



godina
years

Geodetskog fakulteta
of Faculty of Geodesy

Zagreb, 26.-27. rujna 2002.

M. Roić, H. Matijević, V. Cetl

OBJEKTNOORJENTIRANO MODELIRANJE KATASTRA

SAŽETAK. Upravljanje podacima već dulje vrijeme olakšava njihova organizacija relacijskim modelom. Zbog svoje, kako kvalitativne tako i kvantitativne strukture, geoprostorni podaci postavljaju visoke zahtjeve pri modeliranju. Kompleksnost katastarskih podataka u odnosu sa jednostavnošću, a time i ograničenjima relacijskog modela, rezultiraju potrebom za sustavom osnovanom na potpuno drugačijim temeljima. Objektnoorijentirane baze podataka daju odgovor na navedene probleme. Ovim je radom prikazano objektnoorijentirano modeliranje kroz važne segmente koji se u njemu pojavljuju, zajedno s konceptijskim prikazom modela katastarskih podataka. Dan je i posebni osvrt na unified modeling language (UML) kao snažno i učinkovito sredstvo interakcije između proizvođača softvera i geodetskog stručnjaka koji može i mora odrediti zahtjeve koje konačni sustav treba ispunjavati.

Ključne riječi: katastarski podaci, objektnoorijentirano modeliranje, UML.

OBJECT-ORIENTED CADASTRE MODELLING

ABSTRACT. Data management has been facilitated for a longer period already through their organisation by means of a relation model. Because of their quality, as well quantity structure, geospatial data pose great demands upon modelling. The complexity of cadastral data as related to the simplicity, and thereby to the limitations of a relation model, results in the need for a system based on completely different bases. The object-oriented databases give an answer to the above-mentioned problems. This paper presents the object-oriented modelling through important segments appearing in it together with the conceptual presentation of cadastral data model. There is also a comment given on unified modelling language (UML) as a powerful and efficient means of interaction between the software producer and a surveyor that can and must determine the demands that the final system is to fulfil.

Key words: cadastral data, object-oriented modelling, UML.

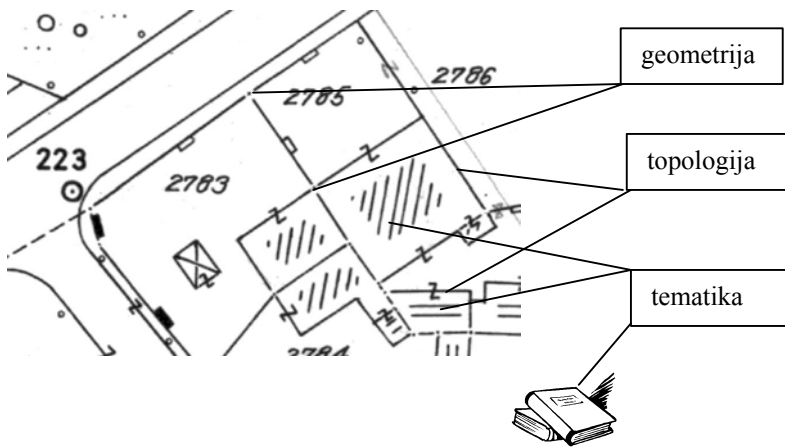
Prof. dr. sc. Miodrag Roić, Hrvoje Matijević, dipl. ing., Vlado Cetl, dipl. ing. Sveučilište u Zagrebu, Geodetski fakultet, Zavod za inženjersku geodeziju i upravljanje prostornim informacijama, Kačićeva 26, Zagreb, e-mail: mroiic@geof.hr, hmatijev@geof.hr, vcetl@geof.hr.

1. UVOD

Već stoljećima tradicionalni katastarski sustavi priskrbljuju društvu više ili manje kvalitetne podatke o prostoru. Iako se kod nas, s različitim društvenim uređenjima svrha održavanja katastarskih podataka mijenjala, ona je neovisno o metodama njihova modeliranja, prikupljanja, obrade te održavanja, stalno prisutna (Roić 1998).

Svi prostorni, pa tako i katastarski podaci sastavljeni su od tri vrste: Geometrijske - koja određuje položaj pojedinih objekata u prostoru kroz njihove koordinate, topološke - koja povezuje pojedine dijelove objekata u smislenu cjelinu, te tematske koja kroz prva dva sloja u prostoru određenim objektima daje dodatni sadržaj. Planovi su uvijek bili modeli, no raspoloživa tehnologija nije dozvoljavala njihovo korištenje na fleksibilan način (Kaufmann, Steudler 1998).

Model je osnova katastarskog informacijskog sustava bez obzira na tehnologiju njegova održavanja. Tako su prije uvođenja računalne tehnologije u sustav održavanja katastra geometrijski, topološki i manji dio tematskog sloja podataka održavani na analognim katastarskim planovima (slika 1). Glavnina je tematskog sadržaja, kako se ne bi opterećivali planovi, pohranjivana u odvojenu, također analognu bazu podataka (knjižni dio operata). Vezu između dvije baze predstavljao je jedinstveni identifikator osnovnog objekta sustava, broj katastarske čestice.



Slika 1. Vrste podataka na katastarskom planu

S razvojem računalne tehnologije 1980-ih godina otvorila se mogućnost digitalizacije, u početku samo dijela tematske vrste podataka. Geometrija i topologija i dalje su pohranjivane na analogne planove, sa svim njihovim poznatim nedostacima. Tematski segment je ipak postao lakši za održavanje i korištenje zahvaljujući u prvom redu pojavi relacijskog modela baza podataka koji je idealan za podatke koji nemaju izraženu hijerarhijsku strukturu.

Drugi važan trenutak u evoluciji prostornog podatkovnog sustava ka digitalnoj tehnologiji je popularizacija CAD sustava. Mnoge su institucije upravo u CAD-u vidjele rješenje svih svojih problema povezanih s prostornim podacima. Prva dva sloja podatka pohranjeni su sada u CAD-

u te logički razvrstani u slojeve, a uz to i preko istog onog jedinstvenog identifikatora povezani sa relacijskom bazom tematskih podataka. Na prvi pogled savršeno no i ovaj je model pokazao brojne nedostatke. Redundantnost podataka, problemi standardizacije proizvođačkih (propriety) CAD formata datoteka kao i njihovo centralizirano održavanje isključili su tzv. spaghetti model podataka kao opciju za izgradnju sustava za upravljanje prostornim podacima. Prvi i najjednostavniji oblik strukture podataka čini organizacija vektorskih podataka u tzv. spaghetti-format (Bill, Fritsch 1994).

Pošto su grafički elementi i na njih vezani atributi spremljeni u istu bazu podataka, veza je automatski ažurirana kada se jedan ili drugi ukloni, makne ili promijeni (Mortson 2000)

Kao jedino preostalo rješenje pojavio se dakle model zajedničke baze podataka koja bi držala sva tri njihova sloja. No relacijski model organizacije podataka ovdje nailazi na problem koji uvelike umanjuje njegovu učinkovitost i mogućnost implementacije. Priroda prostornih podataka je kompleksna te zahtijeva intenzivno korištenje *mnogi-na-mnoge* relacija (many-to-many relationships) koje u relacijskom modelu nisu (nativno) podržane.

Tek objektno orijentirani model podataka, odnosno pojava baza podataka (krajem 80-ih godina prošlog stoljeća) koje ga podržavaju daju zadovoljavajući odgovor na zahtjeve koje postavljaju prostorni podaci.

2. OBJEKTNOORIJENTIRANO MODELIRANJE

Za razliku od tematskog dijela prostornih podataka koji se zbog svoje slabe hijerarhijske razvedenosti dobro uklapa u relacijski model njihov je geometrijski dio orijentiran izrazito ka objektnom modelu.

Model podataka je konceptijska predstava podatkovnih struktura koje zahtijeva baza podataka. Podatkovne strukture uključuju same podatke, veze među njima te pravila kojima su oni upravljani (<http://www.utexas.edu/>).

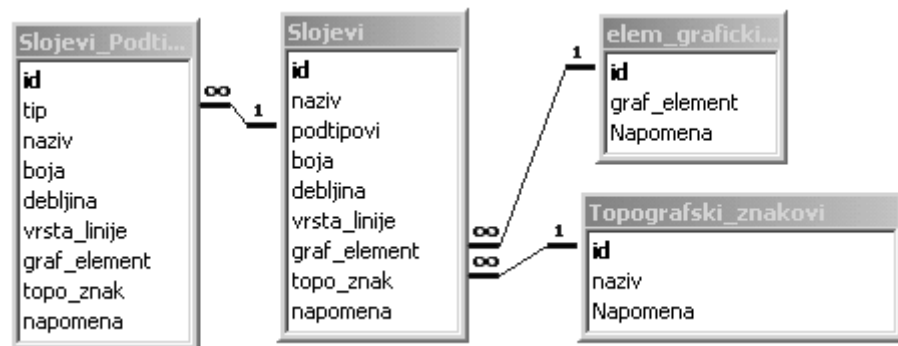
Svrha svake, pa tako i baze geoprostornih podataka je osigurati što je to moguće bolju aproksimaciju promatranog dijela stvarnog svijeta. No prije bilo kojeg koraka u smjeru njezina oblikovanja potrebno je dakako odrediti model podataka kojima će ona kasnije biti popunjena.

Dva su standardna pristupa modeliranju objekata iz stvarnog svijeta za njihovu pohranu u bazu podataka, strukturalni i objektni.

Prije objektno revolucije sve je bilo dizajnirano na osnovu strukturalnog modela sustava (Haigh 2001). Ovaj se pristup temelji na razlaganju objekata iz stvarnog svijeta na dijelove, a njegova pokretačka snaga je proces. Sve je dakle utemeljeno na unaprijed strogo definiranom skupu operacija odnosno procesu. Svaki njegov dio mora ponovo razmjenjivati podatke s bazom preko sučelja koje osigurava skup općenitih metoda upravljanja. Ugrađivanje specifičnijih metoda koje bi olakšale manipulaciju podacima znatno opterećuje promatrano sučelje do te mjere da dovodi u opasnost njegovu funkcionalnost.

Plod ovakvog načina razmišljanja su baze podataka temeljene na relacijskom modelu (slika 2). Objekti iz stvarnog svijeta rastavljeni su na dijelovi koji ih u dovoljnoj mjeri uopćavaju. Ti su sastavni elementi tada pohranjeni u odvojene dijelove (tablice) jedne baze podataka koji su međusobno povezani nekim zajedničkim atributom (ključ). Korištenjem unaprijed definiranog

sučelja (SQL) prema tako nastaloj bazi moguće je iz pojedinih dijelova, a korištenjem ključa, sastaviti predstavu pojedinog objekta ili neku od kombinacija njegovih dijelova.



Slika 2. Relacijski model podataka

Objektnoorijentirano modeliranje s druge strane koncentrirano je na podatke koji prolaze kroz sustav. Podaci koji predstavljaju objekt, umjesto podijeljeni po tablicama, čuvaju se zajednički u "snopovima". Nekoliko je prednosti ovakvog pristupa. Kao prvo sustav je prilagodljiviji. Pojava nove vrste objekta ne zahtijeva redefiniranje sučelja jer ono nije jedinstveno za cijeli sustav. Svaki pojedinačni tip (klasa) podataka sadržava u sebi skup postupaka koji služi za interakciju s njim.

Strukturalno kompleksne objekte, kakvi u prirodi uglavnom i jesu, puno je jednostavnije i prirodnije modelirati u objektnoorijentiranom okruženju iz jednostavnog razloga što i u prirodi objekt, a ne njegova okolina, sadrži podatke o sebi. Sve je i u prirodi podijeljeno u neke općenite klase (ljudi) koje imaju svoje podklase (muškarci, žene) koje imaju svoje podklase (djeca, mladi, odrasli) itd.

2.1. Objekti i klase objekata

Objekt je stvarni ili apstraktni predmet koji sadrži informaciju (atribute koji ga opisuju) i metode koje dozvoljavaju da njime bude upravljano (Haigh 2001).

Osvrnimo se na prvi dio definicije dakle na atribute koji opisuju neki objekt i to na primjeru perilice rublja. Ako za neku perilicu rublja znamo *marku*, *model*, *serijski broj* i *kapacitet* možemo reći da imamo neku predstavu o tom objektu. Ta četiri obilježja do određene mjere opisuju perilicu tako da npr. možemo dati odgovor na pitanje koliko rublja može oprati. No zanima li nas koliko vrsta programa ona može obaviti trebat će nam dodatni atribut *broj programa*. Objekt je opisan određenim brojem podataka o njemu (njegovih obilježja ili atributa) ovisno o vrsti upita koji se prema njemu postavlja.

Osim podataka o obilježjima (atributima) dodatnu informaciju o nekom objektu dati će nam i podatak o načinima interakcije s njim, koje ćemo dalje zvati postupcima (methods). Skup postupaka koji služe za interakciju s objektom čini sučelje. Sučelje je zaduženo da određuje koje metode objekt podržava, kako se one koriste, što čine, koje argumente zahtijevaju te ako daje rezultate kakvi su oni. U perilicu možemo umetati rublje, vaditi rublje, dodati sredstvo za

pranje, možemo je uključiti, isključiti itd. Sve navedeno su postupci koje nam perilica stavlja na raspolaganje odnosno oni čine njeno sučelje.

No osim obilježja i postupaka koje s nekim objektom iz prirode možemo učiniti, on može biti i u različitim stanjima. Perilica može biti u stanju uzimanja vode, predpranja, ispiranja itd. Ono može nastati kao rezultat nekog postupka (postupak-uključiti rezultira stanjem-uključena) ili može služiti za prikaz objekta tijekom postupka.

Svaki objekt možemo dakle definirati njegovim:

- atributima,
- metodama i
- stanjem.

Dajemo li tako prema navedenom, definicije nekoliko objekata učit ćemo da oni imaju zajedničke attribute, metode i stanja. Kako bi izbjegli pojedinačno definiranje svakog objekta, a s tim istim obilježjima, možemo reći da postoji skup na kojem su ona definirana, a svaki objekt koji ih ima je član tog skupa. Nazovemo li taj skup klasom (*class*) odredili smo vrlo važan odnos objektnoorijentiranog modeliranja. Uopćeno možemo reći: klasa je definicija objekta, a objekt je instanca (pojava) klase.

2.2. Osnovni mehanizmi objektnoorijentiranog modeliranja

Nakon što smo kroz objekte i klase definirali temelje objektnoorijentiranog modeliranja, a kako bi omogućili razumijevanje nastavka teksta treba reći nekoliko riječi i o osnovnim mehanizmima.

Iako smo prethodno utvrdili važnost zajedničkog čuvanja povezanih podataka, odvajanje generičkih (zajedničkih) podataka od konkretnih jednako je važan mehanizam. Sustav će biti brži i jednostavniji ako može upravljati podacima na općenitijoj razini.

Dobar sustav za upravljanje podacima korisniku pokazuje samo onoliko koliko on mora znati. Korisnik dakle ne treba biti svjestan načina na koji je objekt implementiran te ne mora uopće znati koliko je on složen. Ako se korisniku na raspolaganje stavi samo skup postupaka koji su mu potrebni za interakciju sa sustavom direktno je uklonjena mogućnost da on nesvjesno učini štetu. Definiranje takvog sučelja koje od korisnika skriva ono što on ne treba znati zove se ućahurenje (encapsulation).

Općenito u statičnom svijetu objektnoorijentirano modeliranje ne pruža posebne prednosti. Razmotrimo trivijalan primjer iz stvarnog svijeta, odlazak u trgovinu. Svaki dan idemo u trgovinu, točno znamo na kojoj se polici što nalazi te lako dolazimo do željenih namirnica sami. Dok se ne promijeni raspored polica nema problema. No mi živimo u veoma dinamičnom svijetu u kojem su promjene stalne, pa se tako i raspored polica promijenio i zakomplicirao nam pronalaženje namirnica. Ako u trgovini postoji službenik koji uvijek zna gdje se što nalazi, te posjeduje sučelje koje nam stavi na raspolaganje za postavljanje upita o onome što nas zanima, ne moramo ni znati gdje se točno nalazi koja namirnica. Jednostavno od njega zatražimo potrebne namirnice, ne opterećujući se razmišljanjem otkuda njemu ta informacija. Ti su podaci ućahureni u njemu.

Drugi veoma važan mehanizam implementiran u objektnoorijentiranom modeliranju je nasljeđivanje (*inheritance*). Prema definiciji objektivne orijentacije, ono svojstva zajednička skupu dječjih (*child*) klasa svrstava u jednu klasu koja onda postaje njima roditeljska (*parent*). Dva su načina nasljeđivanja izvedeno (*derived*) i uopćeno (*abstracted*). Izvedeno nasljeđivanje je postupak u kojem iz jedne roditeljske klase izvodimo nove dječje koje imaju neka specifičnija svojstva dok od roditeljske nasljeđuju zajednička. Uopćeno nasljeđivanje je obrnuti postupak. Traženjem zajedničkih svojstava skupa klasa izvodimo njihove roditeljske klase.

3. UNIFIED MODELING LANGUAGE

Popularizacijom, posebno objektnoorijentiranog programiranja ali isto tako i općenito modeliranja pojavila se potreba za ujednačavanjem metoda i tehnika koje se pri tome koriste. Vodeći stručnjaci navedenog polja sastali su se sredinom devedesetih godina prošlog stoljeća i postavili temelje za Unified Modeling Language.

UML je grafički jezik za vizualizaciju, specificiranje, konstruiranje i dokumentiranje dijelova softverski intenzivnih sustava. UML nudi standardni način konstruiranja nacrtu sustava, uključujući konceptijske stvari kao poslovne procese i sustavne funkcije, kao i konkretne stvari poput naredbi programskih jezika, šema baza podataka te ponovoiskoristivih softverskih komponenti (OMG 2001).

Osvrnimo se ovdje i na nekoliko najosnovnijih pravila pisanja u UMLu. Prisjetimo se prethodno navedenog primjera i perilice. Kako bi se razlikovalo od ostalog teksta dijelovi UML notacije biti će pisani podebljano (bold). Notacija za klasu je **ImeKlase**, a za primjer perilice bilo bi to **PerilicaRublja**. Govorimo li o nekom neodređenom objektu određene klase koristimo **:PerilicaRublja**, dakle potertano ime klase koje počinje s dvotočkom. Definiranje objekta neodređene klase je **ImeObjekta**; to je dakle objekt kojem nije određena klasa. Želimo li konačno definirati objekt **MojaPerilica** koja pripada u klasu **PerilicaRublja** napisati ćemo **MojaPerilica:PerilicaRublja** ili objekt **Mazda323** iz klase **Automobil** pišemo **Mazda323:Automobil**.

Veoma važno svojstvo svake klase ali i pojedinog objekta jesu atributi. Općenito, opis atributa određenog objekta (instance klase) sastoji se od riječi bez razmaka od kojih samo prva počinje malim slovom, naprimjer **kapacitetMojePerilice**. Jedina razlika glede notacije atributa klase je činjenica da ćemo cijelu riječ podcrtati kao npr. **kapacitetPeriliceRublja**. Potpuno jednaka pravila važe i za definiranje metoda. Kada definiramo klasu poredati ćemo njeno ime, attribute i metode u pravokutnik kao na slici 3.

PerilicaRublja
marka, model, kapacitet,
staviRublje,izvadiRublje, uključiPerilicu

Slika 3. Definicija klase u UMLu

Pojednostavljeno, grafički možemo klasu prikazivati samo pravokutnikom u kojem je navedeno njeno ime.

4. KLASSE OBJEKATA KATASTRA

Pokušajmo sada odrediti osnovne objekte odnosno klase objekata katastra. Najzornije je možda opet se vratiti na sliku iz uvoda. Ukratko, na njoj vidimo nekoliko stambenih i gospodarskih zgrada te nekoliko katastarskih čestica s pripadajućim međama i brojevima. Većina lomnih točaka pojedinih objekata predstavljene su točkom koja označava njihovu koordinatu. Osim toga na zgradama je nacrtan i znak pripadnosti zgrade pojedinoj čestici.

Pokušajmo sada na osnovu prethodne dvije rečenice koje opisuju djelić katastarskog plana doći do ponekog zaključka o objektima katastra. Gospodarsku i stambenu zgradu možemo bez imalo razmišljanja svrstati u jednu klasu koju nazovimo **Gradevina**.

Jednako očigledan i trivijalan primjer kao i prethodni svakako je i katastarska čestica. Sasvim je sigurno da sve čestice jednog katastarskog sustava imaju zajednička obilježja, bez obzira o kojoj se njegovoj razini radilo (katastarska općina, kotar, ...). Dakle, u skladu s, u prethodnim poglavljima, iznesenim zahtjevima sve katastarske čestice možemo svrstati u klasu koju nazovimo **Katastarska Čestica**.

Proanalizirajmo sada sljedeći pretpostavku:

- U okviru objekta **:Katastarska Čestica** mogu biti kao njegovi atributi pohranjene koordinate pojedinih međnih točaka.

Sasvim je točno da ne postoje nikakve konceptijske prepreke za navedeno, no treba razmisliti o tome što se događa kada se iz nekog razloga promijeni koordinata točke koja je zajednička za više čestica. Ako je ona pohranjena u okviru svake čestice, potrebno je svaku posebno i promijeniti. Samo dodatno se stvar komplicira ako je ista točka i neka od točaka građevine izgrađene na nekoj od promatranih čestica.

Ako tek neznatno promijenimo navedenu pretpostavku tako da glasi:

- U okviru objekta **:Katastarska Čestica** mogu biti pohranjeni pokazivači na informacije o njenim pojedinim međnim točkama,

dobili smo potpuno novi koncept modela sustava. Njime nam se otvara mogućnost vođenja jedinstvenog popisa koordinata svih točaka, odnosno jedinstvene bazu položaja svih promatranih točaka u kojoj, kao uostalom i u prirodi, svako pojedino mjesto prostora ima u okviru jednog jedinstvenog referentnog sustava jedne jedinstvene koordinate. Drugim riječima, svaka lomna točka čestice je zaseban objekt i ima jedinstveni identifikator u okviru sustavne cjeline (npr. katastarska općina). Uvedimo dakle u popis klasa kojima ćemo modelirati naš katastarski sustav i onu koja kao svoje objekte ima položaje pojedinih međnih točaka i nazovimo je **Položaj Točke**. Pomoću navedenog identifikatora ostvarena je jednoznačna veza između geometrijskog sloja podataka (koordinate točke), topološkog (čvor) i tematskog (međna točka).

I iz prakse nam je poznato da se više od jednog tematskog sadržaja može nalaziti na istom mjestu. Vratimo li se još jednom na sliku 1. primijetiti ćemo da se gotovo sve na njoj ucrtane građevine, nalaze upravo na međama čestica odnosno da međe čestica prolaze upravo rubovima zgrada. Vidljivo je dakle da ne samo da nije potrebno uvoditi posebnu klasu za objekte koji opisuju geometriju građevina, već bi to bilo uvođenje redundantnih podataka. Već

prethodno određena klasa **PoložajTočke** poslužiti će nam vrlo dobro i za definiranje geometrijskog sloja podataka o građevinama, a i jednoznačnost je očuvana.

Prije definiranja konceptijskog modela katastarskih podataka, razmotrimo još jedan vid odnosa među mogućim osnovnim objektima odnosno klasama. Nekog posebnog interesa ka definiranju pojedine točke pojedine građevine nećemo imati. One su, barem u našem primjeru, potpuno homogene i u okviru jedne građevine neće imati različitih atributa pa oni mogu biti definirani na razini cjeline (građevine). S točkama međa čestica stvar je ipak složenija. Kod njih će se u okviru jedne čestice ipak pojaviti točke s različitim atributima kao npr. način stabilizacije, elaborat kojim je nastala, kvaliteta itd.

Kako bi u našem modelu omogućili i ovakvu diferencijaciju uvedimo još jednu klasu objekata. Ona će sadržavati tematsku razinu međnih točaka, a zvat ćemo je **MeđnaTočkaČestice**. Geometrijska razina svake međne točke ostaje i dalje definirana u klasi **PoložajTočke**.

Ove četiri klase objekata nikako ne predstavljaju potpuni funkcionalni model katastarskih podataka i nikome ne bi palo napamet formirati katastarski sustav na njima. One predstavljaju samo teoretsku osnovu potrebnu za objašnjenje općenitih načela funkcioniranja objektnoorijentiranog sustava za upravljanje katastarskim podacima.

5. KONCEPT MODELA KATASTRA

U prethodnom poglavlju odabrali smo dakle četiri osnovne klase (vrste) objekata pomoću kojih ćemo konceptijski prikazati, funkcioniranje jednog rudimentarnog objektnoorijentiranog katastarskog sustava. Izaberimo prvo po nekoliko najvažnijih atributa i metoda pojedine klase kao u Tablici 1.

Tablica 1. atributi i metode klasa katastarskog modela

	KatastarskaČestica	Građevina	MeđnaTočkaČestice	PoložajTočke
atributi	broj katastarskaOpćina elaboratNastanka površina međneTočke	namjena elaboratNastanka položajneTočke pripadnostČestici	stabilizacija elaboratNastanka kvaliteta	Y X h
metode	cijepajČesticu poništiČesticu	promijeniNamjenu promijeniGeometriju	promijeniKvalitetu	promijeniKoordinat e

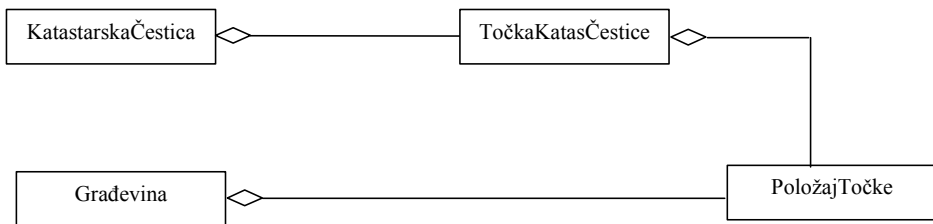
Zadržimo se prvo na klasi **KatastarskaČestica** te njenim atributima i metodama. Broj, kao jedinstveni identifikator služi za povezivanje svih tematskih podataka iz na primjer knjižnog dijela operata, ali isto tako kao eventualni identifikator pripadnosti građevine. Oznaka K.O. i elaborata su tek primjeri mnoštva mogućih atributa koji dodatno opisuju česticu i neophodni su za dinamički sustav kakav je katastarski operat.

Atributi katastarske čestice i planovi su bili spremljeni u različite sustave i jedini zajednički čimbenik im je bio korisnik. To je značajno povećalo mogućnost pogrešaka prilikom prvovođenja promjena (Tuomaala, Uimonen 1998).

Metode, implementirane na razini objekta, prikazati će nam samo ono što trebamo znati i ovisno o izboru obaviti traženu operaciju. Što se događa kada npr. cijepamo česticu.

Implementacije tražene metode će putem nekog grafičkog ili tekstualnog sučelja od nas zatražiti način cijepanja čestice. Ugrađeni, za korisnika nevidljivi, algoritmi će tada pronaći **:MednaTočkaČestice** između kojih ubacujemo nove. Novonastale objekte **:MednaTočkaČestice** i **:PoložajTočke** sustav će popuniti odgovarajućim atributima te će se pobrinuti da u promijenjenom i novonastalom (cijepanjem) objektu **:KatastarskaČestica** također budu ažurirani. Sve se ovo događa bez poticaja od strane korisnika, u potpunosti uklanjajući mogućnost pogreške.

Veza između objekata **KatastarskaČestica** i **MednaTočkaČestice** ostvarena je atributom prve koji sadrži popis tih točaka, a **MednaTočkaČestice** i **PoložajTočke** povezani su jedinstvenim identifikatorom. Klasa **Građevina** vezana je izravno na **PoložajTočke**. Dodatni atribut klase **Građevina** vezuje svaki njen objekt sa česticom s kojom dijeli podatke o vlasništvu (slika 4).



Slika 4. Grafički prikaz odnosa u modelu

6. ZAKLJUČAK

Prednosti objektnoorijentiranog modeliranja, a posebno prostornih podataka, su mnogobrojne. Priroda je sastavljena od objekata koji sami sadržavaju podatke o sebi. Sustav za upravljanje podacima o stvarnim objektima koji je temeljen na istim principima na kojima oni u prirodi postoje ima sve pretpostavke za najveću moguću učinkovitost.

Upravo objektnoorijentirani GIS sustavi povezani s bazama podataka koje funkcioniraju na istim principima pokazuju najbolje rezultate pri upravljanju svakom vrstom geoprostornih podataka. Prelaskom na takve sustave za upravljanje prostorom idemo u korak s modernim svjetskim trendovima u znanosti o upravljanju prostornim informacijama.

LITERATURA

- Bill, R., Fritsch, D. (1994): Grundlagen der Geo-Informationssysteme - Band 1 Hardware, Software und Daten, Wichmann Verlag GmbH, Heidelberg.
- Haigh, A. (2001): Object -Oriented Analysis & Design, Osborne / McGraw-Hill.
- Kaufmann, J., Steudler, D. (1998): Cadastre 2014 - A Vision for a future Cadastral System, FIG Commission 7, Working Group 7.1 (modern Cadastral Systems).
- Morton, J. (2000): An object oriented approach to parcel history tracking, URISA annual conference proceedings [CD-ROM] / Urban and Regional Information Systems Association (URISA). - Orlando (Fl.) : URISA, 2000.

- OMG (2001): Unified Modeling Language Specification - version 1.4 / September 2001, OMG, <http://www.omg.org/technology/documents/>.
- Roić, M. (1998): Improvement of Cadastre in Croatia. In "Proceedings of the XXI. International FIG Congress", Commission 7 - Cadastre and Land Management, pp. 505-516, July 19.-25., Brighton.
- Tuomaala, J., Uimonen, M. (1998): Introduction of Finland's new object-oriented cadastral information system (JAKO), XXI International Congress : developing the profession in a developing world : Commission 7 : cadastre and land management / International Federation of Surveyors. - Brighton : FIG, 1998 ; pp. 328-341.