Transition* sadrži attribute identifikator, naziv te okidač (*WF\_TransitionTriggerType*) i vrijeme isteka (*transitionTimeout*). Okidač definira način pokretanja radnje koja je definirana u prijelazu. Prijelaz predstavlja određeni zadatak koji se provodi, a vrsta prijelaza (atribut *transitionType*) može biti web servis, SQL naredba, funkcija ili računalni program. Još je jedan bitan atribut prijelaza atribut *vital* koji označava je li prijelaz bitan (eng. *vital*)

ili nebitan (eng. *non-vital*). Zadaci koji se temelje na bitnim prijelazima bitni su zadaci i njihovo neuspješno izvođenje uzrokuje odbacivanje promjena u zadatku i povratak na posljednje konzistentno stanje. Nebitni se zadaci mogu provesti neuspješno, ali to neće utjecati na konačni ishod procesa. Tijek rada može imati jedan ili više prijelaza. Prijelaz može imati refleksivnu vezu na jedan ili više kompenzirajućih prijelaza. Prijelazi i mjesta spojeni su lukovima. Klasa *WF\_Arc* ima attribute identifikator te smjer (*WF\_ArcDirectionType*) i tip (*WF\_ArcType*). Smjer luka može biti od mjesta do prijelaza ili od prijelaza do mjesta. Ostale kombinacije nisu dopuštene (primjerice, mjesto-mjesto ili prijelaz-prijelaz). Tip luka omogućava korištenje elemenata za različite vrste tijekova rada (primjerice, sekvencijalni, paralelni ili izbor). Klasa *WF\_Condition* sadrži definiciju uvjeta koji mogu biti pridruženi određenome mjestu kao preduvjet ili postuvjet prijelaza. U modelu je uvjet vezan uz klasu *WF\_Arc* da bi se lakše modelirali preduvjeti, odnosno postuvjeti određenoga mjesta. Ako je uvjet vezan uz luk koji ide od prijelaza do mjesta, tada je riječ o postuvjetu prijelaza, a u suprotnome se radi o preduvjetu. Uvjet sadrži attribute: identifikator, naziv, tip (*WF\_ConditionType*) i definiciju uvjeta (*condition*). Tip uvjeta definira način na koji se uvjet pokreće. Moguće su vrijednosti SQL upit, web servis, interna funkcija ili logički izraz.

Klasa *WF\_Case* predstavlja poslovni proces koji je potrebno riješiti. Vezana je uz klasu *WF\_Workflow* na način da za jedan tijek rada može biti vezano više slučajeva, odnosno više se slučajeva može instancirati iz iste definicije procesa. Klasa *WF\_Workitem* predstavlja pojedinu radnju (radnu stavku) unutar procesa koja može biti dodijeljena korisniku. Kad postane omogućena, odnosno kada se provodi od strane resursa, naziva se još i aktivnost. Definicija je aktivnosti sadržana u klasi *WF\_Transition*. Prijelaz definira strukturu i način funkcioniranja pojedine aktivnosti. Zato jedan prijelaz može biti vezan na više aktivnosti. Osim identifikatora, aktivnost sadrži temporalne attribute koji određuju status aktivnosti. Vrijednost atributa *enabledDate* zapisuje se nakon što su ostvareni preduvjeti prijelaza ispunjeni. Atribut *finishedDate* se označava trenutak kada se ispune postuvjeti prijelaza. Vrijednost atributa *cancelledDate* zapisuje se ako korisnik odustane od aktivnosti ili ako se pokrenu kompenzirajuće aktivnosti. Atribut *deadline* ukoliko je popunjen, služi kao informacija korisniku dokad aktivnost treba biti obavljena. Klasa *WF-Token* drži informaciju na kojemu se mjestu nalazi određeni proces. Uz identifikator i status, sadrži temporalne attribute koji ukazuju na status oznake. Vrijednost atributa *enabledDate* zapisuje se u trenutku kada su postuvjeti prethodnoga prijelaza ispunjeni. Vrijednost atributa *consumedDate* zapisuje se u trenutku kada se sljedeća aktivnost pokreće, odnosno kada su ispunjeni preduvjeti vezanoga prijelaza. Jedan slučaj može imati jednu ili više oznaka. Klasa *WF\_CaseLog* pomoćna je klasa koja služi za pohranjivanje dnevnika o provedenim aktivnostima unutar procesa i za lakše otkrivanje pogrešaka. Atribut tip (*WF\_CaseLogItemType*) određuje je li stavka lokalna (poruka na razini određene funkcije) ili globalna (na razini procesa). Ukoliko je riječ o topološkoj ili geometrijskoj pogrešci, taj se podatak zapisuje kao vrijednost atributa *geometry*.

Ostali su moduli referentnoga modela WFMS-a implementirani u minimalnom obliku kako bi se mogli provjeriti slučajevi korištenja. Upravljanje korisnicima i njihovim pra-

vima obavezna je funkcionalnost svakoga WFMS-a. U ovome se radu implementirani model upravljanja korisnicima temelji na Role-based Access Control (RBAC) konceptu kao što je objašnjeno u (Ferraiolo i dr., 1995; Hansen i Oleshchuk, 2003; Ahn i dr., 2000). Prednost je korištenja RBAC koncepta da korisnici nemaju prava pristupa objektima, već se prava pristupa objektima povezuju s ulogama, a korisnicima se pridružuju određene uloge. Uloga predstavlja skup prava, a svakome se korisniku može dodijeliti jedna ili više uloga. Međutim, sustavna analiza prikladnoga modela za upravljanje korisnicima i njihovim pravima je izvan opsega ovog rada i ne predlaže se navedeni RBAC koncept kao jedino moguće i najbolje rješenje.

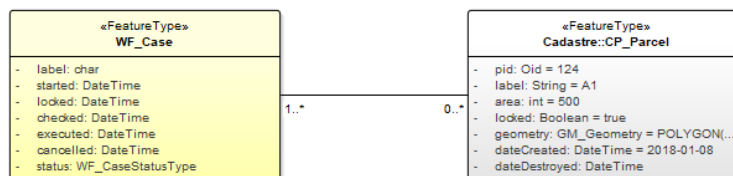
WFMS ne treba biti upoznat s internom strukturom pojedine aktivnosti ili učinkom cjelokupnoga procesa na podatke. Na taj se način postiže općenitost WFMS-a koji može biti korišten za modeliranje poslovnih procesa u raznim područjima. Definicijom su procesa određene vrste aktivnosti te njihov način i redoslijed provedbe. Prava izvođenja određene aktivnosti određuju se integracijom sustava za upravljanje korisnicima.

### 4.3. Povezivanje modela WFMS-a s katastarskim česticama

U ovome je odjeljku objašnjeno povezivanje modela WFMS-a (vidi odjeljak 4.2) s katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Općeniti je model definiran u prethodnom odjeljku proširen kako bi podržao sljedeće zahtjeve:

- prostorna definicija zahvaćenoga područja procesa,
- osiguravanje serijalizabilnosti procesa ograničavanjem pristupa podacima unutar zahvaćenoga područja i
- osiguravanje topološke i geometrijske ispravnosti prostorne sastavnice katastarskih čestica.

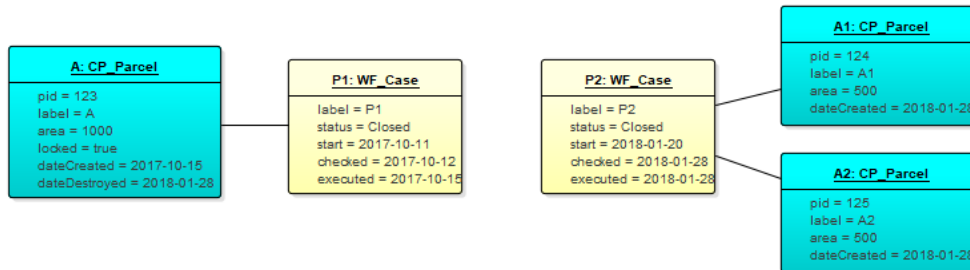
U nastavku ispitujemo različite pristupe u povezivanju katastarskih čestica s procesima da bi se utvrdilo ispunjava li pojedini model postavljene zahtjeve. Za početak se ispituje povezivanje katastarskih čestica kojima su dodijeljeni temporalni atributi s općenitim modelom WFMS-a pomoću jednostruke veze na klasu *WF\_Case* (Slika 4.7).



Slika 4.7: Povezivanje općenitoga modela WFMS-a s katastarskim česticama

Na slici 4.7 je katastarska čestica (klasa *CP\_Parcel*) prikazana sa skupom osnovnih atributa i povezana je sa slučajem (klasa *WF\_Case*). Atribut *locked* omogućava zaključavanje

na razini objekta. Atributi *dateCreated* (datum nastanka objekta) i *dateDestroyed* (datum nestanka objekta) služe za modeliranje temporalne sastavnice katastarskih čestica dodjeljivanjem temporalnih atributa objektima čime se povećava konkurentnost procesa.

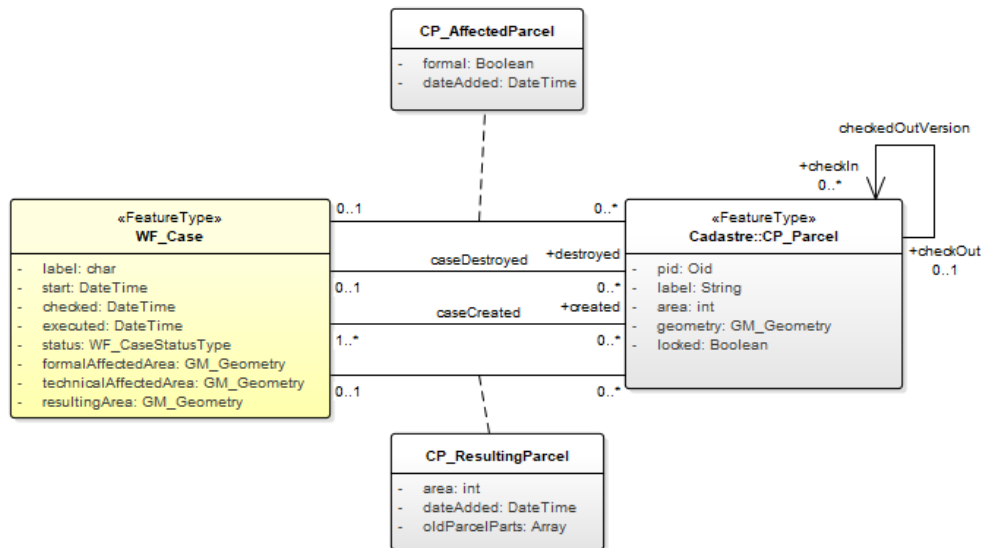


Slika 4.8: Objektni dijagram čestica povezanih pomoću jednostavnog modela

Na slici 4.8 je prikazan primjer čestice *A* nastale u procesu *P1* koja se u procesu *P2* dijeli na dvije nove čestice *A1* i *A2*. Budući da nema eksplicitne veze između procesa *P2* i ulazne čestice *A*, nije definirano radno područje procesa, već bi korisnik trebao samostalno onemogućiti pristup konkurentnim procesima promjenom vrijednosti atributa *locked=true* na katastarskoj čestici *A*. Prilikom provedbe procesa *P2* svim novim katastarskim česticama koje su nastale u tome procesu, odnosno objektima koje ih predstavljaju trebao bi se dodijeliti datum nastanka, a ulaznoj je čestici *A* potrebno dodijeliti vrijednost atributa *dateDestroyed* kako bi se arhivirala. Ukoliko se ne mijenjaju granice formalno zahvaćenoga područja, osigurana je ispravnost prostorne sastavnice. Međutim, ako se mijenjaju granice formalno zahvaćenoga područja, tada nije moguće osigurati ispravnost prostorne sastavnice zbog posrednih promjena na susjednim česticama jer se zaključavanje obavlja na razini objekta. Zbog ovih nedostataka ovakav model primjene WFMS-a nije prikladan za modeliranje procesa nad katastarskim česticama čija je prostorna sastavnica predstavljena poligonima.

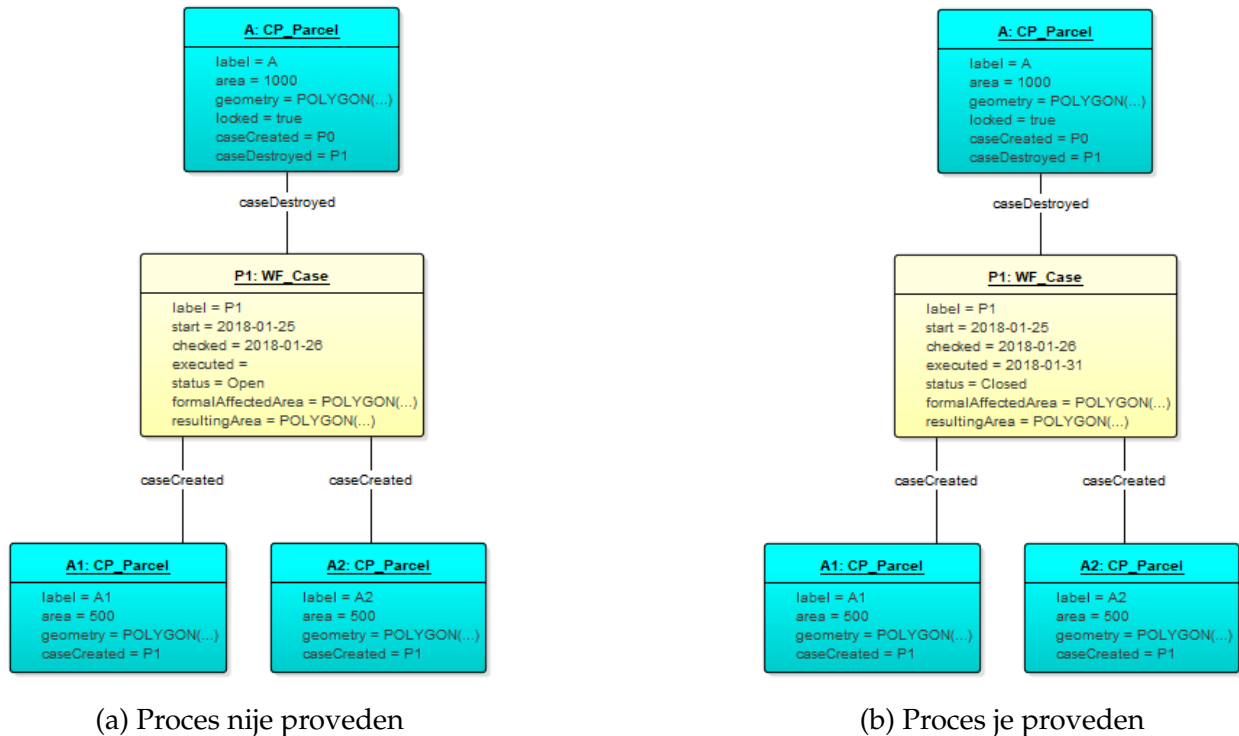
Zbog toga u model uvodimo zahvaćeno područje kao što je objašnjeno u prethodnom odjeljku 4.1. Vranić i dr. (2015) definiraju zahvaćeno i rezultirajuće područje kao attribute transakcije (klasa *Transaction*) koja je vezana s katastarskim česticama dvjema vezama. Prva veza (*created*) postoji između svake katastarske čestice i transakcije. Druga je veza (*destroyed*) neobavezna i postoji između katastarske čestice i transakcije u kojoj je ona prestala postojati. Osim te dvije veze koje određuju postojanje čestica, autori uvode još dvije vezne klase u koje su pohranjeni podaci o zahvaćenim (klasa *AffectedParcel*) i rezultirajućim (klasa *ResultingParcel*) katastarskim česticama. Uvjeti integriteta pohranjeni u zasebnoj klasi koja je povezana s vrstom transakcije. U ovome se radu taj model mijenja kako bi bio u skladu s razvijenim konceptualnim modelom WFMS-a. Katastarska je čestica povezana sa slučajem (klasa *WF\_Case*) (Slika 4.9), a uvjeti integriteta mogu biti implementirani kao dio jednog ili više zadataka. Vezna klasa zahvaćena čestica (*CP\_AffectedParcel*) proširena je atributom *formal* koji omogućava razlikovanje formalnih i tehnički zahvaćenih čestica te datumom dodavanja zahvaćene čestice u proces (atribut *dateAdded*). Vezna

klasa rezultirajuća čestica (klasa *CP\_ResultingParcel*) proširena je atributima površine, datuma dodavanja te atributom *oldParcelParts* koji sadrži skup pokazivača na čestice iz kojih je rezultirajuća čestica nastala kako se ista ne bi trebala dohvaćati prostornim upitom. Katastarska je čestica proširena atributom *locked* koji omogućava zaključavanje čestice na razini objekta. Dodana je još jedna reflektivna veza (*checkedOutVersion*) koja omogućava pohranjivanje pokazivača na katastarsku česticu koja se uređuje. Moguće je da više procesa uređuje istu česticu istovremeno.



Slika 4.9: Prošireni konceptualni model WFMS-a

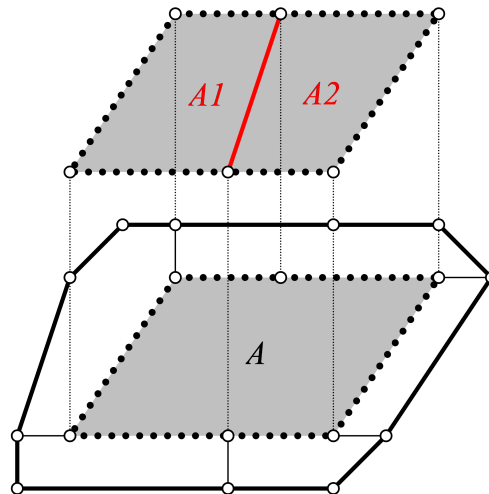
Prednost je ovoga pristupa da provedbom procesa, odnosno upisivanjem temporalnog atributa u vrijednost atributa *executed* određenoga procesa sve čestice koje imaju vezu *created* postaju važeće, a sve koje imaju vezu *destroyed* postaju arhivirane (Slika 4.10).



Slika 4.10: Promjena statusa verzija katastarskih čestica

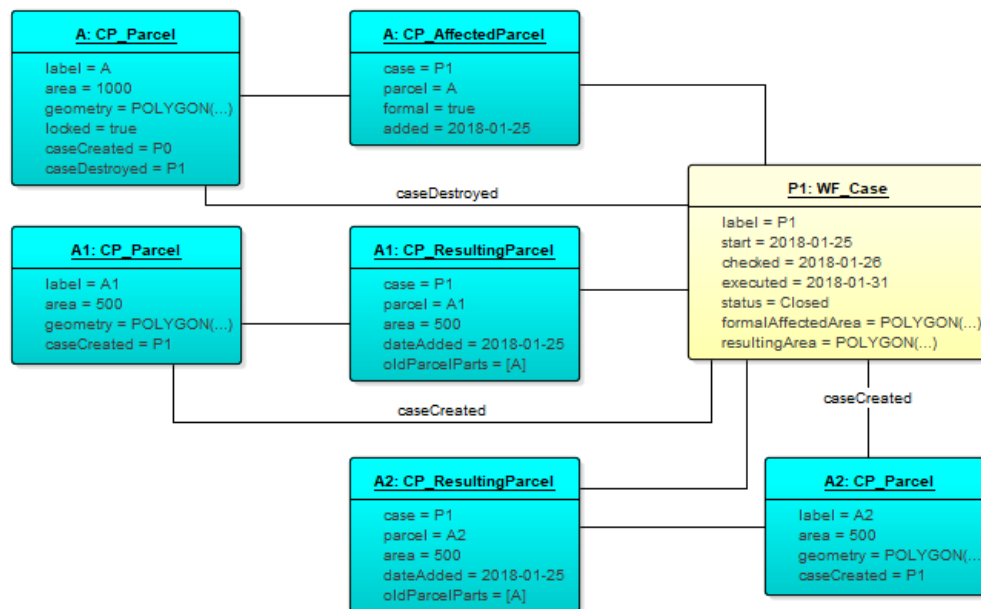
Slika 4.10a prikazuje dvije čestice A1 i A2 koje predstavljaju privatne verzije procesa, a čestica A još je uvijek važeća zato što proces P1 još nije proveden. Na slici 4.10b proces P1 je proveden (atribut *executed* ima vrijednost) te su čestice A1 i A2 postale važeće, a čestica A je arhivirana. Na taj način nije potrebno ažurirati svaki objekt uključen u proces već je ažuriranjem atributa procesa promijenjen status svih objekata vezanih na taj proces.

U nastavku je prikazano na koji se način modeliraju različiti slučajevi pomoću proširenoga konceptualnog modela. Na slici 4.11 prikazan je uobičajeni primjer diobe katastarskih čestica. Katastarska je čestica A formalno dodana u proces. Budući da dioba ne uzrokuje promjene na granici formalno zahvaćenoga područja, znači da nije potrebno dodavati tehnički zahvaćene katastarske čestice u proces.



Slika 4.11: Dioba katastarske čestice

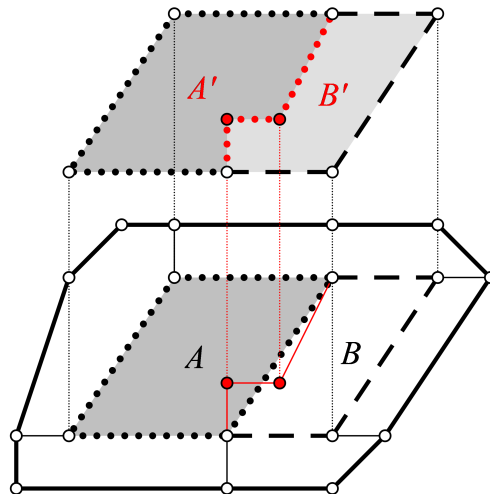
Formalno zahvaćena katastarska čestica  $A$  nastala je u prethodno provedenome procesu  $P0$  te je vezana s procesom  $P1$  vezom *caseDestroyed*. Isto je tako s procesom  $P1$  povezana preko vezne klase *CP\_AffectedParcel* (Slika 4.12). U procesu  $P1$  postoje dvije rezultirajuće čestice  $A1$  i  $A2$  koje su vezane na proces  $P1$  vezom *caseCreated*. S obzirom da se čestica  $A$  poništava u procesu  $P1$  i nastaju dvije nove čestice koje nemaju atribut *checkedOutVersion* jer niti jedna nije preuzela identitet od  $A$ . Atribut *executed* objekta *WF\_Case* označava vrijeme kada je proces proveden. Provedbom procesa (ažuriranjem jednog atributa objekta *WF\_Case*) svi objekti koji su vezani vezom *caseCreated* postaju važeći, a svi se objekti koji su vezani vezom *caseDestroyed* arhiviraju.



Slika 4.12: Dioba katastarske čestice (objektni dijagram)

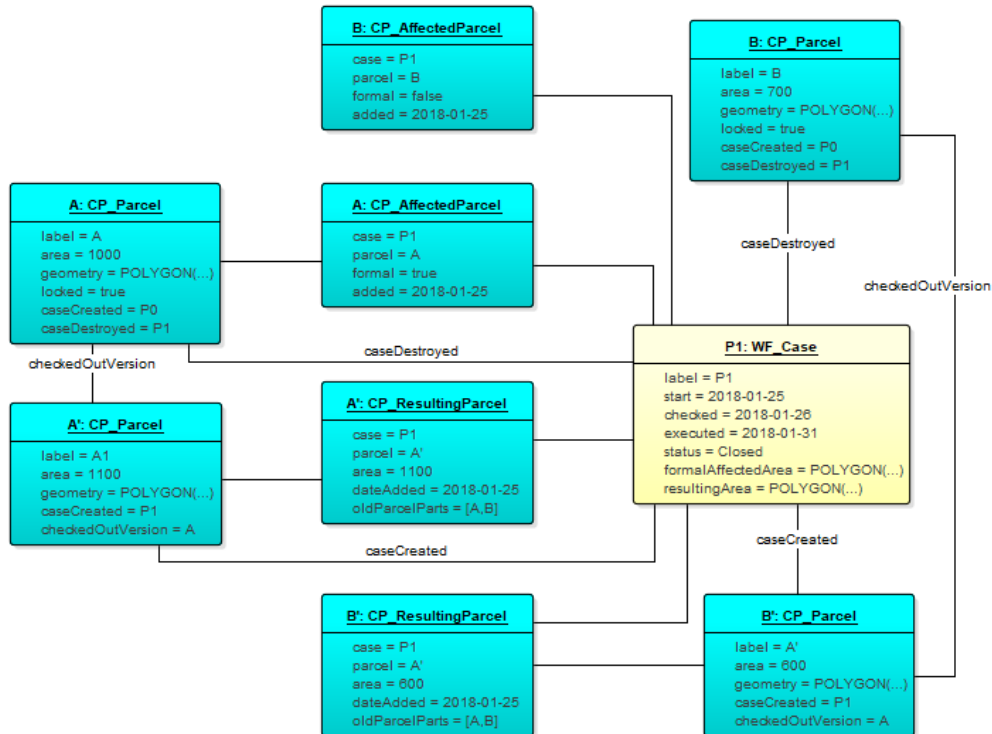


Na slici 4.13 prikazan je primjer korekcije granice između dvije katastarske čestice. Formalno zahvaćena katastarska čestica je čestica *A* koja ujedno predstavlja formalno zahvaćeno područje. Budući da se na granici formalno zahvaćenoga područja dodaju dvije točke, potrebno je tehnički dodati u proces česticu *B* kako bi se osigurala ispravnost prostorne sastavnice.



Slika 4.13: Korekcija granice katastarske čestice

U procesu *P1* čestica *A* zadržava identitet budući da se mijenja samo njezina granica. Da bi se zadržala ispravnost prostorne sastavnice katastarskih čestica, potrebno je promijeniti granicu čestice *B* (Slika 4.14). Korekcijom se granice katastarske čestice stvaraju nove verzije čestica *A'* i *B'*. Objekt *A'* ima refleksivnu vezu na objekt *A* iz kojega je nastao (*checkedOutVersion=A*). Isto tako, objekt *B'* ima refleksivnu vezu *checkedOutVersion* na objekt *B* iz kojega je nastao. Obje rezultirajuće čestice (*CP\_ResultingParcel*) imaju vrijednost atributa *oldParcelParts=[A,B]*, jer su obje nastale iz dijelova zahvaćenih čestica *A* i *B*. Na taj se način može otkriti radi li se o promjeni s fiksnim ili promijenjivim zahvaćenim područjem, odnosno je li promjena jedne čestice utjecala na druge čestice. Provedbom procesa *P1* rezultirajuće čestice *A'* i *B'* postaju važeće, a zahvaćene (formalno i tehnički) se čestice *A* i *B* arhiviraju.



Slika 4.14: Korekcija granice katastarske čestice (objektni dijagram)

U prethodnim je primjerima prikazano kako se modeliraju veze između katastarskih čestica i procesa, uključivanje uvjeta integriteta u model podataka te način modeliranja temporalne sastavnice katastarskih čestica. U nastavku se predstavljaju pristupi za modeliranje procesa koji u obzir uzimaju mogućnost da postoje konkurentni procesi koji istovremeno pokušavaju promijeniti iste podatke što dosad nije bilo razmatrano.

## 4.4. Modeliranje procesa

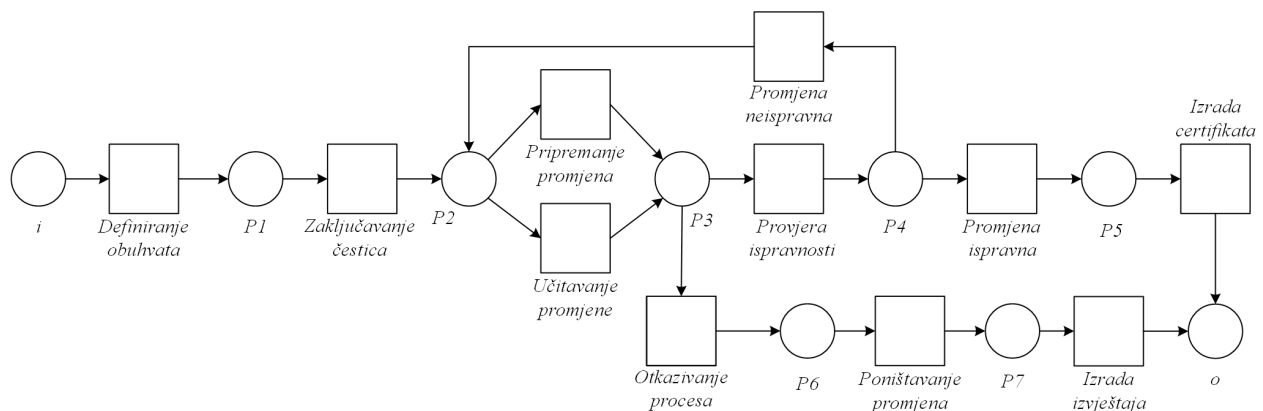
Ispravnost se procesa postiže primjenom uvjeta integriteta na podatke. Kako bi se osigurala ispravnost procesa i u slučaju konkurentnoga pristupa podacima, definira se zahvaćeno područje procesa unutar kojega je pristup katastarskim česticama ograničen. U ovom je odjeljku objašnjeno upravljanje konkurentnim procesima nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima pomoću dva tijeka rada koji osiguravaju ispravnost prostorne sastavnice pomoću elemenata tijeka rada u skladu s modelom integracije transakcijskih koncepata *WF*. Ideja pesimističnog i optimističnoga tijeka rada se temelji na analognim pojmovima pesimističnih i optimističnih protokola za upravljanje konkurentnim transakcijama u bazama podataka, ali su prilagođeni prostornoj sastavnici katastarskih čestica temeljenoj na poligonima.

Pesimistični tijek rada osigurava izolaciju procesa tako da se zaključavaju sve katastarske čestice koje su formalno i tehnički zahvaćene procesom. Kod optimističnoga se tijeka

rada katastarske čestice ne zaključavaju, već se prije same provedbe procesa provjerava postoje li konkurentni procesi prostornim presjekom zahvaćenih područja aktivnih procesa. Ako postoje konkurentni procesi, proces nastavlja dalje prema tijeku rada - odbija se ili vraća proces na ponovno pripremanje.

#### 4.4.1. Pesimistični tijek rada

Pesimistični tijek rada (Slika 4.15) osigurava izolaciju procesa tako da se zaključaju sve formalno i tehnički zahvaćene katastarske čestice. Nakon definiranja zahvaćenoga područja zaključavaju se sve formalno i tehnički zahvaćene katastarske čestice. Po uspješnome zaključavanju (mjesto *P2*) atribut *locked* objekta *WF\_Case* dobiva vrijednost i drugi procesi zaključane čestice ne mogu zaključati niti mijenjati ako atribut *locked* ima dodijeljenu vrijednost. Sljedeći je prijelaz priprema promjene na česticama ili učitavanje pripremljene promjena. Prijelaz *Provjera ispravnosti* provjerava ispravnost prostorne sastavnice katastarskih čestica unutar procesa prema uvjetima ispravnosti. Ukoliko su svi uvjeti integriteta zadovoljeni, promjena se označava kao ispravna (atribut *checked* dobiva vrijednost) i proces nastavlja prema provedbi. U suprotnome se mogu ispraviti pogreške i ponovno provjeriti ispravnost. Ukoliko se pogreške ne mogu ispraviti, proces se može otkazati, a promjene poništiti pozivanjem kompenzirajućih zadataka. Uspješnom provedbom procesa promjene postaju vidljive drugim procesima (ažuriranje atributa *executed* objekta *WF\_Case*), a katastarske se čestice otključavaju. Prije završetka procesa može se izraditi izvještaj (certifikat) o provedenome procesu. Ovdje se posljednji zadatak može definirati kao nebitan, što znači da neuspješno izdavanje izvještaja ne uzrokuje otkazivanje čitavoga procesa.

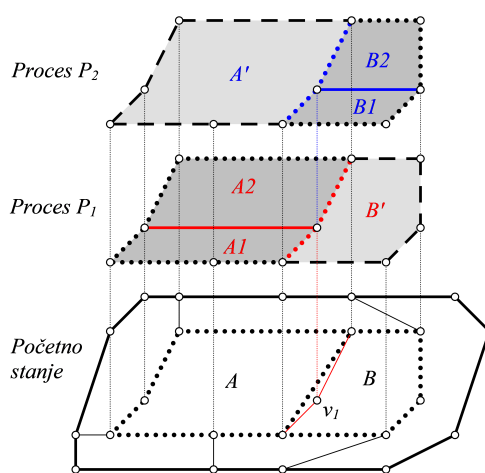


Slika 4.15: Pesimistični tijek rada

Zaključavanjem formalno i tehnički zahvaćenih katastarskih čestica osigurava se da drugi procesi ne mogu pristupiti katastarskim česticama na kojima se rade promjene, čime je osigurana i serijalizabilnost i oporavak procesa. Promjene se rade na privatnim

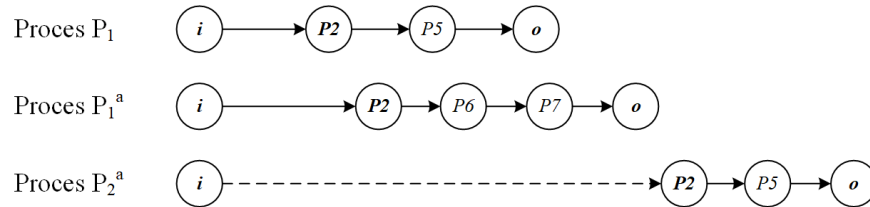
kopijama podataka tako da se operacije čitanja mogu izvršavati, odnosno pripremanje promjene (zapisivanje) ne utječe na operacije čitanja.

Na slici 4.16 prikazane su dvije komplementarne promjene u dva konfliktna konkurentna procesa. U procesu  $P_1$  katastarska se čestica  $A$  dijeli na dvije nove  $A1$  i  $A2$ . Zbog toga se stvara novi čvor na zajedničkoj granici s katastarskom česticom  $B$ . Da bi se očuvala topološka ispravnost prostorne sastavnice katastarskih čestica, dodaje se čvor  $v_1$  na granicu katastarske čestice  $B$  te se stvara nova verzija čestice  $B'$ . U drugome se procesu  $P_2$  čestica  $B$  dijeli na dvije nove  $B1$  i  $B2$  te se također stvara čvor ( $v_1$ ) na istome mjestu kao i u procesu  $P_1$ . Pozicija čvora  $v_1$  slučajno je ista u oba procesa (isti bi problem bio i da se čvor nalazi na različitoj poziciji na zajedničkom bridu). Iako se čini da bi istovremenom provedbom konkurentnih procesa topološka ispravnost prostorne sastavnice bila osigurana, to nije tako, jer se u procesu  $P_1$  stvaraju dvije nove čestice  $A1$  i  $A2$ , dok se u procesu  $P_2$  na istome mjestu stvara nova verzija čestice  $A'$ . Ista je situacija i s česticom  $B$ . Zbog toga je potrebno ograničiti pristup česticama kako bi se izbjegao problem duplih čestica na istome prostoru.



Slika 4.16: Konkurentni procesi (dioba čestica)

Ideja se pesimističnoga tijeka rada temelji na protokolu zaključavanja u dvije faze. Prva je faza zaključavanje objekata nad kojima će se raditi promjene, odnosno zaključavanje objekata koji su navedeni u prethodnome zadatku *Definiranje obuhvata*. Trenutak u kojem oznaka dođe u mjesto  $P2$  označava završetak zaključavanja objekata i od toga trenutka drugi procesi ne mogu zaključavati bilo koji od zaključanih objekata niti raditi izmjene nad njima. Objekti se otključavaju uspješnom provedbom procesa ili otkazivanjem procesa i poništavanjem promjena. Kada oznaka dođe u mjesto  $P5$  (u slučaju uspješne provedbe procesa) ili mjesto  $P7$  (u slučaju neuspješne provedbe procesa) taj trenutak označava završenu fazu otključavanja. Od toga trenutka drugi procesi mogu te objekte zaključavati i raditi izmjene na njima.



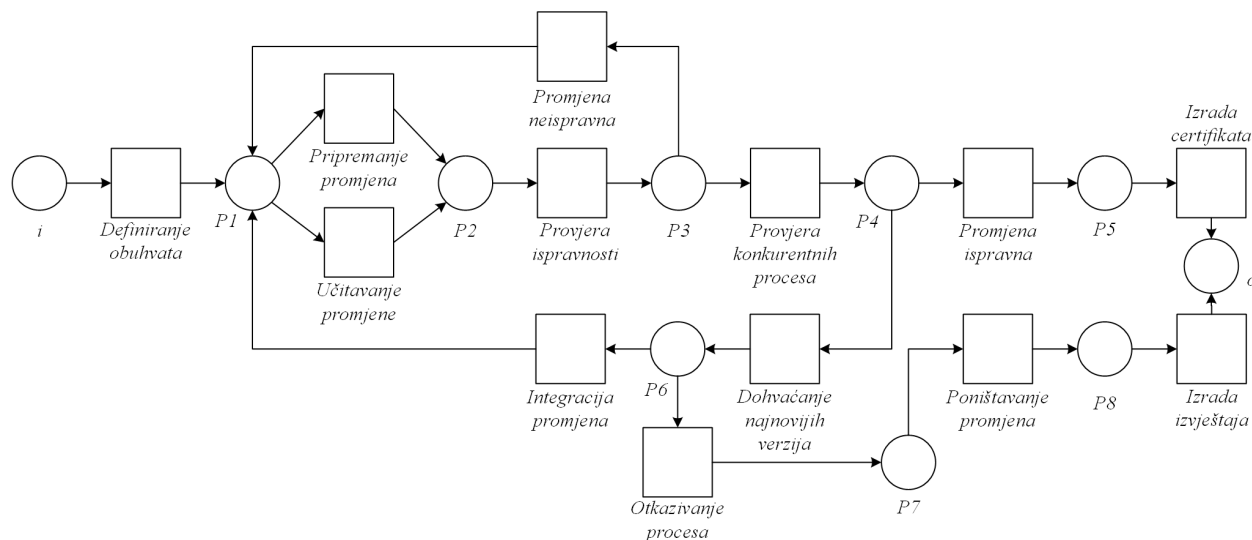
Slika 4.17: Provedba procesa prema pesimističnome tijeku rada

Na slici 4.17 prikazan je vremenski raspored provedbe procesa, odnosno pojedinih aktivnosti konkurentnih procesa. Proces  $P_1$  prvi zaključava katastarske čestice  $A$  i  $B$  te ima pravo prvi napraviti promjene na česticama. Proces  $P_2$  ima dvije mogućnosti: nakon neuspješnoga zaključavanja pokrenuti otkazivanje procesa i poništiti eventualne promjene pozivanjem kompenzirajućih zadataka ( $P_1^a$ ); ili pričekati sa zaključavanjem čestica dok se proces  $P_1$  ne provede i tek onda zaključati potrebne čestice i nastaviti proces ( $P_1^b$ ). Kod kompleksnih se promjena može dogoditi da jedan proces čeka duže vrijeme na pripremanje promjena. Optimistični tijek rada može skratiti trajanje procesa budući da omogućava istovremeno pripremanje promjena u konkurentnim procesima.

#### 4.4.2. Optimistični tijek rada

Kod optimističnoga tijeka rada (Slika 4.18) objekti se ne zaključavaju što omogućava da se promjena na podacima pripremi, a tek se prije provedbe procesa provjerava postoje li konkurentni procesi te može li proces nastaviti s provedbom i zapisivanjem promjena. U poglavlju 2 objašnjena su dva optimistična protokola: BOCC koji provjerava konkurentnost transakcija s već provedenim transakcijama i FOCC koji provjerava konkurentnost transakcija s transakcijama koje su u fazi čitanja. Ispitivanje konkurentnosti procesa nad katastarskim česticama prema BOCC protokolu nije prihvatljivo jer će proces biti označen kao ispravan jedino ako je konkurentni proces već završio. Time konkurentnost procesa nije povećana u odnosu na pesimistični tijek rada. Zato je u nastavku objašnjen modificirani FOCC protokol koji zahtijeva da presjek skupa podataka nad kojim jedan proces zapisuje i skupa nad kojim drugi proces čita podatke bude prazan skup u trenutku  $n$  ( $WS(t_j) \cap RS^n(t_i) = \emptyset$ ). Faza čitanja započinje u mjestu  $P_1$ , jer iako samom definicijom obuhvata procesa nije učinjena izmjena na podacima, ipak su definirane čestice na kojima će se raditi promjene. Konkurentni proces koji prvi provjeri ispravnost (dode u mjesto  $P_3$ ) bit će ispravno validiran, a ostali se procesi otkazuju ili vraćaju na ponovno pripremanje. Provjera se konkurentnih procesa obavlja usporedbom prostornog odnosa zahvaćenih područja procesa prema definicijama 4.1 i 4.2. Ako se zahvaćena područja procesa ne preklapaju ili ako se samo dodiruju, tada se ne radi o konkurentnim procesima. Granice se katastarskih čestica na rubu zahvaćenoga područja ne smiju mijenjati zato što je granica zahvaćenoga područja definirana geometrijskom unijom zahvaćenih čestica. U slučaju da se mijenjaju granice katastarskih čestica na granici zahvaćenoga područja tada

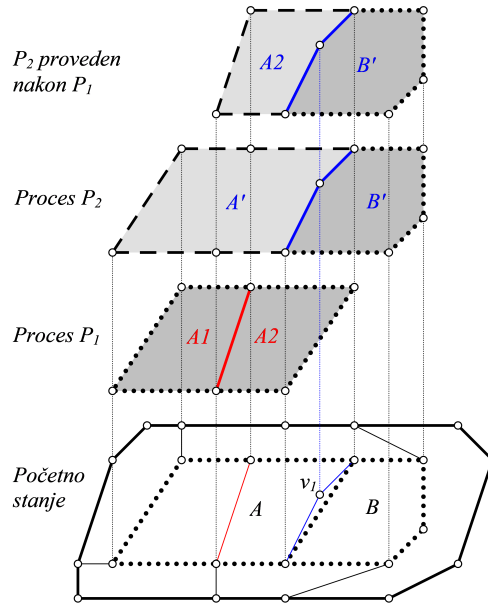
polu-bridovi zahvaćenoga područja neće imati komplemente što će prouzročiti topološku pogrešku. Ako je potrebno mijenjati granicu susjednih katastarskih čestica, tada se susjedna čestica koja formalno ne sudjeluje u procesu dodaje u proces kao tehnički zahvaćena čestica. U suprotnome se dodatno provjerava i uspoređuje vremenski period između trenutka u mjestu  $P1$  i  $P5$ . Ukoliko postoji preklapanje između ta dva perioda, smatra se da su procesi konkurentni. Proces koji je ranije provjerio ispravnost (došao prvi u mjesto  $P3$ ) označava se ispravnim i može nastaviti prema provedbi, a ostali se konkurentni procesi vraćaju na ponovnu pripremu ili se otkazuju.



Slika 4.18: Optimistični tijek rada

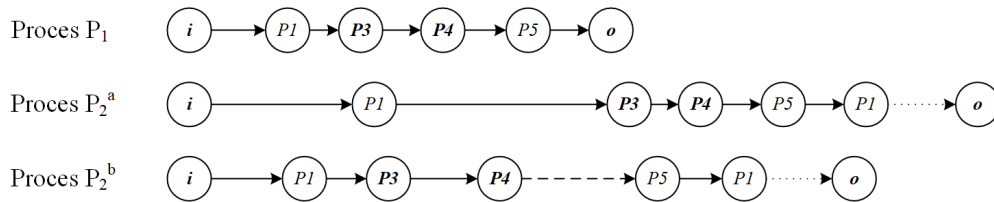
Ukoliko postoje konkurentni procesi, proces se ne odbija automatski već pokušava dohvatiti nove verzije konkurentnih objekata, a proces se vraća na ponovno pripremanje i provjeravanje ispravnosti. Vraćanjem se procesa na ponovnu pripremu pokušava postići serijalizabilni raspored provedbe procesa. Međutim, ukoliko se radi o konfliktnim konkurentnim procesima gdje se mijenjao identitet uključenih objekata, proces se mora otkazati, a promjene poništiti, jer postoji mogućnost da su se promijenile granice zahvaćenoga područja. Dakle, kod konkurentnih procesa integracija je promjena iz konkurentnog procesa moguća, dok kod konfliktnih konkurentnih procesa to nije moguće.

Na slici 4.19 prikazana su dva konfliktna konkurentna procesa. U procesu  $P_1$  katastarska se čestica  $A$  dijeli na dvije nove čestice  $A1$  i  $A2$ . Ne mijenjaju se granice zahvaćenoga područja pa nema tehnički zahvaćenih katastarskih čestica. U drugome se konkurentnom procesu  $P_2$  radi korekcija granice između katastarskih čestica  $A$  i  $B$ . Identiteti će obje katastarske čestice provedbom toga procesa biti zadržani. Istovremenom bi se provedbom konfliktnih konkurentnih procesa  $P_1$  i  $P_2$  pojavila topološka pogreška (preklapanje čestica  $B'$  i  $A2$ ) te duple čestice na istome mjestu (čestice  $A$  i  $A1, A2$ ).



Slika 4.19: Konfliktni konkurentni procesi (spajanje čestica i korekcija granice)

Na slici 4.20 prikazan je vremenski redoslijed provedbe konkurentnih procesa prema optimističnome tijeku rada. Proces  $P_1$  prvi je provjerio ispravnost procesa, nakon čega provjerava postoje li konkurentni procesi. Iako je proces  $P_2$  (opcija  $P_2^b$ ) provjeren uspješno, u međuvremenu je ipak provjeren nakon procesa  $P_1$  te proces  $P_1$  nastavlja s izvođenjem i zapisuje promjene te završava. Drugi mogući vremenski redoslijed, proces  $P_2^a$  koji stoji u stanju pripreme duže vrijeme te se  $P_1$  u međuvremenu u potpunosti provodi. Tek nakon toga  $P_1$  provjerava ispravnost, a nakon toga i konkurentne procese. Budući da se proces  $P_1$  proveo u međuvremenu, potrebno je dohvatiti najnovije verzije ako su zahvaćene katastarske čestice zadržale identitet. Ukoliko se radi o konfliktnim konkurentnim procesima kao što je u primjeru na slici 4.19 gdje katastarska čestica  $A$  nestaje u procesu  $P_1$ , proces se otkazuje, jer se zahvaćeno područje procesa mora drugačije definirati.



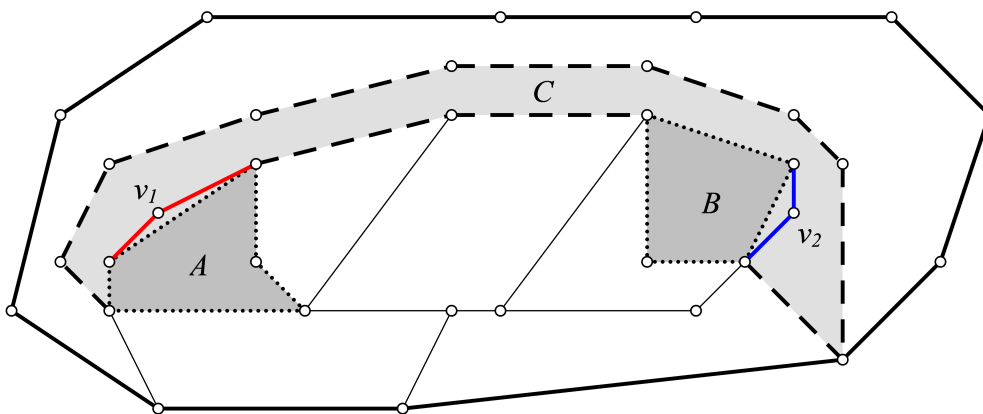
Slika 4.20: Provedba procesa prema optimističnome tijeku rada

Potencijalni je problem optimističnoga tijeka rada da određeni proces bude iznova odbijan i vraćan na ponovno pripremanje promjene. To se može dogoditi u slučaju da konkurentni proces koji je prvi provjerio ispravnost iz nekog razloga ostane u toj fazi i ne nastavi prema provedbi. Drugi se konkurentni proces u tome slučaju ne može provesti dokgod se ne provede proces koji je provjerio ispravnost prvi. Druga je mogućnost da

više procesa radi izmjene na istoj katastarskoj čestici gdje svaki put različiti proces napravi promjene na čestici i provjeri ih prije istoga procesa koji se vraća na ponovno pripremanje. Ipak, optimistični tijek rada donosi povećanje konkurentnosti procesa u odnosu na pesimistični tijek rada jer omogućava istovremeno pripremanje promjena na katastarskim česticama. Ako je validacija procesa neuspješna, vraća se na ponovno pripremanje promjene pri čemu se promjene iz konkurentnoga procesa koji je ranije provjeren integriraju u neuspješno validirani proces što također pridonosi ubrzanju pripreme promjene, a time i povećanju konkurentnosti procesa. Ako se kod konfliktnih konkurentnih procesa mijenja identitet zahvaćenih katastarskih čestica to znači da se konfliktni konkurentni proces mora otkazati, a zahvaćeno područje iznova definirati, jer čestica koja je definirana u zahvaćenome području po provedbi procesa neće više postojati.

## 4.5. Poboljšanje konkurentnosti procesa uvođenjem međnih točaka

U dosadašnjim je razmatranjima osnovna jedinica nad kojom se osiguravala ispravnost procesa bila katastarska čestica, odnosno poligon kojim je prostorno predstavljena. Ograničavanjem pristupa čitavim katastarskim česticama može se uzrokovati da se onemogućuje promjene na velikome broju katastarskih čestica, pogotovo ako se radi o velikim ili izduženim česticama poput puteva ili vodotoka. Na slici 4.21 prikazana su dva konkurentna procesa na udaljenim katastarskim česticama koje imaju zajedničku susjednu izduženu katastarsku česticu (primjerice, put ili vodotok). Dokgod traje proces nad katastarskom česticom *A*, nijedan konkurentni proces ne može pripremiti, odnosno provesti promjenu na izduženoj katastarskoj čestici *C*. Iako se promjene iz konkurentnih procesa nad česticama *A* i *B* sa slike 4.21 mogu jednostavno integrirati, zbog očuvanja serijalizabilnosti i oporavka procesa potrebna je provedba ispravnim redoslijedom svakoga procesa.

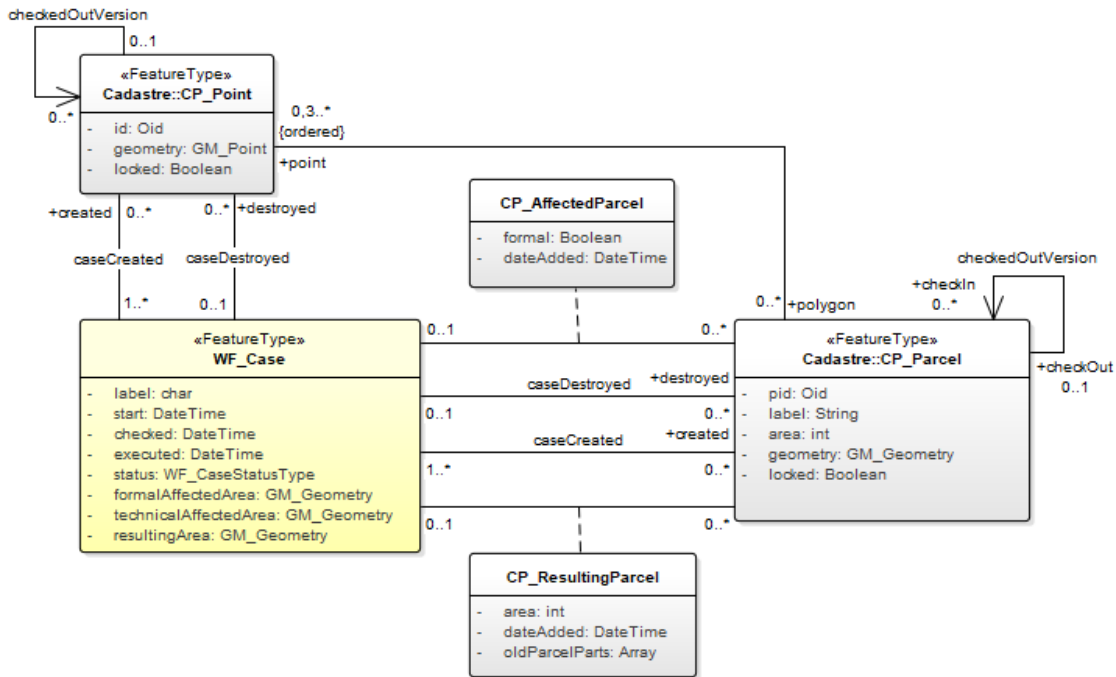


Slika 4.21: Prednost uvođenja međnih točaka u model podataka

Zbog toga je proširen dijagram klasa modela procesa s točkama kako bi se omogućilo



osiguravanje ispravnosti procesa na razini točke (Slika 4.22) što omogućava da se zaključa manji skup podataka, odnosno poveća konkurentnost procesa.



Slika 4.22: Dijagram klasa modela procesa proširen s točkama

Samom se usporedbom topološkog odnosa poligona može utvrditi je li se promjena na poligonu dogodila te se može odrediti prostorni odnos između dva poligona. Preciznije se određivanje vrste i lokacije promjene može utvrditi usporedbom svakoga pojedinačnog čvora poligona. Slijedeći koncepte iz rada Matijević i dr. (2008) u kojemu je opisana metodologija kontrole ispravnosti ravninske particije kada se prostorna sastavnica temelji na poligonima, u nastavku je objašnjen način određivanja vrste i lokacije promjene na razini čvora poligona rastavljanjem geometrijske strukture poligona na topološku polu-brid strukturu. Prilikom rastavljanja na polu-brid strukturu svaki se čvor poligona uspoređuje s međnom točkom kako bi preuzeo njezin identitet na temelju kojega se može odrediti status čvora. Određivanjem je statusa svakoga čvora određen status polu-bridova pomoću kojih se može testirati ispravnost ravninske particije. S obzirom na to da je uvođenjem međnih točaka omogućeno preciznije određivanje vrste i lokacije promjene, definiran je tijekom rada temeljen na ideji altruističnoga zaključavanja koje omogućuje doniranje prava izmjene na podacima drugome procesu što može omogućiti veću konkurentnost procesa.

#### 4.5.1. Prepoznavanje promjena

Skup formalno i tehnički zahvaćenih katastarskih čestica (odnosno poligona koji ih predstavljaju) označit ćemo s  $F_i$ , a skup rezultirajućih katastarskih čestica s  $F_o$ . Zahvaćeni i

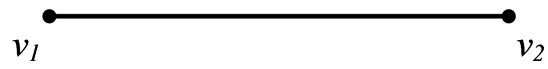
rezultirajući poligoni katastarskih čestica rastavljaju se na polu-bridove i čvorove. Prilikom rastavljanja poligona na polu-brid strukturu za svaki se čvor određuje status (vrsta promjene) te identitet koji se preuzima od međne točke. Moguće je provjeriti ispravnost i bez uvođenja međnih točaka kao što je objašnjeno u (Matijević i dr., 2008), ali u tome slučaju čvorovi imaju lokalni identitet u okviru procesa te se ne mogu zaključavati, jer ne postoje izvan procesa. Uvođenjem međnih točaka, svakome se čvoru pronalazi odgovarajuća međna točka koja ima globalni identitet na razini baze podataka te se nad njom mogu obavljati operacije zaključavanja i otključavanja. Status je polu-brida određen statusom početnog i krajnjega čvora. Za testiranje mogućih promjena na jednome polu-bridu u obzir su uzeta dva svojstva krajnjih čvorova:

- geometrija
- identitet

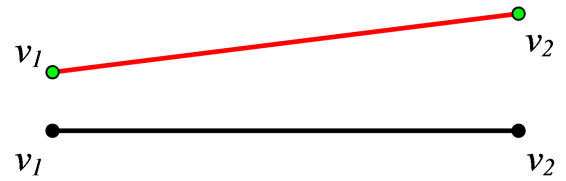
Na slici 4.23  $v_1$  predstavlja početni, a  $v_2$  krajnji čvor promatranoga polu-brida. Mogući slučajevi koji se mogu pojaviti kod svakoga čvora (ukupan je broj mogućih varijacija navedenih promjena čvorova na polu-bridu  $n = 4^2 = 16$ ):

- nema promjene - fiksno ( $F$ ): čvor je identičan u zahvaćenom i rezultirajućemu poligonu;
- promjena geometrije ( $G$ ): čvor ima identičan identitet, ali drugačiju geometriju;
- dodavanje čvora ( $N$ ): čvor postoji u rezultirajućemu, ali ne i u zahvaćenomu poligonu;
- brisanje čvora ( $B$ ): čvor postoji u zahvaćenomu, ali ne i u rezultirajućemu poligonu.

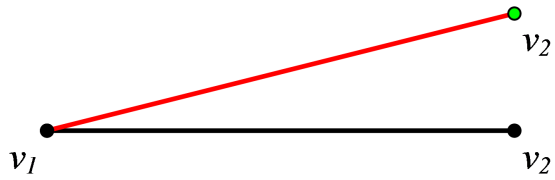
Na slici 4.23 polu-bridovi prikazani crnom bojom su polu-bridovi zahvaćenoga poligona, a crveni su polu-bridovi rezultirajućega poligona. Čvorovi označeni crnom bojom čvorovi su zahvaćenoga poligona, a crvenom bojom čvorovi zahvaćenoga poligona koji se brišu. Čvorovi označeni zelenom bojom čvorovi su rezultirajućega poligona koji su nastali promjenom ulaznoga čvora ili se radi o novome čvoru. Ukoliko je identitet (naziv) čvora isti, radi se o geometrijskoj promjeni čvora. Ukoliko se naziv čvora razlikuje (različit identitet), radi se o novome čvoru.



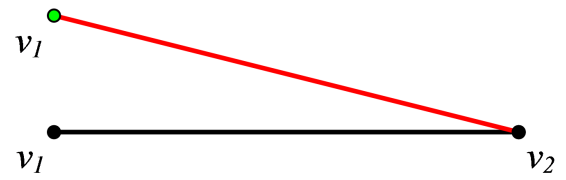
(a)  $v_1 = F; v_2 = F$



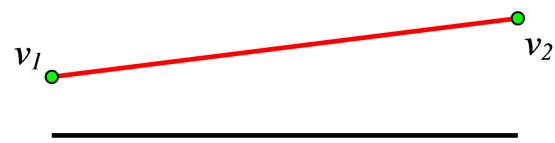
(b)  $v_1 = P; v_2 = P$



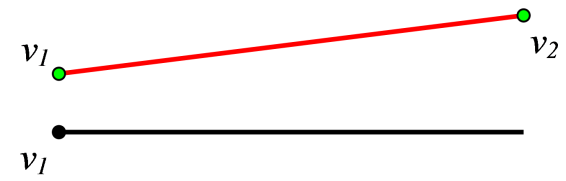
(c)  $v_1 = F; v_2 = P$



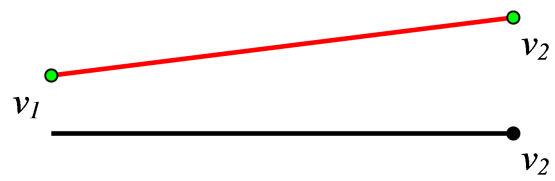
(d)  $v_1 = P; v_2 = F$



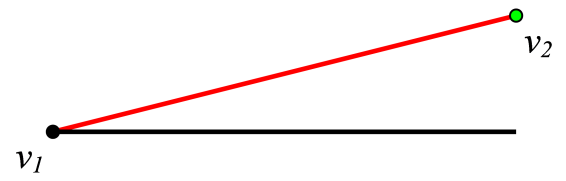
(e)  $v_1 = N; v_2 = N$



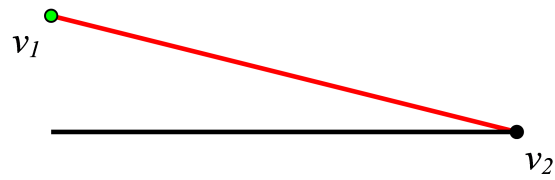
(f)  $v_1 = P; v_2 = N$



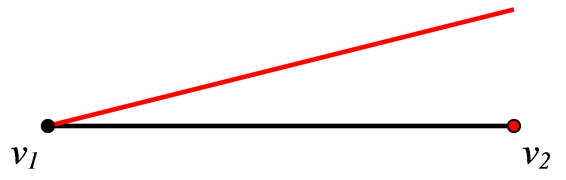
(g)  $v_1 = N; v_2 = P$



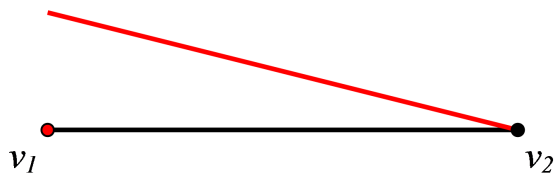
(h)  $v_1 = F; v_2 = N$



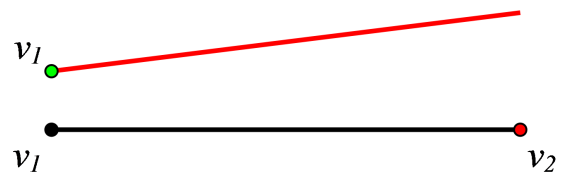
(i)  $v_1 = N; v_2 = F$



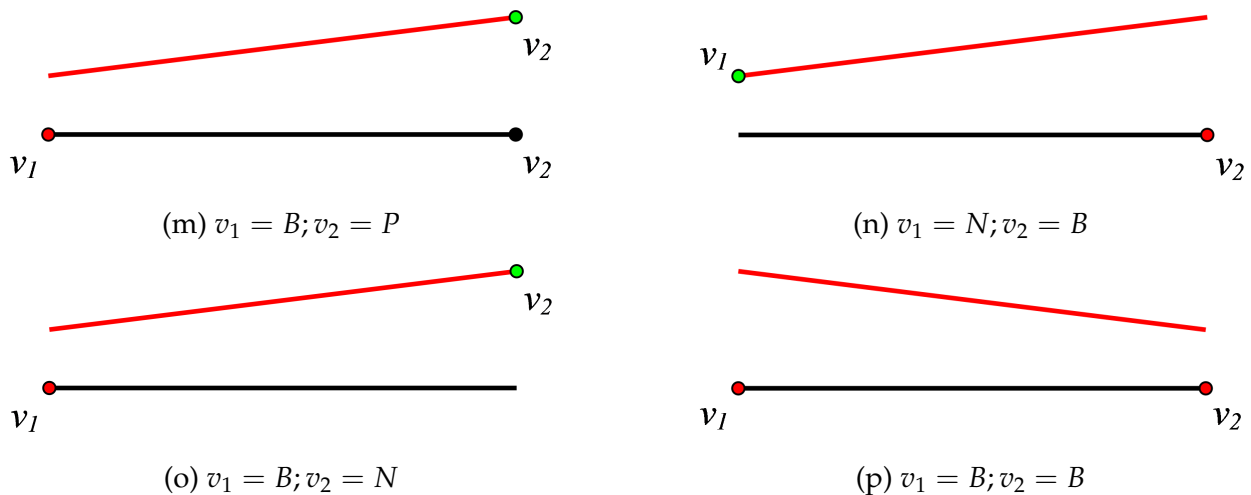
(j)  $v_1 = F; v_2 = B$



(k)  $v_1 = B; v_2 = F$



(l)  $v_1 = P; v_2 = B$



Slika 4.23: Moguće promjene nad čvorovima

Slika 4.23 prikazuje sve varijacije promjena na krajnjim čvorovima s obzirom na definirana svojstva, identitet i geometriju. S obzirom na utjecaj koji određena promjena na čvoru ima na poligon, slučajevi se sa slike 4.23 mogu podijeliti kao što je prikazano u tablici 4.1.

Grupe promjena prikazane u tablici 4.1 također predstavljaju moguće vrijednosti za status polu-brida. Svakome se polu-bridu dodjeljuje atribut status i smjer koji se koriste za ispitivanje ispravnosti ravninske particije. Geometrijska promjena (G) uzrokovana je promjenom položaja jednog ili oba čvora polu-brida. Ako su oba čvora promijenjena, smjer je 0 (nula). Ako je samo završni čvor promijenjen, smjer je pozitivan (+), a ako je samo početni čvor promijenjen, smjer je negativan (-). Te se vrijednosti koriste kasnije kod provjere ispravnosti ravninske particije.

Topološka promjena (T1) nastaje dodavanjem novoga čvora na postojeći polu-brid. Ako su oba čvora nova, smjer je 0 (nula). Ako je samo završni čvor novi, smjer je pozitivan (+), a ako je samo početni čvor novi, smjer je negativan (-).

Topološka promjena (T2) predstavlja promjenu statusa iz geometrijske promjene u topološku i obratno. Ukoliko je promjena u smjeru s promijenjenoga na novi čvor, smjer je negativan (-), a u suprotnome je pozitivan.

Topološka promjena (T3) predstavlja slučaj kada su jedan ili oba čvora promatranoga polu-brida obrisani. Brisani su čvorovi vezani uz zahvaćeni poligon te se ne pojavljuju u rezultirajućem poligonu, ali svejedno su bitni jer uzrokuju topološku promjenu na zahvaćenim poligonima.

Čvorovi označeni za brisanje mogu se lako utvrditi razlikom skupova čvorova zahvaćenih i rezultirajućih poligona prema identitetu svakoga čvora. Uklanjanjem se brisanih čvorova iz zahvaćenoga poligona pojednostavljuje postupak prepoznavanja ostalih vrsta promjena. Zbog toga se prepoznavanje promjena dijeli na dva dijela: prepoznavanje promjene brisanja koja je vezana uz zahvaćene poligone te prepoznavanje promjene dodavanja i izmjene čvorova koje su vezane uz rezultirajuće poligone (Slika 4.24).

Tablica 4.1: Vrste promjena na polu-bridovima

Vrsta promjene	Varijacija	Slučajevi sa slike 4.23
Nema promjene (F)	Fiksni-fiksni	4.23a
	Promijenjeni-promijenjeni ( $G^0$ )	4.23b
Geometrijska promjena (G)	Fiksni-promijenjeni ( $G^+$ )	4.23c
	Promijenjeni-fiksni ( $G^-$ )	4.23d
Topološka promjena (T1)	Novi-novi ( $T1^0$ )	4.23e
	Fiksni-novi ( $T1^+$ )	4.23i
	Novi-fiksni ( $T1^-$ )	4.23h
Topološka promjena (T2)	Promijenjeni-novi ( $T2^-$ )	4.23f
	Novi-promijenjeni ( $T2^+$ )	4.23g
Topološka promjena (T3)	Fiksni-brisani	4.23j
	Brisani-fiksni	4.23k
	Promijenjeni-brisani	4.23l
	Brisani-promijenjeni	4.23m
	Novi-brisani	4.23n
	Brisani-novi	4.23o
	Brisani-brisani	4.23p

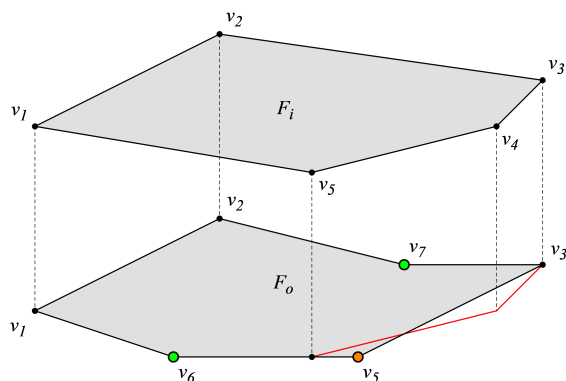
**Definicija 4.3 (Skup brisanih čvorova)** Skup brisanih čvorova ( $S_D$ ) je rezultat razlike skupa čvorova zahvaćenih  $F_i$  i rezultirajućih  $F_o$  poligona. Kriterij prema kojem se čvorovi uspoređuju jest identitet čvora.

$$S_D = F_i \setminus F_o = \{v_j \in S_D : v_j \in F_i \wedge v_j \notin F_o\}$$

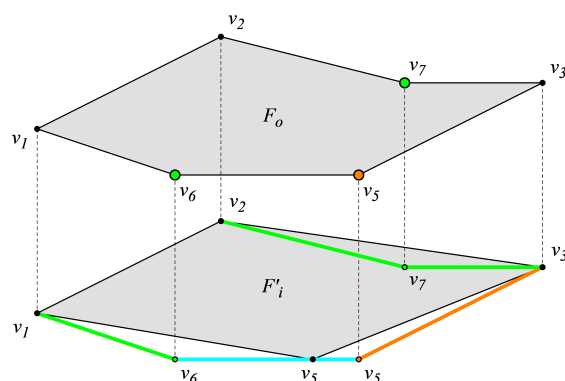
Poligon iz kojega su uklonjeni brisani čvorovi nazvat ćemo reducirani poligon ( $F'_i$ ) i on se upotrebljava u daljnjemu postupku određivanja statusa čvorova:

$$F'_i = F_i \setminus S_D$$

Na slici 4.24a prikazan je zahvaćeni  $F_i = \{v_1, v_2, v_3, v_4, v_5, v_1\}$  i rezultirajući poligon  $F_o = \{v_1, v_2, v_7, v_3, v_5, v_6, v_1\}$ . U ovome slučaju skup brisanih čvorova ima samo jednoga člana  $S_D = \{v_4\}$ . Iz zahvaćenoga se poligona uklanja skup brisanih čvorova i dobiva se reducirani poligon  $F'_i$  koji se dalje koristi za određivanje statusa preostalih čvorova (Slika 4.24b).



(a) Prepoznavanje brisanih čvorova



(b) Prepoznavanje novih i promijenjenih čvorova

Slika 4.24: Prepoznavanje promjena na poligonu

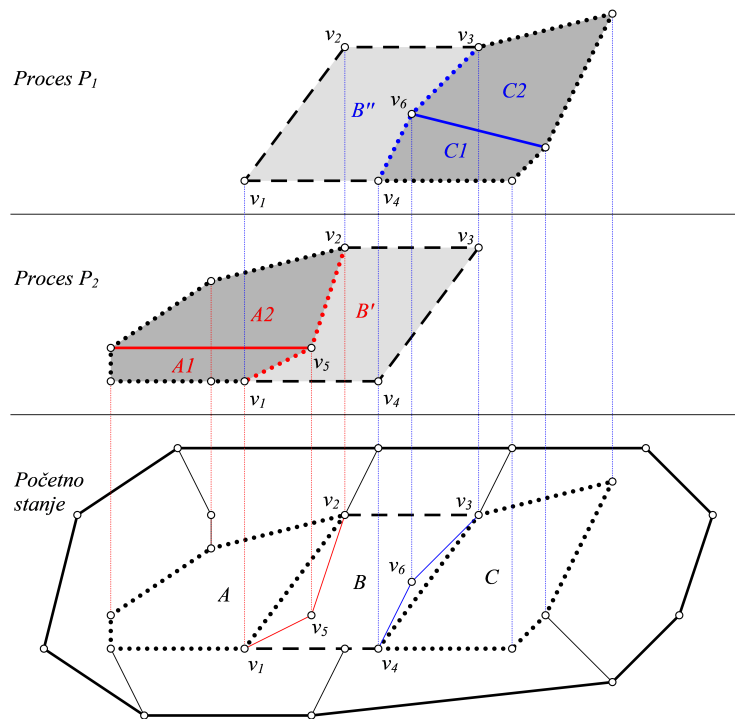
Na slici 4.24b prikazan je postupak određivanja statusa svakoga polu-brid rezultirajućega poligona. Zelenom je bojom označena topološka promjena  $T1$ , narančastom bojom geometrijska promjena  $G$  i tirkiznom bojom topološka promjena  $T2$ . Nakon što je svakome polu-bridu određen status i smjer promjene, ispituje se ispravnost ravninske particije. Kod topološke polu-brid strukture, svaki polu-brid mora imati komplement da bi ravninska particija bila ispravna. Budući da je sada polu-brid nositelj informacije o statusu (odnosno vrsti promjene i smjeru te promjene), potrebno je preciznije definirati pojam komplementarnih polu-bridova na temelju kojih se ispituje ispravnost ravninske particije.

**Definicija 4.4 (Ispravnost ravninske particije)** Svaki polu-brid  $b_{xy}$  mora imati komplement  $b_{yx}$  istoga statusa, ali suprotnog smjera ako je status ovisan o smjeru. Ako je status polu-brida invarijantan na smjer (npr.  $T1^0$ ,  $T2^0$ ,  $G^0$ ) tada komplementarni polu-bridovi moraju imati identičan status.

Prethodna definicija znači da ako je status polu-brida  $b_{12}$ , primjerice, topološka promjena ( $T1$ ) ( $T1^+$ ), njegov komplement  $b_{21}$  mora imati isti status, ali suprotnoga smjera ( $T1^-$ ). Primjer je drugoga slučaja iz definicije kada polu-bridovi  $b_{12}$  i  $b_{21}$  imaju isti status ( $T1^0$ ) invarijantan na smjer. Ukoliko to nije slučaj, ravninska particija na tome mjestu nije ispravna te je potrebno promjenu propagirati s testiranoga poligona na susjedni.

#### 4.5.2. Altruistični tijek rada

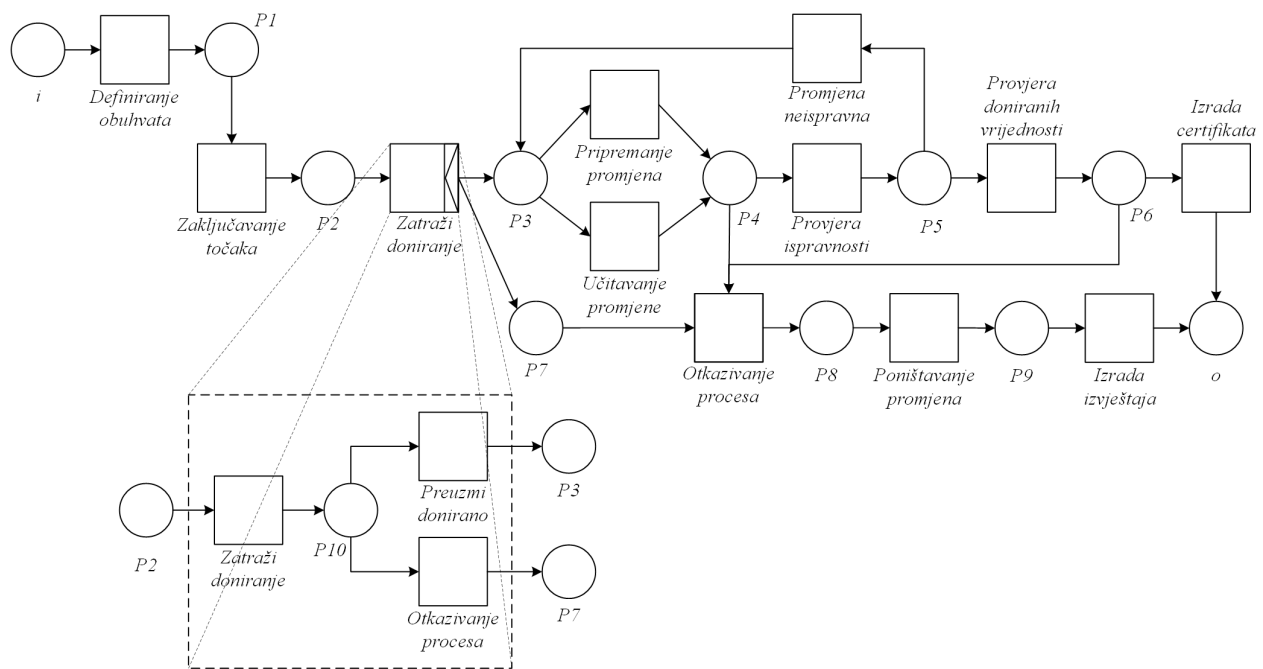
Pesimistični i optimistični tijek rada mogu se primijeniti i na procese koji uključuju upravljanje međnim točkama povezanih s katastarskim česticama, ali je potrebno uvesti dodatne uvjete integriteta kako bi se zadržala ispravnost prostornog odnosa točaka i poligona kao i referencijalni integritet između poligona i točaka.



Slika 4.25: Konkurentni procesi (dioba čestica) - s točkama

Situacija prikazana na slici 4.25 ne predstavlja konkurentne procese na razini točaka jer se promjene događaju na različitim čvorovima, ali ipak predstavlja konkurentne procese budući da se rade promjene na istome poligonu (katastarskoj čestici). Istovremenom bi se provedbom oba procesa pojavio problem izgubljene vrijednosti. S obzirom da su u model podataka uključene točke, moguće je obavljati operacije nad njima te je na slici 4.26 prikazan altruistični tijek rada koji omogućava doniranje prava izmjene na privatnoj kopiji matičnoga procesa konkurentnom procesu i zatim vraćanje promjena iz konkurentnoga procesa matičnome procesu. Budući da je svaki čvor preuzeo identitet od međne točke

te je svakom čvoru određen status, moguće je prepoznati promjenu koja se dogodila u konkurentnome procesu i integrirati je u matični proces.



Slika 4.26: Altruistični tijek rada

Altruistični tijek rada temelji se na pesimističnome tijeku rada, jer se ispravnost podataka kod konkurentnih procesa osigurava zaključavanjem na razini točaka. Zaključavaju se točke koje su vezane uz formalno zahvaćene katastarske čestice. Nakon zaključavanja točaka provjerava se postoje li konkurentni procesi koji bi mogli donirati pravo izmjene na zajedničkim podacima. Ukoliko proces  $P_1$  donira pravo izmjene podataka procesu  $P_2$ , tada se proces  $P_1$  ne može provesti dok se ne provede proces  $P_2$  i time otkluča zajedničke podatke. Ta se provjera obavlja u prijelazu *Provjera doniranih vrijednosti*. Provjerom vrijednosti atributa *executed* utvrđuje se je li konkurentni proces kojemu je donirano pravo izmjene podataka proveden. Ako konkurentni proces ima dodijelenu vrijednost atributu *executed*, znači da je proveden. Ako je konkurentni proces proveden, matični se proces može također provesti. U suprotnome, proces mora ostati u stanju čekanja dok ne završi konkurentni proces. Ideja je altruističnoga tijeka rada da jednostavni procesi kratkoga trajanja zatraže pravo doniranja od kompleksnog i dugotrajnoga matičnog procesa kako bi mogli napraviti manju izmjenu na podacima koju po provedbi vraćaju matičnome procesu. Da bi se spriječilo da konkurentni proces dugo traje i matični proces mora čekati da se provede konkurentni proces, može se uvesti pravo matičnoga procesa da povuče pravo doniranja konkurentnome procesu. Takva se radnja ne smije obaviti nakon mjesta  $P_5$  kako bi se spriječilo da matični proces povuče pravo doniranja, a konkurentni proces u međuvremenu provjeri svoju ispravnost i bude proveden što bi moglo prouzročiti nepredvidive rezultate. Prilikom povlačenja prava doniranja konkurentnome procesu provje-



rava se trenutno stanje konkurentnoga procesa. Ako je oznaka u mjestu  $P5$  ili dalje, pravo se doniranja ne može povući.

Konkurentni proces može zatražiti pravo doniranja promjene, a drugi proces može to pravo donirati ili odbiti doniranje. Ukoliko proces odbije doniranje, konkurentni se proces mora otkazati ili ostati u statusu čekanja dok se proces od kojega je zatraženo doniranje ne provede. Odluka o doniranju može biti određena automatski logikom tijekom rada ili na temelju odluke operatera zaduženoga za upravljanje procesom. U slučaju da je za odluku zadužen operater, on može procijeniti hoće li proces dugo trajati te u skladu s procjenom prihvatiti ili odbiti doniranje prava izmjene konkurentnome procesu.

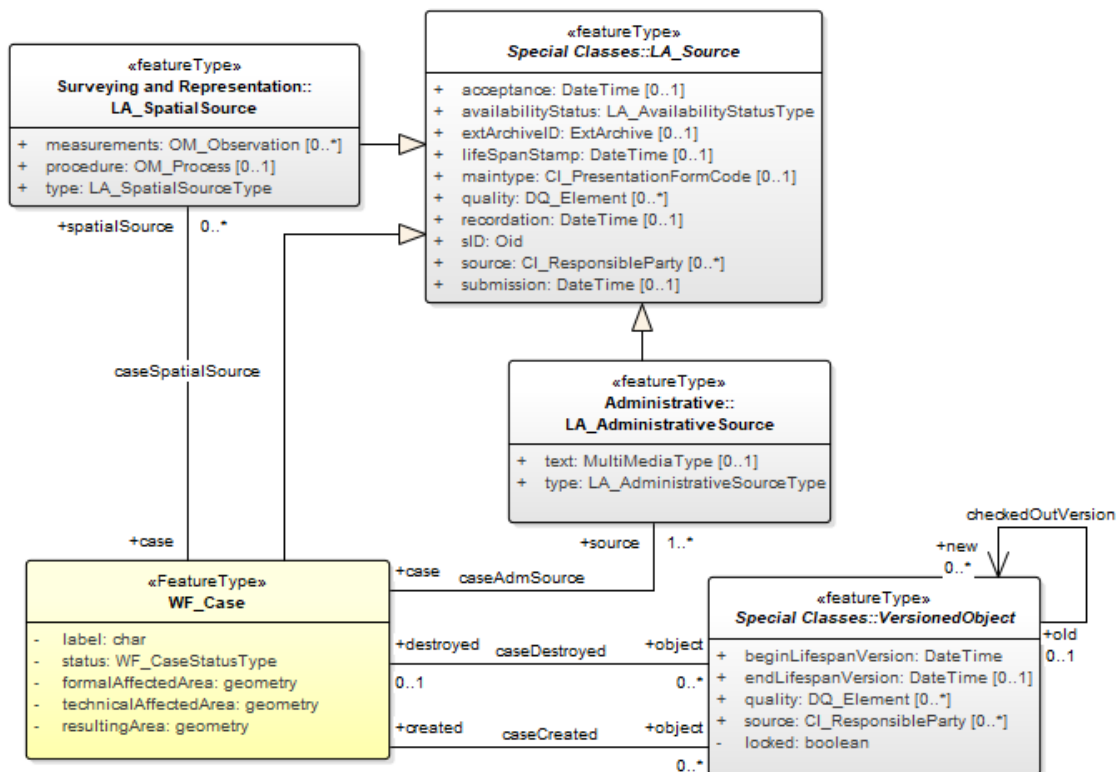
Doniranjem prava izmjene između procesa moguće je povećati konkurentnost, pogotovo u slučajevima kada kompleksni procesi koji dugo traju doniraju pravo promjene jednostavnijim procesima koji traju kratko. Da bi se spriječilo da se doniranje prava zloupotrijebi tako da matični proces koji je donirao pravo promjene mora čekati konkurentni proces da se provede jer neočekivano dugo traje, može se uvesti pravo matičnoga procesa da povuče pravo doniranja konkurentnome procesu, ali isključivo prije provjere ispravnosti konkurentnoga procesa kako bi se spriječilo uvođenje nekonzistentnosti u podatke.

Primjenom altruističnoga tijekom rada konkurentni procesi sa slike 4.21 mogu istovremeno pripremati promjene na katastarskim česticama  $A$  i  $B$ , jer se katastarska čestica  $C$  ne zaključava, već se zaključavaju točke vezane uz formalno zahvaćene katastarske čestice  $A$  i  $B$ . Ako proces nad katastarskom česticom  $A$  prvi započne, tada proces nad katastarskom česticom  $B$  može zatražiti doniranje prava izmjene na katastarskoj čestici  $C$ . Nakon što doda čvor  $v_2$  na katastarsku česticu  $C$  i završi pripremu promjena, proces se nakon provjere provodi, a promjene se vraćaju u proces nad česticom  $A$  koji je u međuvremenu mogao raditi promjene na čestici  $A$ , ali ne i na  $C$ . Nakon što se provede proces nad česticom  $B$ , može se provesti proces nad česticom  $A$  ukoliko je u skladu s uvjetima integriteta.

### 4.6. Integracija WFMS-a u LADM

LADM podržava pohranjivanje događaja i njihovo povezivanje s ostalim klasama korištenjem klase *LA\_Source*, ali ne podržava modeliranje interne strukture događaja, odnosno redoslijed zadataka unutar procesa (događaja). Zbog toga je prethodno razvijeni prošireni konceptualni model WFMS-a integriran u LADM kako bi se omogućila podrška za modeliranje procesa nad LADM klasama te se provjerila mogućnost integracije postavljenih koncepata u LADM. Slika 4.27 prikazuje povezanost WFMS-a s LADM-om. Veza se ostvaruje preko klase *WF\_Case* koja je potomak klase *LA\_Source* i vezana je na klasu *VersionedObject* dvjema vezama *caseCreated* i *caseDestroyed*. Većina LADM klasa su potomci klase *VersionedObject* i zato je ona povezana s klasom *WF\_Case* budući da svi potomci klase *VersionedObject* nasljeđuju veze na klasu *WF\_Case*. Jedan je od glavnih ciljeva ovoga rada modeliranje procesa nad katastarskim česticama čija je prostorna sastavnica predstavljena poligonima. Zato je u nastavku objašnjeno povezivanje LADM prostornih jedinica (od-

nosno katastarskih čestica) s WFMS-om. Glavni su elementi za modeliranje procesa nad ostalim LADM klasama kao i drugačijim prostornim prezentacijama prostornih jedinica definirani u ovome radu (modeliranje povezanosti procesa s podacima, modeliranje temporalne sastavnice, transakcijski koncepti i mogućnost definiranja uvjeta integriteta na razini zadatka), ali dodatna su istraživanja potrebna kako bi se istražili specifični zahtjevi koji se postavljaju kod drugih LADM klasa ili drugih prostornih prezentacija prostornih jedinica. Na klasu *WF\_Case* vezane su klase *LA\_SpatialSource* i *LA\_AdministrativeSource* koje pohranjuju podatke o izvornim podacima izmjere i službenim dokumentima prema kojima se određeni proces odvija. Veza na klasu *LA\_SpatialSource* nije obavezna, jer je moguće da procesi u sustavima za upravljanje zemljištem ne zahtijevaju izmjeru (primjerice, u slučaju promjene nositelja prava) dok je veza na *LA\_AdministrativeSource* obavezna, jer se svi procesi u sustavima za upravljanje zemljištem moraju temeljiti na odgovarajućim službenim dokumentima. Klase *LA\_SpatialSource* i *LA\_AdministrativeSource* također su potomci klase *LA\_Source* čime je zadržana kompatibilnost s trenutnom verzijom LADM-a. Budući da svaki proces radi izmjene na privatnim kopijama podataka, na klasu *VersionedObject* je dodana refleksivna veza *checkedOutVersion* koju nasljeđuju svi njezini potomci. Klasa *VersionedObject* također je proširena s dodatnim atributom *locked* koji omogućava zaključavanje objekata radi onemogućavanja pristupa konkurentnim procesima.



Slika 4.27: Dodavanje podrške za modeliranje procesa u LADM

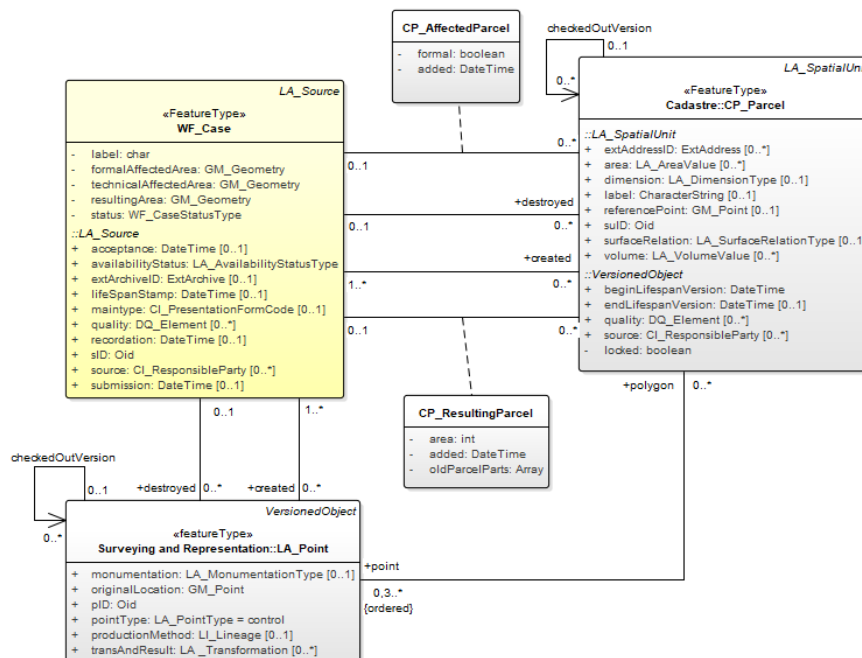
Kao potomak klase *LA\_Source*, klasa *WF\_Case* preuzela je njezine attribute koji se tiču

kvalitete i temporalne sastavnice. Temporalni atributi klase *WF\_Case* sa slike 4.9 zamijenjeni atributima klase *LA\_Source*:

- *submission* označava početak procesa (atribut *start* sa slike 4.9),
- *recordation* označava vrijeme provedbe procesa, odnosno transakcijsko vrijeme (atribut *executed* sa slike 4.9),
- *acceptance* predstavlja vrijeme valjanosti, a označava kada je proces stupio na snagu (u formalnome, odnosno pravnome smislu).

Budući da je pravni dio procesa izvan opsega ovog rada, ovdje se koristi atribut *recordation* kao temporalni atribut koji određuje trenutak u kojemu je određen proces proveden.

Model povezivanja katastarskih čestica s WFMS-om (Slika 4.28) razlikuje se od modela na slici 4.9 po tome što je katastarska čestica (klasa *CP\_Parcel*) potomak klase *LA\_Parcel* koja je pak potomak klase *VersionedObject* te od njih nasljeđuje sve atribute. Klasa *CP\_Parcel* povezana je s klasom *WF\_Case* s dvije veze koje je naslijedila od klase *VersionedObject* (veze *caseCreated* i *caseDestroyed*) te s dvije vezne klase, *CP\_AffectedParcel* i *CP\_ResultingParcel*.



Slika 4.28: Veza između WFMS-a i LADM katastarskih čestica

Model sa slike 4.28 sličan je proširenome konceptualnom modelu sa slike 4.22 te podržava modeliranje procesa na pesimističan, optimističan i altruističan način.

## 4.7. Rekapitulacija

U skladu s referentnim modelom WFMS-a izrađen je konceptualni model koji omogućava pohranjivanje elemenata tijeka rada za modeliranje procesa u sustavima za upravljanje zemljištem. Opći model podržava modeliranje procesa na katastarskim podacima, a prošireni je model specijaliziran kako bi podržao specifične zahtjeve koje postavljaju katastarske čestice čija se prostorna sastavnica temelji na poligonima. Prostornom je definicijom zahvaćenoga područja procesa osigurana ispravnost prostorne sastavnice katastarskih čestica.

Na temelju su toga konceptualnog modela definirana tri tijeka rada za osiguravanje ispravnosti prostorne sastavnice katastarskih čestica u slučaju konkurentnih procesa te su prikazani pomoću WF mreža. Pesimistični tijek rada osigurava ispravnost zaključavanjem svih katastarskih čestica unutar zahvaćenoga područja. Optimistični tijek rada ne zaključava katastarske čestice te ostavlja validaciju za posljednju fazu procesa prije provedbe. Altruistični tijek rada omogućava osiguravanje ispravnosti kod modela podataka koji uključuje međne točke i podržava konkurentne procesa dok konfliktne procese ne podržava. Primjenom se bilo kojega tijeka rada osigurava ispravnost prostorne sastavnice katastarskih čestica te izolacija u slučaju konkurentnih procesa. Različiti tijekovi rada podržavaju različite razine konkurentnosti procesa. Optimistični i altruistični tijek rada omogućavaju istovremenu pripremu promjena na katastarskim česticama čime se povećava učinkovitost procesa, jer se smanjuje vrijeme čekanja.

Prijedlog je integracije WFMS-a u LADM definiran na generalnoj razini, a za katastarske je čestice i međne točke detaljnije razrađen. Integracijom je WFMS-a u LADM dodana podrška za modeliranje procesa nad LADM klasama čime je dokazana primjenjivost konceptualnoga modela transakcijskoga WFMS-a na LADM.

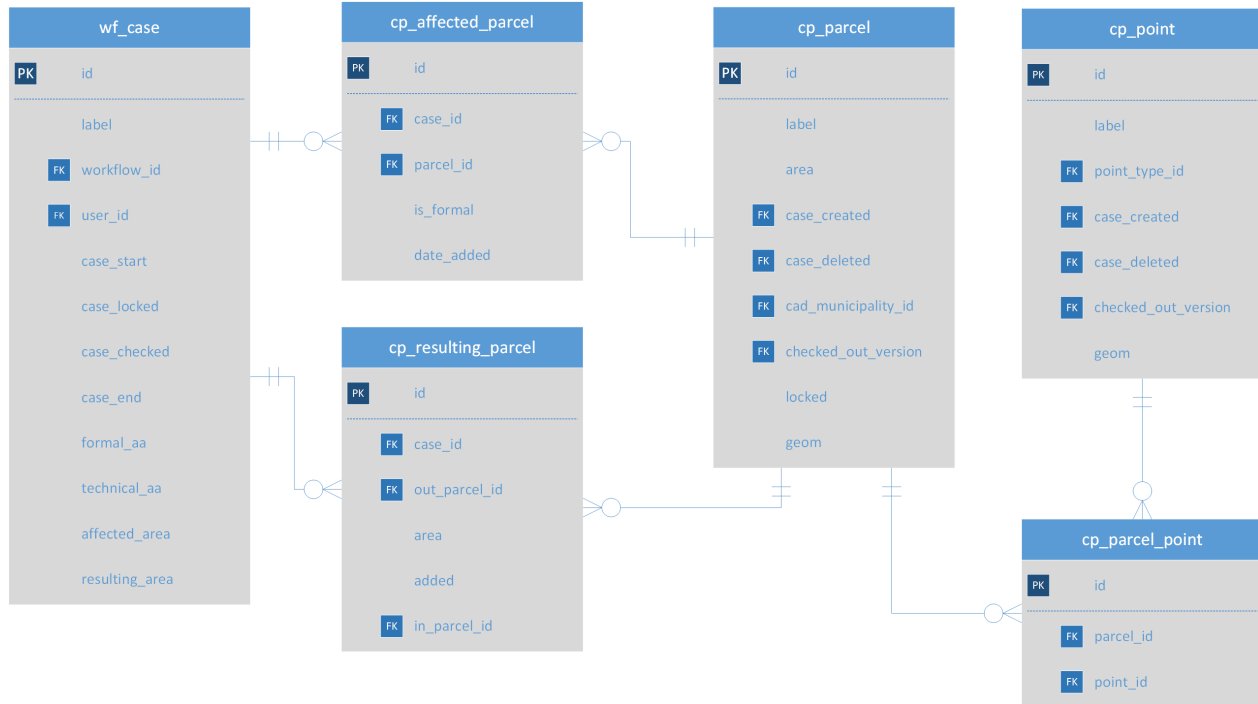
## 5. Arhitektura testnoga sustava i provjera slučajeva korištenja

Testni je sustav implementiran kako bi se provjerila istinitost postavljenih hipoteza te primjenjivost definiranoga konceptualnog modela. U okviru je ovog rada implementiran transakcijski WFMS koji omogućava definiciju i interpretaciju procesa. Definirani procesi djeluju u skladu s ACID svojstvima. Ispravnost i izolacija procesa su osigurani primjenom bilo kojega tijeka rada definiranog u prethodnome poglavlju.

U prvome je dijelu ovoga poglavlja objašnjen model podataka, ulazni podaci koji su korišteni za ispitivanje slučajeva korištenja te način njihova učitavanja i prilagodbe razvijenome modelu podataka. Nakon toga je objašnjen postupak definicije tijeka rada u razvijenom WFMS-u te su testirani slučajevi korištenja primjenom različitih tijekova rada kako bi se utvrdile razlike u smislu broja zaključanih objekata i konkurentnosti procesa.

### 5.1. Model podataka

Elementi Petrijevih mreža pomoću kojih se modeliraju procesi pohranjuju se u skladu s konceptualnim modelom podataka prikazanima na slici 4.6. Na temelju je konceptualnoga modela napravljen logički model relacijske baze podataka implementiran u PostgreSQL bazi podataka. U nastavku je prikazan dio logičkoga modela podataka koji prikazuje način na koji su katastarske čestice i međne točke pohranjene i povezane (Slika 5.1). Model se temelji na predloženom modelu integracije WFMS-a u LADM prikazanoga na slici 4.28. Veze (*caseCreated* i *caseDestroyed*) između tablice *wf\_case* i *cp\_parcel* te *cp\_point* nisu prikazane, ali su prikazani strani ključevi u tablicama. Međne se točke i katastarske čestice verzioniraju, a veze su između njih pohranjene eksplicitno u tablici *cp\_parcel\_point* za svaku verziju katastarske čestice posebno.



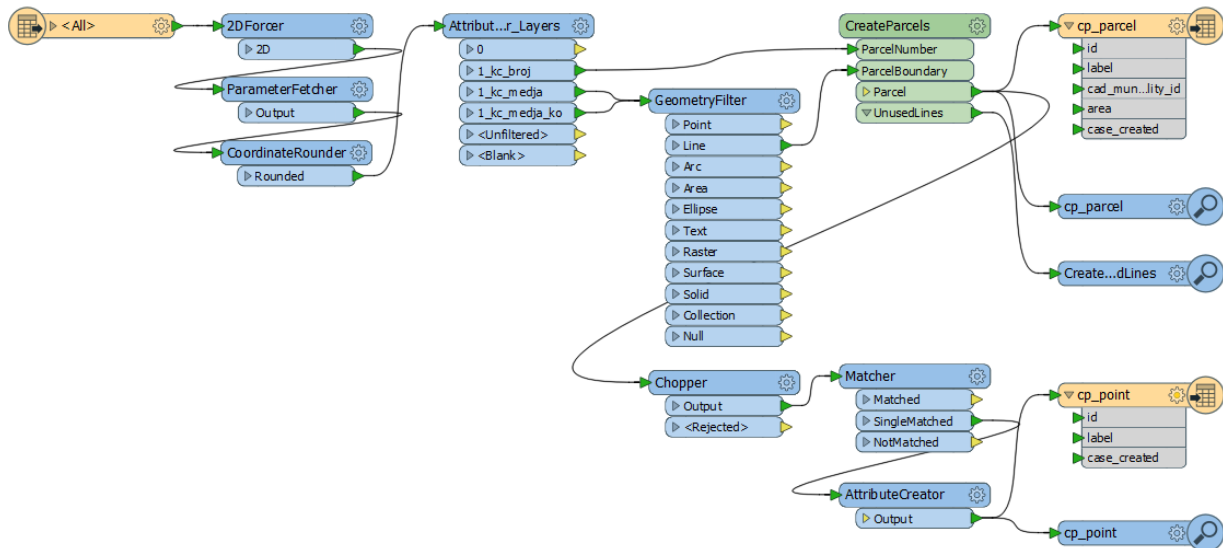
Slika 5.1: Dio logičkoga modela podataka (katastarski podaci)

## 5.2. Izvorni podaci i učitavanje u bazu podataka

Za ispitivanje slučajeva korištenja koriste se podaci o katastarskim česticama preuzeti s DEMLAS spremišta podataka<sup>1</sup>. Preuzeti podaci o katastarskim česticama učitani su u PostgreSQL bazu podataka pomoću programskoga sustava FME, a međne su točke generirane iz geometrije katastarskih čestica (transformer *Chopper*). Pomoću transformera *Matcher* višestruke međne točke generirane iz susjednih poligona objedinjene su kako ne bi bilo duplih međnih točaka (Slika 5.2). Podaci su u službenoj ravninskoj kartografskoj projekciji u RH, HTRS96/TM<sup>2</sup>.

<sup>1</sup>DEMLAS spremište podataka: <https://demlas.geof.unizg.hr/warehouse/portal/>

<sup>2</sup>HTRS96/TM: <http://spatialreference.org/ref/epsg/htrs96-croatia-tm/>



Slika 5.2: FME radna površina za učitavanje podataka u PostgreSQL bazu podataka

Nakon učitavanja katastarskih čestica i međnih točaka kreirane su eksplicitne veze između njih (Ispis koda 5.1). Kreiranjem svake nove verzije katastarske čestice, stvaraju se veze između nove verzije čestice i međnih točaka.

Ispis koda 5.1: Povezivanje međnih točaka s vrhovima poligona

```
INSERT INTO cp_parcel_point (
  SELECT nextval('cp_parcel_point_seq') AS id,
         p.id AS parcel_id, po.id point_id
  FROM (
    SELECT id, (dp).path[2] As index,
           st_x((dp).geom) AS x, st_y((dp).geom) AS y
    FROM (
      SELECT id, ST_DumpPoints(geom) AS dp
      FROM cp_parcel
    ) AS p1
  ) p
  JOIN cp_point po ON p.x=st_x(po.geom) AND p.y=st_y(po.geom)
);
```

### 5.3. Arhitektura testnoga sustava

Web stranica prikazuje sadržaj statičnih datoteka. Web aplikacija prikazuje sadržaj koji se dinamički generira, ovisno o parametrima za prikaz koji su određeni unosom korisnika, pravima pristupa, i tako dalje. Razvojem Interneta i računala, web aplikacije postaju sve

složenije te se od 1-slojnih, 2-slojnih (klijent/server) dolazi do 3-slojnih i višeslojnih web aplikacija. Razlog je podjele na slojeve povećanje sigurnosti, skalabilnosti, fleksibilnosti te jednostavnije upravljanje (Shklar i Rosen, 2009). Arhitektura testnoga sustava temelji se na konceptu troslojne web aplikacije. Najniži se sloj temelji na bazi podataka PostgreSQL s proširenjem za prostorne podatke PostGIS. Na srednjemu se sloju nalazi glavina programske logike napisane u programskome jeziku PHP te GeoServer koji poslužuje prostorne podatke putem WMS i WFS servisa. Najviši (prezentacijski) sloj predstavlja sučelje web aplikacije implementirano pomoću HTML-a, CSS-a te korištenjem JavaScript razvojnih okvira jQuery i OpenLayers. Na slici 5.3 učitani su katastarski podaci prikazani u testnome sustavu.



Slika 5.3: Korišteni katastarski podaci prikazani u testnome sustavu

Klase za upravljanje tijekom rada prikazane na slici 4.6 implementirane su na srednjemu sloju u programskome jeziku PHP. Sustav za upravljanje tijekom rada ne treba biti upoznat s načinom na koji funkcionira pojedini zadatak. Zato je implementiran dodatni modul nazvan *GeoApp* kojim je dodan još jedan sloj apstrakcije web aplikacije. Metode modula za upravljanje tijekom rada pozivaju uvijek iste metode modula *GeoApp* preko naziva zadatka. Metode *GeoApp* modula čitaju metapodatke pozvanoga zadatka pohranjene u JSON formatu, a sadrže informacije o vrsti zadatka, načinu pozivanja, ulaznim i izlaznim vrijednostima. Na slici 5.4 prikazana su dva procesa koji su u različitim fazama. Na slici 5.4a prikazan je proces kojemu je sljedeći zadatak provjeravanje ispravnosti, a na slici 5.4b prikazan je proces koji tek treba dodati katastarske čestice u proces. WFMS-u je poznat status svakoga procesa i nakon što odredi koji je sljedeći zadatak, prosljeđuje upit modulu *GeoApp* o definiciji samoga zadatka.



Postupak					
Dodaj Uredi Ažuriraj Prethodne aktivnosti Sljedeća aktivnosti Provjeri ispravnost					
Id	Postupak	Tijek rada	Korisnik	Početak	Završetak
23	Test 6	Testni tijek rada	svranic	2017-02-22 13:19:08.265988	
4	Case 4	Dioba katastarskih čestica	svranic	2016-10-20 08:00:00	
5	Testni case	Testni tijek rada	svranic		
6	Testni case 2	Testni tijek rada	svranic		
25	Test 2 - izolirani	Slučaj 1 - bez kontroliranja konkurentnih procesa	svranic	2017-03-02 05:27:16.337581	2017-03-02 05:43:59
1	Case 1	Dioba katastarskih čestica	svranic	2016-09-30 08:00:00	2017-01-01 08:00:00
27	Test H2	Slučaj 1 - bez kontroliranja konkurentnih procesa	svranic	2017-03-02 08:12:15.651536	

(a) Proces A

Postupak					
Dodaj Uredi Ažuriraj Prethodne aktivnosti Sljedeća aktivnosti Dodaj/izbriši parcelu					
Id	Postupak	Tijek rada	Korisnik	Početak	Završetak
23	Test 6	Testni tijek rada	svranic	2017-02-22 13:19:08.265988	
4	Case 4	Dioba katastarskih čestica	svranic	2016-10-20 08:00:00	
5	Testni case	Testni tijek rada	svranic		
6	Testni case 2	Testni tijek rada	svranic		
25	Test 2 - izolirani	Slučaj 1 - bez kontroliranja konkurentnih procesa	svranic	2017-03-02 05:27:16.337581	2017-03-02 05:43:59
1	Case 1	Dioba katastarskih čestica	svranic	2016-09-30 08:00:00	2017-01-01 08:00:00
27	Test H2	Slučaj 1 - bez kontroliranja konkurentnih procesa	svranic	2017-03-02 08:12:15.651536	

(b) Proces B

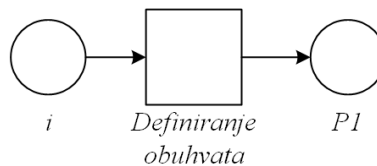
Slika 5.4: Dohvaćanje dostupnih akcija u proces

## 5.4. Definicija tijeka rada u sustavu

Konceptualni model podataka prikazan na slici 4.6 omogućava pohranjivanje elemenata WF mreža. Pohranjivanjem elemenata WF mreža definiran je redoslijed i način pozivanja prijelaza, ali semantika je samih prijelaza definirana metapodacima modula *GeoApp*. Na taj način WFMS nije upoznat s načinom funkcioniranja prijelaza čime se zadržava općenitost WFMS-a koji se onda može primijeniti u različite svrhe. U nastavku su objašnjeni pojedini elementi tijekova rada definiranih u ovom radu.

### 5.4.1. Pesimistični tijek rada

Pesimistični tijek rada osigurava izolaciju procesa zaključavanjem svih formalno i tehnički zahvaćenih katastarskih čestica (Slika 4.15). *Definiranje obuhvata* prvi je prijelaz koji omogućava dodavanje katastarskih čestica u proces (Slika 5.5). Prilikom dodavanja svake nove katastarske čestice ažurira se formalno i tehničko zahvaćeno područje, ovisno o tome je li dodana formalno ili tehnički zahvaćena čestica.



Slika 5.5: Prijelaz *Definiranje obuhvata*

Prijelaz je s početkom procesa automatski omogućen (nema preduvjeta) jer predstavlja prvi zadatak u procesu. Prijelaz je uspješno završen ako je dodana barem jedna formalno zahvaćena čestica. Dok se taj uvjet ne ispuni, tijek rada ne može napredovati do mjesta *P1*. Navedeni uvjet predstavlja postuvjet prijelaza *Definiranje obuhvata* te se povezuje s lukom koji spaja taj prijelaz s mjestom *P1*. Slika 5.6 prikazuje kako se definira mjesto i prijelaz. Kod definicije su mjesta najbitniji atributi naziv i vrsta mjesta. Kod definicije se

prijelaza unosi naziv i okidač prijelaza. Ako okidač prijelaza nije korisnička radnja, može se definirati i istek vremena nakon omogućavanja prijelaza nakon kojega će se zadatak pokrenuti.

(a) Definicija mjesta

(b) Definicija prijelaza

Slika 5.6: Definicija elemenata tijeka rada

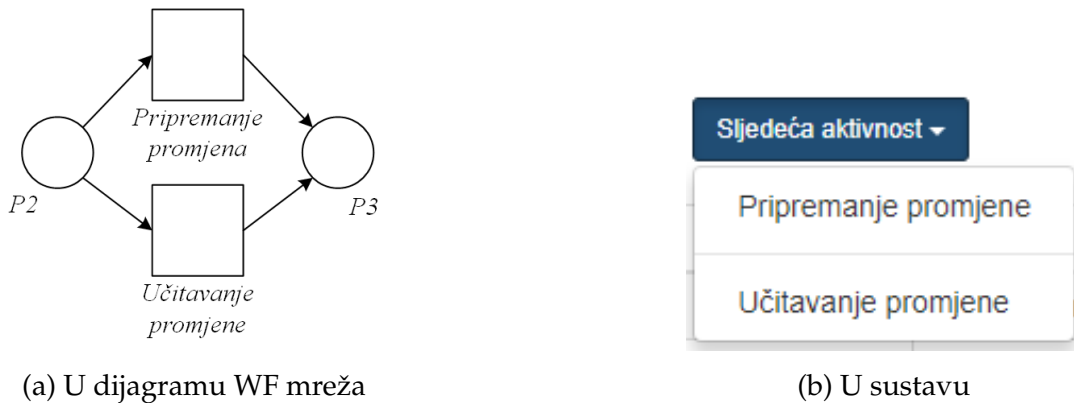
Elementi prikazani na slici 5.6 nisu povezani, stoga se povezuju lukovima kao što je prikazano na slici 5.7. Definicija luka sadrži informaciju o smjeru luka, vrsti te mjesto i prijelaz koje povezuje. Definiira se i veza luka na tijek rada.

(a) Definicija luka od mjesta do prijelaza

(b) Definicija prijelaza do mjesta

Slika 5.7: Definicija sekvencijalnih lukova

Slika 5.8 prikazuje uzorak implicitnog OR dijeljenja koji omogućava da tijekom rada, ovisno o odabiru korisnika ili predefiniranim uvjetima, izvršava različite prijelaze. Prikaz OR dijeljenja u sustavu je jednostavan. Korisnik ima pravo odabrati sljedeću aktivnost (Slika 5.8b). Nakon što odabere sljedeću aktivnost, omogućava se odabrani prijelaz i više nije moguće odabrati drugu aktivnost osim u slučaju povratka na prethodno mjesto i kompenziranjem učinjenih promjena.



Slika 5.8: Prikaz uzorka OR dijeljenje

Slika 5.8 prikazuje definiciju dva luka s mjesta *P2* do prijelaza *Pripremanje promjene* i *Učitavanje promjene*.

(a) Luk od mjesta P2 do prijelaza za pripremanje promjene

(b) Luk od mjesta P2 do prijelaza za učitavanje promjene

Slika 5.9: Definicija lukova za modeliranje OR dijeljenja

Slika 5.10 prikazuje definiciju spajanja lukova u jedinstveni tijek rada.

(a) Luk od prijelaza za pripremanje promjene do mjesta P3

(b) Luk od prijelaza za učitavanje promjene do mjesta P3

Slika 5.10: Definicija lukova za modeliranje OR spajanja

Na početku poglavlja je objašnjen prijelaz *Definiranje obuhvata*, a u nastavku način funkcioniranja svakoga pojedinog prijelaza sa slike 4.15:

- Prijelaz *Zaključavanje čestica* omogućava zaključavanje formalno i tehnički zahvaćenih katastarskih čestica. Preduvjet je prijelaza da nijedna od zahvaćenih katastarskih čestica nije zaključana u drugome procesu. Ako je bilo koja od zahvaćenih čestica zaključana u drugome procesu, zaključavanje se odbija i tijek rada ne može nastaviti dalje dok se ne promijeni obuhvat procesa ili dok se ne otključaju potrebne čestice. Postuvjet je prijelaza da su sve zahvaćene katastarske čestice zaključane. Nakon što se zaključaju sve zahvaćene katastarske čestice, u vrijednost se atributa *locked* u tablici *wf\_case* upisuje trenutno vrijeme. Od toga trenutka drugi procesi ne mogu raditi izmjene niti zaključavati te katastarske čestice. Time je osigurana izolacija procesa. Nakon toga tijek rada može napredovati dalje. Ovisno o odabranome načinu pripremanja promjene sljedeći prijelaz može biti *Pripremanje promjena* ili *Učitavanje promjena*.
- Prijelaz *Pripremanje promjena* omogućava korisniku da samostalno napravi izmjene na katastarskim česticama. Postuvjet je ovoga prijelaza kao i sljedećega da je napravljena barem jedna izmjena na katastarskim česticama.
- Prijelaz *Učitavanje promjena* omogućava učitavanje cjelovite promjene pripremljene u računalnome programu izvan WFMS sustava.
- Prijelaz *Provjera ispravnosti* omogućava provjeru ispravnosti prema uvjetima integriteta definiranim u (Vranić i dr., 2015). Ukoliko svi testovi vrte istinitu logičku vrijednost, geometrijsko-topološka ispravnost procesa je osigurana.

- Prijelaz *Promjena neispravna* predstavlja alternativni tijek rada koji nastupa ukoliko provjera topološke i geometrijske ispravnosti nije zadovoljena.
- Prijelaz *Promjena ispravna* označava čitavi proces ispravan i popunjava vrijednost atributa *checked* u tablici *wf\_case*.
- Prijelaz *Izrada certifikata* nije posebno razrađen u ovom radu, ali predstavlja nebitan zadatak čije neuspješno obavljanje ne uzrokuje otkazivanje procesa. Nakon što zadnji prijelaz završi s izvođenjem, vrijednost se atributa *executed* u tablici *wf\_case* pohranjuje i od toga su trenutka sve promjene vidljive ostalim procesima, a zahvaćene se katastarske čestice otključavaju.
- Prijelaz *Otkazivanje procesa* predstavlja još jedan alternativni tijek rada koji omogućava oporavak-natrag. Atributu *cancelled* dodjeljuje se vrijednost koja označava trenutak kada je otkazivanje procesa započelo.
- Prijelaz *Poništavanje promjena* poziva sve kompenzirajuće prijelaze koji su povezani s prijelazima refleksivnom vezom. Kompenzirajući prijelazi pozivaju se redoslijedom suprotnim od onoga kako su pozivani pripadajući prijelazi.
- Prijelaz *Izrada izvještaja* također je nebitan prijelaz koji omogućava izradu izvještaja o učinjenim promjenama na podacima koje su u određenome trenutku otkazane.

#### 5.4.2. Optimistični tijek rada

Optimistični je tijek rada prikazan na slici 4.18. Dio je prijelaza objašnjen u prethodnom odjeljku budući da se koristi jedan dio prijelaza kod pesimističnog i optimističnoga tijeka rada. U nastavku su objašnjeni prijelazi specifični za optimistični tijek rada:

- Prijelaz *Provjera konkurentnih procesa* dohvaća sve aktivne procese i ispituje prostorni odnos zahvaćenih područja. Ukoliko zahvaćena područja imaju prostorni odnos različit od DISJOINT ili TOUCH, ispituje se u kojoj je fazi pojedini proces kako bi se utvrdilo jesu li procesi konkurentni. Ukoliko nema konkurentnih procesa, proces se nastavlja dalje prema kraju i označava promjenu kao ispravnu (odnosno popunjava se vrijednost atributa *checked*). U suprotnome se pokušavaju dohvatiti najnovije verzije katastarskih čestica od konkurentnih procesa koji su provedeni za vrijeme pripremanja testiranoga procesa.
- Prijelaz *Dohvaćanje najnovijih verzija* omogućava preuzimanje promjena napravljenih u konkurentnome provedenom procesu. Ukoliko se radi o konfliktnome konkurentnom procesu u kojemu su zahvaćene čestice promijenile identitet, promjene se ne mogu preuzeti, jer se mijenja definicija zahvaćenoga područja i proces se otkazuje.
- Prijelaz *Integracija promjena* omogućava integraciju promjena iz preuzetih verzija iz konkurentnog procesa u stare verzije katastarskih čestica. Nakon toga se tijek rada vraća u mjesto *P1* i promjene se dorađuju kako bi bile u skladu s novim verzijama katastarskih čestica.

### 5.4.3. Altruistični tijek rada

Altruistični tijek rada omogućava definiranje uvjeta integriteta na nižim topološkim strukturama (čvorovi i bridovi) što rezultira manjim skupom zaključanih podataka. Prepoznavanjem je promjena na razini točke moguće prepoznati vrstu i lokaciju promjene što omogućava uvođenje operacije doniranja koja omogućava razmjenu promjena između procesa s ciljem povećanja konkurentnosti procesa. U nastavku su definirani prijelazi specifični za altruistični tijek rada:

- Prijelaz *Zaključavanje točaka* omogućava zaključavanje svih točaka vezanih uz formalno zahvaćene katastarske čestice kako se ne bi mogli dogoditi konfliktni konkurentni procesi. Altruistični tijek rada ne dopušta konfliktno konkurentne procese jer se zaključavaju sve točke vezane uz formalno zahvaćene katastarske čestice te ne postoji mogućnost da se dogode dva procesa čija se formalno zahvaćena područja dodiruju.
- Prijelaz *Zatraži doniranje* omogućava od konkurentnoga procesa da zatraži pravo rada izmjena na konkurentnim katastarskim česticama.
- Prijelaz *Provjera ispravnosti* omogućava testiranje ispravnosti ravinske particije pomoću čvorova i bridova stvorenih iz poligona katastarskih čestica. Koncept definiran u (Matijević i dr., 2008) proširen je kako bi se omogućilo pohranjivanje informacije o vrsti promjene na bridu što omogućava uvođenje operacije doniranja kako je konceptijski opisano u odjeljku 4.5. Ravninska particija je ispravna ako svaki polubrid ima komplement s komplementarnim statusom.
- Prijelaz *Provjera doniranih vrijednosti* osigurava da se proces koji donira pravo izmjene drugome konkurentnom procesu ne može provesti dok se konkurentni proces ne provede i time otključa donirane vrijednosti. Ukoliko konkurentni proces koji je zatražio pravo izmjene na zajedničkim česticama nije proveden, ne može završiti niti matični proces. Na taj se način osigurava izolacija procesa.

## 5.5. Ispitivanje tijekova rada na primjerima

U poglavlju 4 definirani su različiti tijekovi rada koji osiguravaju ispravnost procesa nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima, ali omogućavaju različite razine konkurentnosti procesa. U nastavku su ti tijekovi rada upotrijebljeni za ispitivanje različitih primjera na testnim podacima kako bi se provjerila njihova ispravnost i primjenjivost.

### 5.5.1. Konkurentni procesi sa zajedničkom susjednom česticom

U ovom primjeru dva konkurentna procesa ne pokušavaju napraviti izmjene na istim katastarskim česticama, već zbog očuvanja ispravnosti prostorne sastavnice rade izmjene na susjednome poligonu. Na tome će se primjeru u nastavku ispitati definirani tijekovi rada

kako bi se utvrdile razlike između njih u smislu ograničavanja pristupa podacima i konkurentnosti procesa. Isto tako, kako bi se ispitala ispravnost korištenih uvjeta integriteta u drugome će se primjeru ispitivati ispravnost procesa koji sadrži topološku grešku.

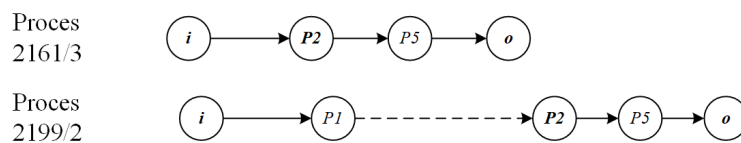
### Primjer ispravnoga slučaja

Slika 5.11 prikazuje dva konkurentna procesa koji imaju zajedničku susjednu katastarsku česticu 3231. Oba procesa dodaju čvor na brid koji dijele s katastarskom česticom 3231.



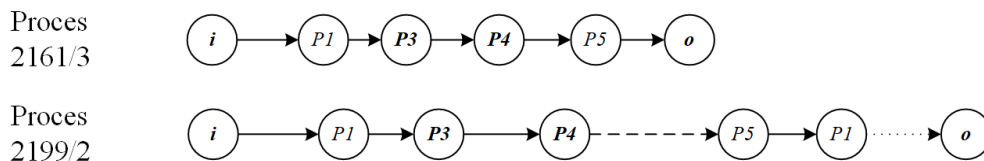
Slika 5.11: Konkurentni procesi sa zajedničkom susjednom česticom

Pesimistični tijek rada zahtijeva da se zaključaju sve formalno i tehnički zahvaćene katastarske čestice kako bi se osigurala izolacija procesa. U primjeru prikazanome na slici 5.11 proces je na čestici 2161/3 prvi je zaključao formalno zahvaćenu česticu 2161/3 i tehnički zahvaćenu česticu 3231 koju je potrebno uključiti u proces zbog očuvanja ispravnosti ravninske particije. Proces na čestici 2199/2 ne može nastaviti s pripremanjem promjene jer ne može zaključati katastarsku česticu 3231 koja je već zaključana. Ispravnost se prostorne sastavnice u procesu provjerava u prijelazu *Provjera ispravnosti* prema uvjetima integriteta za varijabilno formalno zahvaćeno područje definiranim u (Vranić i dr., 2015). Kako bi se osigurala izolacija, proces je na čestici 2199/2 prisiljen čekati dok se proces na 2161/3 ne provede (Slika 5.12).



Slika 5.12: Vremenski raspored provedbe procesa prema pesimističnome tijeku rada

Optimistični tijek rada dopušta pripremanje promjena istovremeno u više procesa. Pripremanje promjena kod procesa u sustavima za upravljanje zemljišta može potrajati duže vrijeme. Zato se validacija odgađa da se dogodi što kasnije (prije provedbe procesa). Primjenom se FOCC protokola proces validira s drugim aktivnim konkurentnim procesima. Ukoliko se ne radi o konfliktnim konkurentnim procesima, kod neuspješne se validacije procesa prvo pokušavaju integrirati promjene iz konkurentnoga procesa. Nakon uspješne integracije slijedi ponovno pripremanje promjene. Budući da se zahvaćeno područje procesa ne može mijenjati nakon što je jednom definirano, kod konfliktnih se konkurentnih procesa gdje čestice mijenjaju identitet promjene ne mogu integrirati već se proces otkazuje. Na slici 5.11 procesi se na katastarskim česticama 2161/3 i 2199/2 mogu pripremati istovremeno. Prema vremenskom rasporedu konkurentnih procesa na slici 5.13 proces je na čestici 2161/3 prvi došao u mjesto *P3* i ima pravo prvenstva kod provedbe. Proces 2199/2 provjerava konkurentne procese kasnije i validacija je procesa neuspješna. Tijek rada se vraća alternativnim putem na mjesto *P1* pri čemu se mogu integrirati promjene iz procesa na čestici 2161/3 jer se ne radi o konfliktnim konkurentnim procesima, odnosno čestica 3231 zadržala je identitet.



Slika 5.13: Vremenski raspored provedbe procesa prema optimističnome tijeku rada

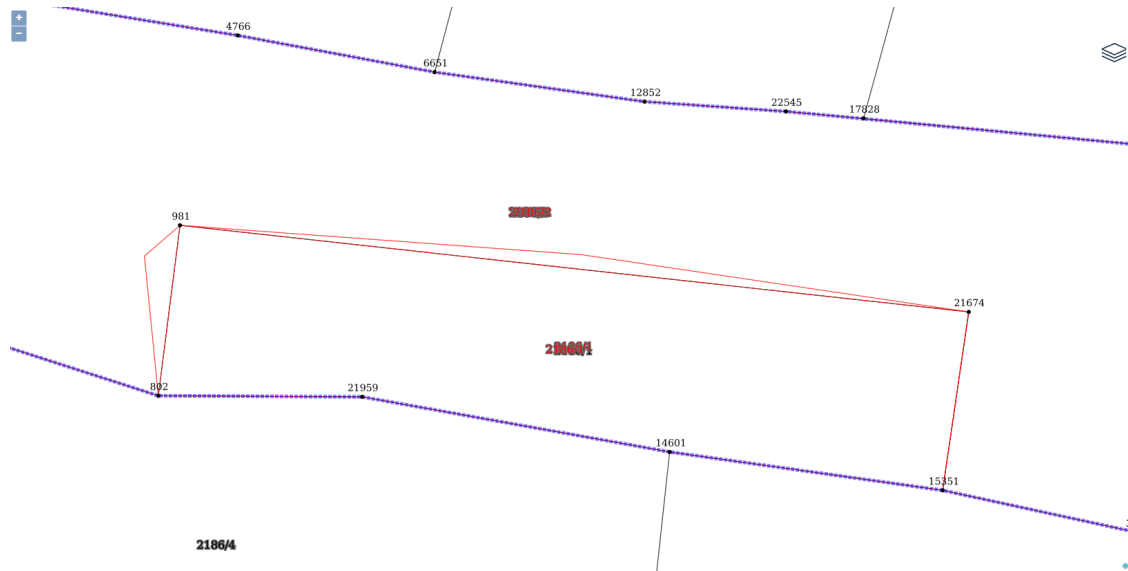
Altruistični tijek rada izolaciju procesa osigurava zaključavanjem svih točaka vezanih uz formalno zahvaćene katastarske čestice. Katastarske se čestice ne zaključavaju iako se definiraju formalno i tehnički zahvaćene katastarske čestice. Primjenom altruističnog tijeka rada na situaciju prikazanu na slici 5.11 moguće je povećati konkurentnost procesa. Proces je na katastarskoj čestici 2161/3 započeo ranije i pripremio promjenu. Proces na katastarskoj čestici 2199/2 započeo je kasnije i traži od procesa na čestici 2161/3 pravo izmjene na čestici 3231. Nakon što dobije pravo izmjene, proces na čestici 2199/2 zaključava česticu 3231 i dodaje čvor na tu česticu te nastavlja do provedbe. Prilikom provedbe otključava česticu 3231 i proces na 2161/3 može nakon toga nastaviti prema provedbi.

Kod altruističnoga tijeka rada ispravnost se ravninske particije obavlja primjenom modificiranoga koncepta iz (Matijević i dr., 2008) kao što je objašnjeno u odjeljku 4.5, dok se geometrijska ispravnost prostorne sastavnice ispituje uvjetima integriteta iz (Vranić i dr., 2015). U primjeru sa slike 5.11 oba konkurentna procesa rade izmjene koje su u skladu s uvjetima integriteta te proces može nastaviti prema provedbi. U sljedećem je odjeljku prikazan primjer na istim česticama, ali novo stanje sadrži namjerno uzrokovanu pogrešku.





Kod optimističnoga je tijekom rada istovremeno pripremanje promjena na konkurentnim česticama moguće. Ako susjedne čestice ne mijenjaju identitet kao što je slučaj na slici 5.15, moguće je da proces na čestici 2186/2 nakon provedbe procesa na čestici 2186/1 integrira promjene i nakon toga nastavi prema provedbi.



Slika 5.15: Konkurentni procesi sa zajedničkom susjednom česticom

Kod altruističnoga tijekom rada nije moguće istovremeno pripremanje promjena na česticama 2186/1 i 2186/2 jer proces koji prvi zaključa sve čvorove vezane uz formalno zadržanu katastarsku česticu može nastaviti dalje prema tijeku rada. U primjeru je sa slike 5.15 proces na čestici 2186/1 prvi zaključao čvorove vezane uz formalno dodanu česticu 2186/1. Proces na čestici 2186/2 ne može zaključati dio čvorova na zajedničkim bridovima s česticom 2186/1 i zato ne može nastaviti dalje dok se ne provede proces na čestici 2186/1.

## 5.6. Rekapitulacija

Na primjerima su prikazane međusobne razlike definiranih tijekova rada u smislu ograničavanja pristupa podacima te razini konkurentnosti koju pružaju. Pesimistični tijek rada predstavlja najstroži pristup te izolaciju konkurentnih procesa osigurava zaključavanjem svih formalno i tehnički zahvaćenih katastarskih čestica. Optimistični tijek rada ne zaključava katastarske čestice što omogućava istovremenu pripremu promjena na katastarskim česticama. Validacija ispravnosti i izolacije se odgađa tako da se obavlja prije sameprovedbe procesa. Integracija je promjena iz konkurentnih procesa moguća ako zahvaćene katastarske čestice nisu promijenile identitet. Ako je identitet zahvaćenih čestica promijenjen, proces se mora otkazati, jer se promijenilo zahvaćeno područje zato što je čestica koja je izvorno zahvaćena procesom, u međuvremenu nestala. Altruistični tijek rada omogućava rad na razini točke što omogućava preciznije određivanje vrste i lokacije promjene

koja se dogodila kao i zaključavanje manjega skupa podataka. Kod konkurentnih procesa altruistični tijek rada može povećati konkurentnost procesa, dok kod konfliktnih procesa ima isti učinak kao i pesimistični tijek rada.

## DIO III:

# REZULTATI ISTRAŽIVANJA

*U ovome se dijelu rada daje koncizni pregled postignutih rezultata, zaključci istraživanja te mogući smjerovi i područja budućih istraživanja.*

## 6. Zaključak

Cilj je ovoga doktorskog rada bila analiza postojećih pristupa te razvoj koncepta koji će omogućiti ubrzavanje i automatizaciju procesa te povećanje učinkovitosti upravljanja procesima nad prostornom sastavnicom katastarskih čestica. Budući da transakcijski modeli ne pružaju dovoljnu razinu fleksibilnosti kod upravljanja procesima nad prostornim podacima u heterogenim informatičkim okruženjima, istražena je primjenjivost sustava za upravljanje tijekom rada. Bitniji su nedostaci primjene općenitoga WFMS-a na rad s katastarskim podacima nedovoljna podrška za osiguravanje ispravnosti i mogućnosti oporavka procesa u slučaju pogreške ili pada WFMS-a. U ovome su se radu ti nedostaci uklonili primjenom transakcijskoga WFMS-a i primjenom ACID svojstava na procese. Isto tako, koncept WFMS-a sam po sebi podržava široki spektar informatičkih tehnologija koje povećavaju fleksibilnost, što može dovesti i do povećanja učinkovitosti.

Temeljem provedenog istraživanja u okviru izrade ovog rada izvedeni su sljedeći zaključci te odgovori na postavljene hipoteze:

- *Konzistentnost prostorne sastavnice moguće je osigurati primjenom koncepta transakcijskoga sustava za upravljanje tijekom rada.*
  - Primjenom ACID svojstava na razini procesa osigurano je da proces predstavlja transformaciju podataka iz jednoga konzistentnog stanja u drugo.
  - Ispravnost prostorne sastavnice osigurana je integracijom uvjeta integriteta definiranih u (Vranić i dr., 2015) u WFMS.
  - Na temelju konceptualnoga modela transakcijskoga WFMS-a definirana su tri tijeka rada: pesimistični, optimistični i altruistični koji osiguravaju konzistentnost prostorne sastavnice katastarskih čestica kod konkurentnoga pristupa katastarskim česticama osiguravanjem serijalizabilnosti procesa.
  - Na testnim je primjerima dokazana ispravnost i primjenjivost razvijenoga konceptualnog modela korištenjem testnoga sustava. Primjeri s namjerno uzrokovanim pogreškama očekivano nisu zadovoljili uvjete integriteta i proces nije mogao nastaviti dalje prema provedbi dok pogreške nisu bile ispravljene.
- *Primjenom koncepta transakcijskoga sustava za upravljanje tijekom rada moguće je povećati učinkovitost te ubrzati i automatizirati kontrolu ispravnosti transakcija na prostornoj sastavnici katastarskih čestica.*

- U ovome su radu za kontroliranje ispravnosti korišteni uvjeti integriteta definirani u (Vranić i dr., 2015) koji su ugrađeni u prijelaz. Na taj je način omogućena automatska kontrola ispravnosti procesa. Dodatni se uvjeti vezani uz druge skupove podataka ili uvjeti koje postavljaju druge institucije mogu također uključiti u tijek rada u isti ili neki drugi prijelaz.
- WFMS izvorno pruža vrlo dobru podršku za upravljanje korisnicima i njihovim pravima unutar sustava za upravljanje zemljištem. Svaki se zadatak može dodijeliti različitom resursu (dioniku, web servisu) čime se resursi mogu raste-retiti. Isto se tako dionicima izvan sustava za upravljanje zemljištem mogu do-dijeliti prava za učitavanje promjena pripremljenih izvan sustava za upravljanje zemljištem i provjera ispravnosti učitanih promjena. Na taj način dionici izvan sustava za upravljanje zemljištem sami mogu provjeravati ispravnost promjena koje pripremaju, odnosno ne trebaju čekati službenu provjeru od dionika unu-tar sustava za upravljanje zemljištem. Raspodjelom se zadataka na različite ko-risnike rasterećuje sustav za upravljanje zemljištem te se može smanjiti trajanje procesa.
- Definicijom su različitih tijekova rada, odnosno protokola omogućene različite razine konkurentnosti procesa. Primjenom je prikladnoga tijeka rada na odre-đene slučajeve (vrste procesa) moguće povećati učinkovitost (konkurentnost) procesa.
- Uvođenjem međnih točaka u model podataka prostorne sastavnice katastar-skih čestica omogućeno je preciznije određivanje vrste i položaja promjene koja se dogodila na poligonu koji predstavlja katastarsku česticu, a primjenom je altruističnoga tijeka rada moguće povećati konkurentnost procesa.
- Ubrzavanjem i automatizacijom kontrole ispravnosti prostorne sastavnice sma-njeno je vrijeme trajanja procesa, a primjenom razvijenih tijekova rada moguće je smanjiti vrijeme čekanja jednoga procesa na podatke koji su zaključani u dru-gome procesu. Na ovaj je način povećana učinkovitost WFMS-a.

## 6.1. Izvorni znanstveni doprinos

U ovom je radu razvijen konceptualni model transakcijskoga WFMS-a koji omogućava po-hranjivanje elemenata tijeka rada za modeliranje dinamičke sastavnice (to jest procesa) u sustavima za upravljanje zemljištem. Taj je model dodatno specijaliziran kako bi se osigu-rala ispravnost procesa nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Na temelju definiranoga konceptualnog modela i primjenom koncepta tran-sakcijskoga WFMS-a definirana su tri tijeka rada: pesimistični, optimistični i altruistični koji osiguravaju konzistentnost prostorne sastavnice katastarskih čestica i omogućavaju različite razine konkurentnosti (učinovitosti) procesa. Dok pesimistični tijek rada isprav-nost i izolaciju osigurava zaključavanjem svih katastarskih čestica unutar zahvaćenoga

područja procesa, optimistični i altruistični mogu povećati učinkovitost procesa omogućavanjem istovremene pripreme promjena na istim katastarskim česticama u više procesa. Integracijom je WFMS-a u LADM dodana podrška za modeliranje procesa nad LADM klasama te je dokazana primjenjivost konceptualnoga modela transakcijskoga WFMS-a na LADM.

Implementacijom je testnoga sustava provjerena ispravnost i primjenjivost postavljениh koncepata te je dokazno kako se može osigurati konzistentnost te automatizirati kontrola ispravnosti prostorne sastavnice katastarskih čestica pomoću WFMS-a. Primjenom se WFMS-a na upravljanje procesa nad prostornom sastavnicom katastarskih čestica povećava učinkovitost korištenjem dostupnih tehnologija (poput web servisa, integracije s računalnim programima) koje olakšavaju suradnju različitih dionika s različitim pravima na procesima, komunikaciju između WFMS-a, učitavanje promjena pripremljenih izvan sustava za upravljanje zemljištem te smanjivanje vremena čekanja jednoga procesa na podatke koji su zaključani u drugome procesu.

## 6.2. Daljnja istraživanja

U ovome su radu u obzir uzeti procesi nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima. Upravljanje se konkurentnim procesima obavlja na razini objekta katastarske čestice, ali različite strukture prostorne sastavnice mogu postavljati specifične uvjete integriteta te su potrebna dodatna istraživanja u tome smjeru.

ACID svojstva primijenjena su na procese kako bi se osigurala ispravnost prostorne sastavnice katastarskih čestica, ali uvođenjem je drugih skupova podataka i definicijom vrsta procesa koji rade promjene na više skupova podataka moguće da će ACID svojstva biti ograničavajuća. Primjerice, opuštanje atomičnosti procesa kod upisa građevina: upis građevina koje zadovoljavaju određene formalne i pravne uvjete, a ostale se odbacuju, odnosno ne upisuju.

U ovome je radu ponuđen prijedlog integracije koncepta transakcijskoga WFMS-a u LADM. Budući da je konceptualni model detaljno razrađen samo za procese nad katastarskim česticama čija se prostorna sastavnica temelji na poligonima, potrebno je dodatno ispitati integraciju koncepta WFMS-a u LADM u smislu podrške za procese nad ostalim LADM klasama (podrška za procese nad pojedinačnim klasama, ali i za procese koji uključuju više klasa) te podrške za različite strukture prostorne sastavnice.

# Literatura

- Ahn, G.-J., Sandhu, R., Kang, M., i Park, J. (2000). Injecting RBAC to secure a web-based workflow system. In *Proceedings of the fifth ACM workshop on Role-based access control*, pages 1–10. ACM.
- Al-Taha, K. K. (1992). *Temporal reasoning in cadastral systems*. PhD thesis.
- Alonso, G., Agrawal, D., El Abbadi, A., i Mohan, C. (1997). Functionality and limitations of current workflow management systems. *IEEE expert*, 12(5):105–111.
- Alonso, G., Agrawal, D., Kamath, M., Günthör, R., Mohan, C., i dr. (1996). Advanced transaction models in workflow contexts. In *12th International Conference on Data Engineering*, pages 574–581. IEEE.
- Alonso, G., Günthör, R., Kamath, M., Agrawal, D., El Abbadi, A., i Mohan, C. (1995). Exotica/FMDC: Handling disconnected clients in a workflow management system. In *Third International Conference on Cooperative Information Systems (CoopIS-95)*.
- Alonso, G. i Hagen, C. (1997). Geo-Opera: Workflow concepts for spatial processes. In *5th International Symposium on Spatial Databases (SSD '97)*, pages 238–258, Berlin, Germany.
- Anand, A. (2017). Blockchain and land administration: Challenges and opportunities. In *Proceedings of the FIG Working Week 2017: Surveying the world of tomorrow—From digitalisation to augmented reality*, Helsinki, Finland. FIG.
- Anand, A., McKibbin, M., i Pichel, F. (2016). Colored coins: Bitcoin, blockchain, and land administration. In *Annual World Bank Conference on Land and Poverty*.
- Anić, V. (2006). *Veliki rječnik hrvatskoga jezika*. Novi Liber.
- Bancilhon, F., Kim, W., i Korth, H. F. (1985). *A model of CAD transactions*. University of Texas at Austin, Department of Computer Sciences.
- Barga, R. i Pu, C. (1995). A practical and modular implementation of extended transaction models. In *Proceedings of the 21st VLDB Conference*.
- Barghouti, N. S. i Kaiser, G. E. (1991). Concurrency control in advanced database applications. *ACM Computing Surveys (CSUR)*, 23(3):269–317.



- Becker, J. i zur Mühlen, M. (1999). Towards a classification framework for application granularity in workflow management systems. In *International Conference on Advanced Information Systems Engineering*, pages 411–416. Springer.
- Birkmeier, D., Kloeckner, S., i Overhage, S. (2010). An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users. In *ECIS*, volume 2010, page 2.
- Börger, E. (2012). Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software and Systems Modeling*, 11(3):305–318.
- Breitbart, Y., Deacon, A., Schek, H. J., Sheth, A., i Weikum, G. (1993). Merging application-centric and data-centric approaches to support transaction-oriented multi-system workflows. *ACM Sigmod Record*, 22(3):23–30.
- Chai, L., Zhang, L., Yu, G., Qu, Y., i Du, K. (2008). Workflow-based system design for the online inspection of spatial data. In *Geoinformatics 2008 and Joint Conference on GIS and Built environment: Advanced Spatial Data Models and Analyses*, Proceedings of SPIE - the International Society for Optical Engineering, pages 71460G–1–71460G–9. International Society for Optics and Photonics.
- Chinosi, M. i Trombetta, A. (2012). BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.
- Choi, J. O., Shin, Y. S., i Hong, B. H. (1999). Update propagation of replicated data in distributed spatial databases. In *Database and Expert Systems Applications*, pages 952–963. Springer.
- Chrysanthis, P. K. i Ramamritham, K. (1992). ACTA: The SAGA Continues. In Elmagarmid, A. K., editor, *Database Transaction Models For Advanced Applications*, chapter 10, pages 349–397. Morgan Kaufmann.
- Chrysanthis, P. K. i Ramamritham, K. (1994). Synthesis of extended transaction models using ACTA. *ACM Transactions on Database Systems*, 19(3):450–491.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.
- Daniilidis, K. (2006). GIS Cadastre Applications without GIS Software. In *5th FIG Regional Conference: Promoting Land Administration and Good Governance*, pages 1–13, Accra, Ghana. FIG.
- Dayal, U., Hsu, M., i Ladin, R. (1991). A transactional model for long-running activities. In *VLDB*, volume 1991, pages 113–122.

- Decker, G., Dijkman, R., Dumas, M., i García-Bañuelos, L. (2008). Transforming BPMN diagrams into YAWL nets. In *BPM*, pages 386–389. Springer.
- Dehnert, J. (2001). Four systematic steps towards sound business process models. In Weber, H., Ehrig, H., i Reisig, W., editors, *International Colloquium on Petri Net Technologies for Modelling Communication Based System*. Fraunhofer Gesellschaft ISST.
- Derks, W., Dehnert, J., Grefen, P., i Jonker, W. (2001). Customized atomicity specification for transactional workflows. In *Cooperative Database Systems for Advanced Applications, 2001. CODAS 2001. The Proceedings of the Third International Symposium on*, pages 140–147. IEEE.
- Du, Y. i Cheng, P. (2017). LWGeoWfMS: A Lightweight Geo-Workflow Management System. In *IOP Conference Series: Earth and Environmental Science*, volume 59, page 012070, Guangzhou, China. IOP Publishing.
- Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., i dr. (2013). *Fundamentals of business process management*, volume 1. Springer.
- Eder, J. i Liebhart, W. (1994). A Transaction-Oriented Workflow Activity Model. In *Proceedings of the 9th International Symposium on Computer and Information Sciences*.
- Elmagarmid, A. K. (1991). Transaction models for advanced database applications. Technical Report 871, Purdue University.
- Enemark, S. i McLaren, R. (2017). Global campaign to eradicate insecurity of tenure by 2030. In *2017 World Bank Conference on Land and Poverty*. World Bank Publications.
- Evans, E. (2009a). Nosql 2009. may 2009. – blog post of 2009-05-12. [http://blog.sym-link.com/2009/05/12/nosql\\_2009.html](http://blog.sym-link.com/2009/05/12/nosql_2009.html). Pristupljeno: 2017-08-08.
- Evans, E. (2009b). Nosql: What’s in a name? october 2009. – blog post of 2009-10-30. [http://blog.sym-link.com/2009/10/30/nosql\\_whats\\_in\\_a\\_name.html](http://blog.sym-link.com/2009/10/30/nosql_whats_in_a_name.html). Pristupljeno: 2017-08-08.
- Fekete, A., Greenfield, P., Kuo, D., i Jang, J. (2003). Transactions in loosely coupled distributed systems. In *14th Australian database conference*, volume 17, pages 7–12. Australian Computer Society, Inc.
- Ferlan, M., Šumrada, R., i Mattsson, H. (2007). Modelling property transactions. In Zevenbergen, J., Frank, A., i Stubkjær, editors, *Real property transactions: Procedures, transaction costs and models*, chapter 2, pages 27–80. IOS Press.
- Ferraiolo, D., Cugini, J., i Kuhn, D. R. (1995). Role-based access control (rbac): Features and motivations. In *Proceedings of 11th annual computer security application conference*, pages 241–48.

- Fridgen, G., Radszuwill, S., Urbach, N., i Utz, L. (2018). Cross-Organizational Workflow Management Using Blockchain Technology—Towards Applicability, Auditability, and Automation. In *Proceedings of the 51st Annual Hawaii International Conference on System Sciences*.
- Galić, Z. (2006). *Geoprostorne baze podataka*. Golden Marketing – Tehnička knjiga.
- Garcia-Molina, H. i Salem, K. (1987). Sagas. *SIGMOD Rec.*, 16(3):249–259.
- Georgakopoulos, D., Hornick, M., Krychniak, P., i Manola, F. (1994). Specification and management of extended transactions in a programmable transaction environment. In *Proceedings of the 10th International Conference on Data Engineering*, pages 462–473, Houston, TX, USA. IEEE.
- Georgakopoulos, D., Hornick, M., i Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and parallel Databases*, 3(2):119–153.
- Georgakopoulos, D. i Hornick, M. F. (1994). A framework for enforceable specification of extended transaction models and transactional workflows. *International Journal of Intelligent and Cooperative Information Systems*, 3(03):225–253.
- Gray, J. (1981). The transaction concept: Virtues and limitations. In *VLDB*, volume 81, pages 144–154.
- Grefen, P. i Vonk, J. (2006). A taxonomy of transactional workflow support. *International Journal of Cooperative Information Systems*, 15(01):87–118.
- Grover, V., Jeong, S. R., Kettinger, W. J., i Teng, J. T. (1995). The implementation of business process reengineering. *Journal of Management Information Systems*, 12(1):109–144.
- Guabtini, A., Charoy, F., i Godart, C. (2006). Customizable isolation in transactional workflow. In *Interoperability of Enterprise Software and Applications*, pages 197–202. Springer.
- Hansen, F. i Oleshchuk, V. (2003). Srbac: A spatial role-based access control model for mobile systems. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORD-SEC'03)*, pages 129–141.
- Härder, T. i Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys (CSUR)*, 15(4):287–317.
- Hespanha, J. P. (2012). *Development Methodology for an Integrated Legal Cadastre: Deriving Portugal Country Profile from the Land Administration Domain Model*. PhD thesis.
- Hoel, E., Menon, S., i Morehouse, S. (2003). Building a robust relational implementation of topology. *Advances in Spatial and Temporal Databases*, pages 508–524.

- Holtkamp, B. (1990). Flexible cooperation in non-standard application environments. Technical report, Naval Postgraduate School, Monterey, CA, USA. Technical Report NPS52-90-018.
- Hunter, G. J. i Williamson, I. P. (1990). The development of a historical digital cadastral database. *International Journal of Geographical Information System*, 4(2):169–179.
- ISO (2004a). ISO 19125-1:2004, Geographic information – Simple feature access – Part 1: Common architecture. Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2004b). ISO 19125-2:2004, Geographic information – Simple feature access – Part 2: SQL option. Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2004c). Systems and software engineering – High-level Petri nets – Part 1: Concepts, definitions and graphical notation. Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2004d). Systems and software engineering – High-level Petri nets – Part 2: Transfer Format. Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2012a). ISO 19152:2012, Geographic information – Land Administration Domain Model (LADM). Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2012b). ISO/IEC 19505-1:2012 Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure. Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2013). ISO/IEC 19510:2013 Information technology – Object Management Group Business Process Model and Notation. Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- ISO (2016). ISO/IEC 9075-1:2016 Information technology – Database languages – SQL – Part 1: Framework (SQL/Framework). Međunarodna norma, International Organization for Standardization, Geneva, Switzerland.
- Kaiser, G. E. (1990). A flexible transaction model for software engineering. In *Proceedings of Sixth International Conference on Data Engineering*, pages 560–567. IEEE.
- Kamath, M. i Ramamritham, K. (1996). Bridging the gap between transaction management and workflow management. In *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-art and Future Directions*.

- Kung, H.-T. i Robinson, J. T. (1981). On optimistic methods for concurrency control. *ACM Transactions on Database Systems (TODS)*, 6(2):213–226.
- Kuo, D., Gaede, V., i Taylor, K. L. (1998). Using constraints to manage long duration transactions in spatial information systems. In *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems, New York City, New York, USA, August 20-22, 1998, Sponsored by IFCIS, The Intl Foundation on Cooperative Information Systems*, pages 383–395.
- Langran, G. i Chrisman, N. R. (1988). A framework for temporal geographic information. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 25(3):1–14.
- Lee, K. C., Mansfield, W. H., i Sheth, A. P. (1991). An interactive transaction model for distributed cooperative tasks. *Data Engineering*, 14(1):24–28.
- Lemmen, C. i van Oosterom, P. (1995). Efficient and automatic production of periodic updates of cadastral maps. In *Joint European Conference and Exhibition on Geographical Information*, volume 95, pages 137–142, Hague, Netherlands.
- Lemmen, C., Vos, J., i Beentjes, B. (2017). Ongoing development of land administration standards : Blockchain in transaction management. *European property law journal*, 6(3):478–502.
- Leymann, F. i Roller, D. (2000). *Production workflow: concepts and techniques*. Prentice Hall PTR Upper Saddle River.
- Matijević, H. (2006). *Modeliranje promjena u katastru*. PhD thesis, Geodetski fakultet, Kačićeva 26, Zagreb.
- Matijević, H., Biljecki, Z., Pavičić, S., i Roić, M. (2008). Transaction processing on planar partition for cadastral application. In *Proceedings FIG Working Week 2008, Stockholm, Sweden*.
- Meng, J., Su, S. Y., Lam, H., i Helal, A. (2002). Achieving dynamic inter-organizational workflow management by integrating business processes, events and rules. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 10–pp. IEEE.
- Moss, J. E. B. (1981). *Nested Transactions: An Approach to Reliable Distributed Computing*. PhD thesis, Massachusetts Institute of Technology.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

- Muro, S., Kameda, T., i Minoura, T. (1984). Multi-version concurrency control scheme for a database system. *Journal of Computer and System Sciences*, 29(2):207 – 224.
- Navratil, G. i Frank, A. U. (2004). Processes in a cadastre. *Computers, Environment and Urban Systems*, 28(5):471–486.
- Nodine, M. H. i Zdonik, S. B. (1992). Cooperative transaction hierarchies: Transaction support for design applications. *The VLDB Journal*, 1(1):41–80.
- OGC (2004). OGC Open Modelling Interface Interface Standard. Version: 2.0. Technical report, Open Geospatial Consortium.
- OMG (2013). OMG Business Process Model and Notation (BPMN) – Version 2.0.2. Technical report, Object Management Group.
- OMG (2017). OMG Unified Modeling Language (OMG UML) – Version 2.5.1. Technical report, Object Management Group.
- Petri, C. A. (1962). *Kommunikation mit automaten*. PhD thesis, Institut für instrumentelle Mathematik der Universität Bonn, Bonn, Germany.
- Petri, C. A. (1966). Communication with automata. Technical report, Institut für instrumentelle Mathematik der Universität Bonn.
- Peuquet, D. i Wentz, E. (1994). An approach for time-based analysis of spatiotemporal data. In *Advances in GIS research. Proc. 6th symposium, Edinburgh, 1994. Vol. 1*. Taylor & Francis, London.
- Peuquet, D. J. i Duan, N. (1995). An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 9(1):7–24.
- Phoung, T. H. (2011). *Enhancing transparency in land transaction process by reference architecture for workflow management*. PhD thesis, Faculty of Geo-Information Science and Earth Observation of the University of Twente, Enschede, Netherlands.
- Pu, C., Kaiser, G. E., i Hutchinson, N. C. (1988). Split-transactions for open-ended activities. In *14th International Conference on Very Large Data Bases, VLDB '88*, pages 26–37, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Ramadan, M., Elmongui, H. G., i Hassan, R. (2011). BPMN formalisation using coloured petri nets. In *Proceedings of the 2nd GSTF Annual International Conference on Software Engineering & Applications (SEA 2011)*.
- van Osch, B. i Lemmen, C. (2004). Towards the introduction of workflow management at the netherlands cadastre. In *FIG Working Week 2004*.

- Roić, M. (2012). *Upravljanje zemljišnim informacijama – katastar*. Geodetski fakultet.
- Rusinkiewicz, M. i Sheth, A. (1995). Specification and execution of transactional workflows. In Kim, W., editor, *Modern Database Systems*, pages 592–620. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.
- Russell, N., van der Aalst, W. M., Ter Hofstede, A. H., i Edmond, D. (2005). Workflow resource patterns: Identification, representation and tool support. In *International Conference on Advanced Information Systems Engineering*, pages 216–232. Springer.
- Russell, N., van der Aalst, W. M., Ter Hofstede, A. H., i Wohed, P. (2006). On the suitability of UML 2.0 activity diagrams for business process modelling. In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, volume 53, pages 95–104. Australian Computer Society, Inc.
- Sari, K. W. (2010). The Workflow of maintenance of Cadastral data as based on Land Administration Domain Model (LADM)–A case study in Indonesia. Master's thesis, Faculty of Geo-Information Science and Earth Observation, University of Twente, Enschede, Netherlands.
- Schick, S., Meyer, H., i Heuer, A. (2011). Enhancing workflow data interaction patterns by a transaction model. In *15th East-European Conference on Advances in Databases and Information Systems (ADBIS 2011)*, volume 789 of *CEUR Workshop Proceedings*, pages 33–44, Vienna, Austria.
- Sheth, A. P., Rusinkiewicz, M., i Karabatis, G. (1992). Using polytransactions to manage interdependent data. In Elmagarmid, A. K., editor, *Database Transaction Models for Advanced Application*, pages 219–263. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Shklar, L. i Rosen, R. (2009). *Web application architecture*. John Wiley & Sons.
- Tauro, C. J., Aravindh, S., i Shreeharsha, A. (2012). Comparative study of the new generation, agile, scalable, high performance NOSQL databases. *International Journal of Computer Applications*, 48(20):1–4.
- Unland, R. i Schlageter, G. (1991). A flexible and adaptable tool kit approach for transaction management in non standard database systems. *Data Engineering*, 14(1):49–54.
- Van der Aalst, W. (2000). Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. *Information & Management*, 37(2):67–75.
- Van Der Aalst, W. M. (1996). Three good reasons for using a Petri-net-based workflow management system. In *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, pages 179–201. Cambridge, Massachusetts.

- Van der Aalst, W. M. (1998). The application of Petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66.
- Van Der Aalst, W. M. (2000). Workflow verification: Finding control-flow errors using petri-net-based techniques. *Business Process Management*, 1806:161–183.
- Van Der Aalst, W. M., Aldred, L., Dumas, M., i ter Hofstede, A. H. (2004). Design and implementation of the YAWL system. In *CAiSE*, volume 3084, pages 142–159. Springer.
- Van der Aalst, W. M. i ter Hofstede, A. (2002). Workflow patterns: On the expressive power of (petri-net-based) workflow languages. In *Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002)*, volume 560, pages 1–20.
- Van der Aalst, W. M., ter Hofstede, A. H., Kiepuszewski, B., i Barros, A. P. (2003). Workflow patterns. *Distributed and parallel databases*, 14(1):5–51.
- Van der Aalst, W. M. P. i ter Hofstede, A. (2005). YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275.
- van Oosterom, P. (1997). Maintaining consistent topology including historical data in a large spatial database. In *Proceedings of Auto-Carto 13*, pages 327–336.
- Van Oosterom, P., Stoter, J., Quak, W., i Zlatanova, S. (2002). The balance between geometry and topology. In *Advances in Spatial Data Handling*, pages 209–224. Springer.
- Vossen, G., Weske, M., i Wittkowski, G. (1996). Dynamic workflow management on the web. Technical report, University of Münster. Technical Report No. 24/96-I.
- Vranić, S., Matijević, H., i Roić, M. (2015). Modelling outsourceable transactions on polygon-based cadastral parcels. *International Journal of Geographical Information Science*, 29(3):454–474.
- Wächter, H. i Reuter, A. (1992). The contract model. In Elmagarmid, A. K., editor, *Database Transaction Models for Advanced Application*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Weikum, G. i Schek, H.-J. (1991). Multi-level transactions and open nested transactions. *Data Engineering*, 14(1):60–64.
- Weikum, G. i Schek, H. J. (1992). Concepts and applications of multilevel transactions and open nested transactions.
- Weikum, G. i Vossen, G. (2001). *Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery*. Elsevier.



- Weske, M., Vossen, G., Medeiros, C. B., i Pires, F. (1998). Workflow management in geoprocessing applications. In *6th ACM international symposium on Advances in geographic information systems*, pages 88–93. ACM.
- WFMC (1995). Workflow management coalition—the workflow reference model. Technical report, Workflow Management Coalition.
- WFMC (1999). Terminology & glossary—version 2.2. Technical Report Document Number WFMC-TC-1011, Workflow Management Coalition.
- WFMC (2012). Process Definition Interface—XML Process Definition Language—version 2.2. Technical Report Document Number WFMC-TC-1025, Workflow Management Coalition.
- White, S. A. (2004). Process modeling notations and workflow patterns. In Fischer, L., editor, *Workflow Handbook 2004*, pages 265–294. Future Strategies Inc., Lighthouse Point, FL, USA.
- Widom, J. i Finkelstein, S. J. (1990). Set-oriented production rules in relational database systems. *ACM SIGMOD Record*, 19(2):259–270.
- Wohed, P., van der Aalst, W. M., Dumas, M., ter Hofstede, A. H., i Russell, N. (2005). Pattern-based analysis of the control-flow perspective of UML activity diagrams. In *International Conference on Conceptual Modeling (ER05)*, pages 63–78. Springer Verlag.
- Wohed, P., van der Aalst, W. M., Dumas, M., ter Hofstede, A. H., i Russell, N. (2006). On the suitability of BPMN for business process modelling. *Business Process Management*, 4102:161–176.
- Worah, D. i Sheth, A. (1997). Transactions in transactional workflows. In Jajodia, S. i Kerschberg, L., editors, *Advanced Transaction Models and Architectures*, pages 3–34. Springer US.
- Yue, P., Zhang, M., i Tan, Z. (2015). A geoprocessing workflow system for environmental monitoring and integrated modelling. *Environmental Modelling & Software*, 69:128–140.
- Zevenbergen, J., Frank, A., i Stubkjær (2007). *Real property transactions: Procedures, transaction costs and models*. IOS Press.

# Popis tablica

Tablica 2.1	Vrste transakcija (Vranić i dr., 2015) . . . . .	35
Tablica 4.1	Vrste promjena na polu-bridovima . . . . .	91

# Popis slika

Slika 2.1	Stanja transakcije (Weikum i Vossen, 2001)	11
Slika 2.2	Zaključavanje u dvije faze (Weikum i Vossen, 2001)	14
Slika 2.3	Konzervativno zaključavanje u dvije faze (Weikum i Vossen, 2001)	14
Slika 2.4	Strogo zaključavanje u dvije faze (Weikum i Vossen, 2001)	15
Slika 2.5	Altruistično zaključavanje (Weikum i Vossen, 2001)	15
Slika 2.6	Provedba transakcija u skladu s BTO protokolom	17
Slika 2.7	Graf serijalizabilnosti (Weikum i Vossen, 2001)	18
Slika 2.8	Tri faze transakcije kod optimističnih protokola	19
Slika 2.9	MVTO protokol (Weikum i Vossen, 2001)	21
Slika 2.10	2V2PL protokol (Weikum i Vossen, 2001)	22
Slika 2.11	Prostorni i tablični prikaz podataka	28
Slika 2.12	Konkurentne promjene dvaju korisnika (Kuo i dr., 1998)	31
Slika 2.13	Problematično ažuriranje (van Oosterom, 1997)	32
Slika 2.14	Primjer konfliktne situacije (Choi i dr., 1999)	33
Slika 2.15	Upravljanje konkurentnim transakcijama (Choi i dr., 1999)	33
Slika 2.16	Dioba koja zahtijeva promjene na susjednim česticama (Vranić i dr., 2015)	36
Slika 3.1	Trodimenzionalni pogled na tijek rada (Van der Aalst, 1998)	40
Slika 3.2	Referentni model sustava za upravljanje tijekom rada (WFMC, 1995)	41
Slika 3.3	Vrste usmjeravanja tijeka rada (Van der Aalst, 1998)	43
Slika 3.4	Načini pokretanja zadataka (Van der Aalst, 1998)	44
Slika 3.5	Osnovni elementi BPMN-a (Chinosi i Trombetta, 2012)	45
Slika 3.6	BPMN vrste usmjeravanja tijeka rada (Van der Aalst, 1998)	46
Slika 3.7	Vrste usmjeravanja tijeka rada pomoću UML AD-a (Wohed i dr., 2005)	47
Slika 3.8	Vrste usmjeravanja tijeka rada kod Petrijevih mreža (Van der Aalst, 1998)	49
Slika 3.9	Sistematizacija transakcijskih WFMS-a (Grefen i Vonk, 2006)	54
Slika 3.10	Modeli integracije koncepta transakcije i WFMS (Grefen i Vonk, 2006)	55
Slika 3.11	Kompenzacija zadataka prema modelu WF (Grefen i Vonk, 2006)	56
Slika 3.12	Primjer transakcije s jednostavnom strukturom (Grefen i Vonk, 2006)	56
Slika 4.1	Osnovne aktivnosti procesa u katastru	66

Slika 4.2	Definicija pojmova (temelji se na (Vranić i dr., 2015)) . . . . .	67
Slika 4.3	Procesi koji nisu konkurentni . . . . .	68
Slika 4.4	Konkurentni procesi . . . . .	69
Slika 4.5	Konfliktni konkurentni procesi . . . . .	70
Slika 4.6	UML dijagram klasa modula za definiciju i interpretaciju procesa . . .	72
Slika 4.7	Povezivanje općenitoga modela WFMS-a s katastarskim česticama . .	74
Slika 4.8	Objektni dijagram čestica povezanih pomoću jednostavnoga modela .	75
Slika 4.9	Prošireni konceptualni model WFMS-a . . . . .	76
Slika 4.10	Promjena statusa verzija katastarskih čestica . . . . .	77
Slika 4.11	Dioba katastarske čestice . . . . .	78
Slika 4.12	Dioba katastarske čestice (objektni dijagram) . . . . .	78
Slika 4.13	Korekcija granice katastarske čestice . . . . .	79
Slika 4.14	Korekcija granice katastarske čestice (objektni dijagram) . . . . .	80
Slika 4.15	Pesimistični tijek rada . . . . .	81
Slika 4.16	Konkurentni procesi (dioba čestica) . . . . .	82
Slika 4.17	Provedba procesa prema pesimističnome tijeku rada . . . . .	83
Slika 4.18	Optimistični tijek rada . . . . .	84
Slika 4.19	Konfliktni konkurentni procesi (spajanje čestica i korekcija granice) . .	85
Slika 4.20	Provedba procesa prema optimističnome tijeku rada . . . . .	85
Slika 4.21	Prednost uvođenja međnih točaka u model podataka . . . . .	86
Slika 4.22	Dijagram klasa modela procesa proširen s točkama . . . . .	87
Slika 4.23	Moguće promjene nad čvorovima . . . . .	90
Slika 4.24	Prepoznavanje promjena na poligonu . . . . .	92
Slika 4.25	Konkurentni procesi (dioba čestica) - s točkama . . . . .	93
Slika 4.26	Altruistični tijek rada . . . . .	94
Slika 4.27	Dodavanje podrške za modeliranje procesa u LADM . . . . .	96
Slika 4.28	Veza između WFMS-a i LADM katastarskih čestica . . . . .	97
Slika 5.1	Dio logičkoga modela podataka (katastarski podaci) . . . . .	100
Slika 5.2	FME radna površina za učitavanje podataka u PostgreSQL bazu po- dataka . . . . .	101
Slika 5.3	Korišteni katastarski podaci prikazani u testnome sustavu . . . . .	102
Slika 5.4	Dohvaćanje dostupnih akcija u proces . . . . .	103
Slika 5.5	Prijelaz <i>Definiranje obuhvata</i> . . . . .	103
Slika 5.6	Definicija elemenata tijeka rada . . . . .	104
Slika 5.7	Definicija sekvencijalnih lukova . . . . .	104
Slika 5.8	Prikaz uzorka OR dijeljenje . . . . .	105
Slika 5.9	Definicija lukova za modeliranje OR dijeljenja . . . . .	105
Slika 5.10	Definicija lukova za modeliranje OR spajanja . . . . .	106
Slika 5.11	Konkurentni procesi sa zajedničkom susjednom česticom . . . . .	109

Slika 5.12	Vremenski raspored provedbe procesa prema pesimističnome tijeku rada . . . . .	109
Slika 5.13	Vremenski raspored provedbe procesa prema optimističnome tijeku rada . . . . .	110
Slika 5.14	Konkurentni procesi sa zajedničkom susjednom česticom . . . . .	111
Slika 5.15	Konkurentni procesi sa zajedničkom susjednom česticom . . . . .	112

# Popis kratica

2PL	Two-phase locking
2V2PL	Two-version 2PL
AL	Altruistic locking
ACID	Atomicity, Consistency, Isolation, Durability
BOCC	Backward-oriented optimistic concurrency control
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Management Notation
BPR	Business Process Re-engineering
BTO	Basic timestamp ordering
C2PL	Conservative two-phase locking
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
COLT	Constraint based Long Transaction
DBMS	Database Management System
DEMLAS	Development of Multipurpose Land Administration System
FIFO	First-in-first-out
FME	Feature Manipulation Engine
FOCC	Forward-oriented optimistic concurrency control
GML	Geography Markup Language
ISO	International Standardization Organization

LADM	Land Administration Domain Model
MV2PL	Multiversion two-phase locking
MVTO	Multiversion timestamp ordering
NoSQL	Not Only SQL
OMG	Object Management Group
OGC	Open Geospatial Consortium
PNML	Petri Net Markup Language
RBAC	Role-based Access Control
SGT	Serializability graph tester
SOAP	Simple Object Access Protocol
SRCE	Sveučilišni računalni centar
SQL	Structured Query Language
S2PL	Strict two-phase locking
SS2PL	Strong strict two-phase locking
STO	Strict timestamp ordering
SUBP	Sustav za upravljanje bazama podataka
TO	Timestamp ordering
UML	Unified Modeling Language
UML AD	UML Activity Diagram
WAPI	Workflow application programming interface
WF	Workflow
WFMC	Workflow Management Coalition
WFMS	Workflow Management System
WFS	Web Feature Service
WMS	Web Map Service
XDPL	XML Data Processing Language
XML	eXtensible Markup Language
YAWL	Yet Another Workflow Language

# Životopis

Saša Vranić rođen je 20. ožujka 1986. godine u Sisku. Nakon završene je osnovne škole upisao i pohađao Geodetsku tehničku školu u Zagrebu. Maturirao je 2004. godine s odličnim uspjehom.

Iste se godine upisuje na dodiplomski sveučilišni studij na Geodetskom fakultetu Sveučilišta u Zagrebu. Tijekom studija obavljao je demonstrature iz kolegija Geodetski planovi te sudjelovao u terenskim izmjerama u okviru projekata pod vodstvom prof. dr. sc. Marka Džape. Diplomirao je u srpnju 2009. godine pod mentorstvom prof. dr. sc. Miodruga Roića na temu diplomskog rada "Sučelje katastarske baze podataka" u okviru kojega je bilo potrebno razviti web aplikaciju<sup>1</sup> za pregledavanje, pretraživanje i preuzimanje katastarskih podataka korištenjem standardnih OGC servisa.

Po završetku se studija, u kolovozu 2009. godine zapošljava u tvrtki Geofoto d. o. o. na radnome mjestu GIS konzultanta. Na radnome se mjestu bavi poslovima upravljanja i modeliranja prostornim podacima i sudjeluje u izradi računalnih i web aplikacija.

U rujnu 2012. godine zapošljava se na Geodetskom fakultetu na Katedri za zemljomjerstvo, na radnome mjestu asistenta. Iste godine upisuje poslijediplomski doktorski studij na Geodetskom fakultetu. Održava vježbe iz sljedećih kolegija: Izmjera zemljišta, Digitalni planovi i Geodetski planovi.

Dosad je u koautorstvu objavio nekoliko članaka u međunarodnim znanstvenim časopisima te aktivno sudjelovao na više međunarodnih i domaćih znanstvenih skupova. Kao suradnik je sudjelovao na znanstveno-stručnom projektu "NIPP specifikacija za transformacije prema INSPIRE-u" za potrebe Državne geodetske uprave te projektu "Transformacija i harmonizacija podataka ZIPP-a" za potrebe Grada Zagreba, oba pod voditeljstvom izv. prof. dr. sc. Vlade Cetla. Sudjelovao je na skupu Dani IPP 2015, na kojemu je održao radionicu za članove radnih skupina NIPP-a gdje je prezentirao mogućnosti aplikacije HALE s ciljem harmonizacije i transformacije INSPIRE skupova podataka. Trenutno sudjeluje kao istraživač na projektu DEMLAS<sup>2</sup> koji financira Hrvatska zaklada za znanost.

U stručnom je radu sudjelovao u izradi računalnih aplikacija za održavanje digitalnoga katastarskog plana, potom na izradi mrežnih portala za pregledavanje katastarskih podataka u nekoliko zemalja u Europi. Bavio se i terenskim izmjerama za katastar i inženjersku geodeziju. Koristi programske jezike JavaScript, PHP, SQL, Python i PL/SQL te

---

<sup>1</sup><http://www.pg.geof.unizg.hr/services/katastar/>

<sup>2</sup><http://demlas.geof.unizg.hr>



razvojne okvire OpenLayers, jQuery, GEOS, GDAL/OGR i CGAL.

Član je Hrvatske komore ovlaštenih inženjera geodezije i Hrvatskog geodetskog društva. Govori i piše engleski jezik.

svranic@geof.hr

# Popis objavljenih radova

## Izvorni znanstveni i pregledni radovi u CC časopisima

1. Vučić, Nikola; Roić, Miodrag; Mađer, Mario; **Vranić, Saša**; van Oosterom, Peter. *Overview of the Croatian Land Administration System and the Possibilities for Its Upgrade to 3D by Existing Data*. ISPRS International Journal of Geo-Information. 6 (2017), 7; 223-1-223-20. DOI: 10.3390/ijgi6070223.
2. **Vranić, Saša**; Matijević, Hrvoje; Roić, Miodrag. *Modelling outsourcable transactions on polygon based cadastral parcels*. International journal of geographical information science. 29 (2015), 3; 454-474. DOI: 10.1080/13658816.2014.981190.

## Znanstveni radovi u zbornicima skupova s međunarodnom recenzijom

1. **Vranić, Saša**; Matijević, Hrvoje; Roić, Miodrag. *Application of Workflow Management System to the Modelling of Processes in Land Administration Systems*. Proceedings of the 7th Land Administration Domain Model Workshop / Lemmen, Christiaan; van Oosterom, Peter; Fendel, Elfriede (ur.). Zagreb, Hrvatska. International Federation of Surveyors (FIG), 2018. 125-139.
2. Vučić, Nikola; Mađer, Mario; Roić, Miodrag; **Vranić, Saša**. *Towards a Croatian 3D cadastre based on the LADM*. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W4 / Karas, I. R.; Isikdag, U; Rahman, A. A. (ur.). 2017. 399-409.
3. Vučić, Nikola; Roić, Miodrag; Mađer, Mario; **Vranić, Saša**. *Overview of Legal and Institutional Aspects of Croatian Cadastre and Possibilities for its Upgrading to 3D*. Proceedings 5th International Workshop on 3D Cadastres / Peter van Oosterom, Efi Dimopoulou, Elfriede Fendel (ur.). Copenhagen, Denmark. International Federation of Surveyors (FIG), 2016. 61-80.
4. **Vranić, Saša**; Jurakić, Goran; Matijević, Hrvoje. *Modelling and Dissemination of Land Survey Data*. Proceedings of the 6th International Conference on Engineering Sur-

veying - INGEO 2014 / Kopačik, Alojz; Kyrinovič, Peter, Štroner, Martin (ur.). Prague, Czech Republic: Czech Technical University Prague, Faculty of Civil Engineering, 2014. 149-154.

5. **Vranić, Saša**; Matijević, Hrvoje; Cetl, Vlado. *Modeliranje i diseminacija podataka izmjere zemljišta*. Zbornik radova s petog hrvatskog kongresa o katastru s međunarodnim sudjelovanjem / Medak, Damir; Rezo, Milan; Zrinjski, Mladen (ur.). Zagreb: Hrvatsko geodetsko društvo, 2014. 43-52.
6. Matijević, Hrvoje; Biljecki, Zvonko; **Vranić, Saša**. *Correctness Preconditions For Changes On Geometry Of Cadastral Parcels With Analysis Of Implementation Options*. Proceedings of Papers – 1st Serbian Geodetic Congress / Joksic, Dusan (ur.). Beograd: Republic Geodetic Authority, 2011. 201-208.

## Ostali radovi

1. Roić, Miodrag; **Vranić, Saša**; Kliment, Tomaš; Stančić, Baldo; Tomić, Hrvoje. *Development of Multipurpose Land Administration Warehouse*. Proceedings from FIG Working Week 2017: "Surveying the world of tomorrow - From digitalisation to augmented reality". Copenhagen: International Federation of Surveyors, 2017. 1-12.
2. **Vranić, Saša**; Redovniković, Loris; Pleić, Marko. *Integracija CAD i GIS funkcionalnosti u svrhu izrade mrežne aplikacije za izradu katastarskog plana*. Zbornik radova X. simpozija ovlaštenih inženjera geodezije - Prostorni registri za budućnost. 2017. 1-8.
3. **Vranić, Saša**; Matijević, Hrvoje. *Workflows for Ensuring Consistency of Cadastral Data*. Proceedings from FIG Working Week 2015: "From the Wisdom of the Ages to the Challenges of the Modern World". Copenhagen: International Federation of Surveyors, 2015. 1-11.
4. Kliment, Tomáš; Schleidt, Kathi; **Vranić, Saša**; Tuchyňa, Martin. *A new way to collect INSPIRE data in the field - mobileWFS*. Proceedings of INSPIRE Conference 2016. Barcelona, Španjolska, 2016.
5. **Vranić, Saša**; Mađer, Mario; Matijević, Hrvoje; Bašić, Tomislav. *Zapisivanje podataka CROPOS GNSS mjerenja u XML formatu - korak ka digitalnom geodetskom elaboratu*. Zbornik radova 4. CROPOS konferencije. Bašić, Tomislav ; Pavasović, Marko ; Marjanović, Marijan (ur.). Zagreb: Državna geodetska uprava, 2015. 140-153.
6. Cetl, Vlado; **Vranić, Saša**; Matijević, Hrvoje. *Development of a Model For Land Survey Data based on ISO 19156*. Abstracts of Forum of Young Geoinformaticians 2014. Mogroš, Martin (ur.). Zvolen: Technical University in Zvolen, 2014. 4-4.

7. **Vranić, Saša**; Stančić, Baldo; Cetl, Vlado. *Historical development of land surveying instruments and techniques*. Global Environment, Stakeholders' Profile and Corporate Governance in Geodesy - Programme and Abstracts. Tutić, Dražen (ur.). Zagreb: Intergrafika, 2014. 45-45.
8. **Vranić, Saša**; Pavičić, Stipica; Biljecki, Zvonko. *Katastarski podaci i GML*. Zbornik radova IV. simpozija ovlaštenih inženjera geodezije. Ramić, Siniša (ur.). Zagreb: Hrvatska komora ovlaštenih inženjera geodezije, 2011. 81-90.
9. **Vranić, Saša**; Čarapar, Ivona; Biljecki, Zvonko. *Mogućnosti GML-a kao razmjenskog formata katastarskih podataka*. Zbornik radova III. simpozija ovlaštenih inženjera geodezije - GIS, fotogrametrija i daljinska istraživanja u službi geodezije i geoinformatike. Zagreb: Hrvatska komora ovlaštenih inženjera geodezije, 2010. 91-98.

## **Vođenje diplomskih radova**

1. Pleić, Marko. *Održavanje geodetskog plana na web-u*. Završni rad - diplomski/integralni studij. Zagreb : Geodetski fakultet, 07.07. 2017, 84 str. Mentor: Redovniković, Loris. Voditelj: **Vranić, Saša**.