

ZLEPKI V OBJEKTNO USMERJENEM OKOLJU

Mojca Indihar Štemberger, Janez Grad
Univerza v Ljubljani, Ekonomska fakulteta,
Kardeljeva ploščad 17, 1000 Ljubljana
mojca.stemberger@uni-lj.si, janez.grad@uni-lj.si

Povzetek

Zaradi zagotavljanja konkurenčnih prednosti organizacije vse bolj uporabljajo različna orodja za analiziranje podatkov, kot so na primer orodja za rudarjenje po podatkih. Zlepki so odsekoma polinomske funkcije, ki se uporabljajo tudi pri nekaterih tovrstnih orodjih. Razen tega se jih veliko uporablja pri računalniško podprtem geometrijskem oblikovanju. Prispevek predstavlja objektni model zlepkov, ki je neodvisen od konkretne programske rešitve in lahko služi za osnovo vsake programske rešitve, ki uporablja zlepke. Tako se zmanjšuje neskladje med tovrstnimi programskimi rešitvami in povečuje njihova združljivost. Za modeliranje sva uporabila standardni jezik za objektno modeliranje UML. Predlagava tudi shranjevanje zlepkov v bazi podatkov, ki podpira standard ODMG. Opisan je tudi prototip sistema, ki sva ga razvila v programskem jeziku Java.

Abstract

Data analysis like Data Mining is getting more and more important every day because the organizations want to get their competitive advantage. Splines are piecewise polynomial functions that are used for analysis by some Data Mining tools. Besides, they are widely used in CAGD applications for designing. The article presents the object model of splines that is independent of the concrete application but can serve as basis for any application which uses splines and reduces the gap among such applications and their incompatibility. It has been modeled by standard object-oriented modeling language UML. We also propose saving splines in an ODMG compliant database. In the article the prototype of a system that has been developed in Java is presented as well.



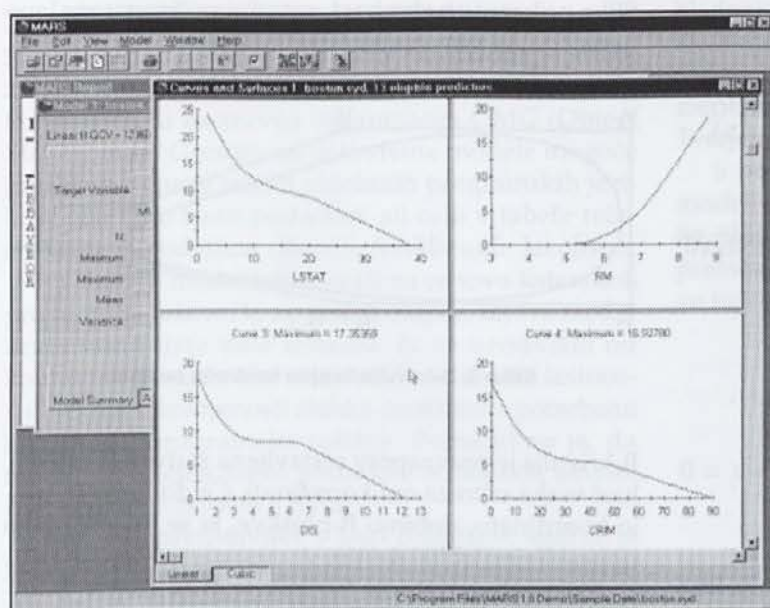
1. UVOD

Če hoče organizacija v vse težjih pogojih uspešno poslovati in si zagotavljati konkurenčne prednosti, mora s podatki upravljati kot s pomembnimi viri. To je glavni razlog vse večjega uveljavljanja različnih orodij za analiziranje podatkov, kot so na primer orodja OLAP (angl. On-Line Analytical Processing) ali pa orodja za rudarjenje po podatkih (angl. Data Mining). Ena izmed metod, ki jih orodja za rudarjenje po podatkih uporabljajo za odkrivanje povezav med velikimi količinami podatkov, je tudi aproksimacija oziroma regresija, pri čemer gre za to, da dobimo povezave med spremenljivkami v obliki matematičnih funkcij. Na ta način dobimo tudi napoved za vrednosti neke vhodne spremenljivke v točki, kjer njene vrednosti ne poznamo. Takšne metode so uporabne predvsem pri napovedovanju, ki lahko služi kot osnova za bolj kakovostno odločanje.

Razvoj na področju matematike in statistike je pokazal, da so v ta namen najprimernejše funkcije zlepki. To so odsekoma polinomske funkcije, ki omogočajo lokalnejšo aproksimacijo, v stičiščih pa gladko pre-

hajajo ena v drugo. Orodja, ki nam omogočajo aproksimacijo in s tem povezano napovedovanje, temeljijo na teoriji aproksimacije s pomočjo zlepkov (De Boor, 1978), (Friedman, 1991), (Eubank, 1999), (Pagan, Ullah, 1999). Slika 1 prikazuje orodje za rudarjenje po podatkih MARS (kratica za angl. Multivariate Adaptive Regression Splines), ki ga razvija podjetje Salford Systems.

Razen za aproksimacijo se zlepke veliko uporablja tudi na področju računalniško podprtega geometrijskega oblikovanja (angl. Computer Aided Geometric Design - CAGD). Z zlepkami so npr. v računalniku predstavljeni znaki, razen tega so del vsakega zmogljivejšega paketa za oblikovanje (npr. CorelDraw). Modernega oblikovanja si brez zlepkov pravzaprav ne moremo predstavljati. Razvoj oblikovanja z zlepkami se je začel v šestdesetih letih v Franciji in ZDA zaradi potreb avtomobilske in letalske industrije (Farin, 1997). Zlepki so na kratko opisani v razdelku 2. Bralec, ki z njimi ni seznanjen, lahko o zlepkah prebere več v navedeni literaturi.



Slika1: orodje MARS za rudarjenje po podatkih, ki uporablja zlepke.

Problem, ki ga vidiva na področju uporabe zlepkov, je neskladje med različnimi programskimi rešitvami, ki zlepke uporabljajo, in njihova nezdržljivost. Predlagava uporabo enotnega objektnega modela, ki je neodvisen od konkretne programske rešitve, sistema za upravljanje baze podatkov (SUBP) in programskega jezika. Za razvoj modela sva uporabila standardni jezik za objektno modeliranje UML (Booch, Rumbaugh, Jacobson, 1999). Razviti model zajema uporabo zlepkov tako na področju aproksimacije kot na področju računalniške grafike ali na katerem drugem področju in predstavlja osnovo tovrstne programske rešitve. Zaradi lastnosti objektnosti ga je mogoče razširiti s potrebami konkretne programske rešitve. Opisan je v razdelku 3.

Prispevek predstavlja tudi rešitev za shranjevanje zlepkov v objektni bazi podatkov (razdelek 4). Pri tem sva uporabila standard ODMG (Cattell et al., 2000), ki ga razvija istoimenska organizacija. Standard je neodvisen od uporabljenega programskega jezika in sistema za upravljanje baze podatkov. Mogoče ga je uporabiti za transparentno shranjevanje v "pravi" objektni bazi podatkov in preko posebne preslikave med objekti in relacijami tudi v relacijskih bazah podatkov.

Peti razdelek prispevka je namenjen kratki predstavitvi prototipa sistema, ki sva ga razvila s pomočjo programskega jezika Java in objektno baze podatkov Poet, šesti pa zaključku.

2. ZLEPKI

Pred razvojem, katerega rezultat so zlepki, se je za interpolacijo in aproksimacijo največ uporabljalo poli-

nome, saj je relativno enostavno izračunati njihove koeficiente. Vendar z naraščanjem števila točk, ki jih želimo interpolirati ali aproksimirati, ni vedno mogoče najti zadovoljive funkcije v razredu polinomov. Polinomi višjega reda lahko oscilirajo (Schumaker, 1993), kar pomeni, da je bilo potrebno vpeljati nov razred funkcij. Zlepki, to so odsekoma polinomske funkcije, ki v stičiščih gladko prehajajo ena v drugo, so takšen razred funkcij. Zlepki je možno izraziti kot zaporedje polinomov definiranih na posameznih odsekih ali pa v posebni obliki kot B-zlepke ali Beziereve krivulje (de Boor, 1978), (Farin, 1997), (Schumaker, 1993).

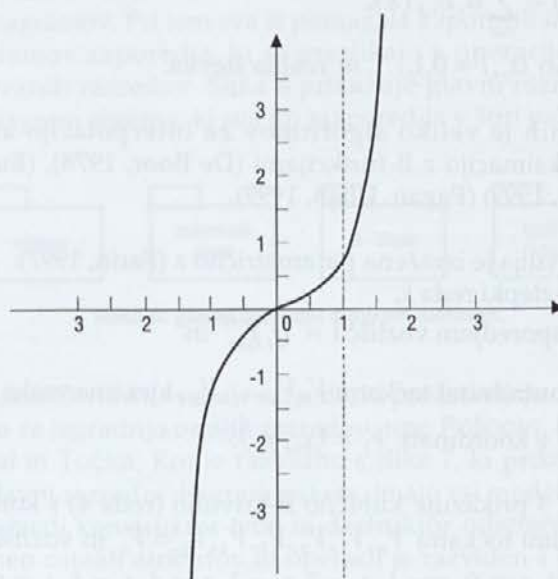
Zlepki, s katerimi se srečujemo pri orodjih za rudarjenje po podatkih, običajno sodijo v prvo skupino, saj funkcijo največkrat želimo izraziti v obliki polinomskega odseka.

Slika 2 prikazuje kubični zlepek, sestavljen iz dveh odsekov:

$$f(x) = \begin{cases} x^3; & x \leq 1 \\ 2x^3 - 3x^2 + 3x - 1; & x > 1 \end{cases}$$

Kot je razvidno s slike, je funkcija v točki $x = 1$ gladka.

B-zlepki, ki služijo za posebno predstavitev polinomskega zlepkov, imajo še dodatne prednosti, kot je npr. večja numerična stabilnost (Schumaker, 1993).



Slika 2: kubični zlepek.

Definiramo jih lahko na več načinov, najenostavneje pa s pomočjo rekurzivne formule:

$$B_{i,k}(x) = \frac{x-t_i}{t_{i+k}-t_i} B_{i,k-1}(x) + \frac{t_{i+k}-x}{t_{i+k}-t_{i+1}} B_{i+1,k-1}(x),$$

kjer je $t = (t_i)_{i=0}^{m+k}$ nepadajoče zaporedje realnih števil, ki jih imenujemo vozlišča, izraza

$$\frac{x-t_i}{t_{i+k}-t_i} B_{i,k-1}(x) \text{ in } \frac{t_{i+k}-x}{t_{i+k}-t_{i+1}} B_{i+1,k-1}(x)$$

pa imata vrednost 0 kadar je $t_{i+k-1} - t_i = 0$ ali $t_{i+k} - t_{i+1} = 0$.

B-zlepek reda k $B_{i,k}(x)$ je polinom reda k , ki ima pomembne lastnosti:

$$B_{i,k}(x) \geq 0, \text{ ko } x \in [t_i, t_{i+k}],$$

$$B_{i,k}(x) > 0, \text{ ko } x \in (t_i, t_{i+k}),$$

$$B_{i,k}(x) = 0, \text{ ko } x \notin [t_i, t_{i+k}].$$

Razlog za vpeljavo B-zlepkov je definicija B-funkcij in B-krivulj. Prve se uporablja predvsem za interpolacijo ali aproksimacijo, slednje pa pri oblikovanju.

B-funkcija reda k z zaporedjem vozlišč $t = (t_i)_{i=0}^{m+k}$ je vsaka linearna kombinacija B-zlepkov reda k :

$$f(x) = \sum_{i=0}^m \alpha_i B_{i,k}(x),$$

kjer so $\alpha_i, i = 0, 1, \dots, m$ realna števila.

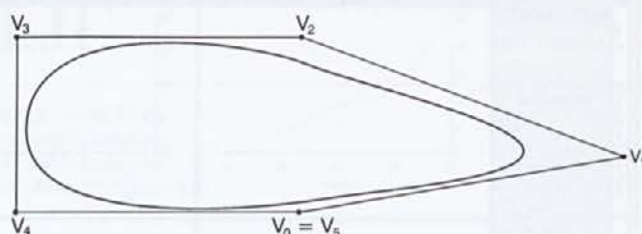
Znanih je veliko algoritmov za interpolacijo ali aproksimacijo z B-funkcijami (De Boor, 1978), (Eubank, 1999) (Pagan, Ullah, 1999).

B-krivulja je izražena parametrično z (Farin, 1997):

- B-zlepek reda k ,
- zaporedjem vozlišč $t = (t_i)_{i=0}^{m+k}$ in
- kontrolnimi točkami $V_0, V_1, \dots, V_m$, kjer ima vsaka x in y koordinati: $V_i = (x_i, y_i)$.

Slika 3 prikazuje kubično B-krivuljo (reda 4) s kontrolnimi točkami $V_0, V_1, V_2, V_3, V_4, V_5 = V_0$ in vozlišči $t_0 = -1, t_1 = 0, t_2 = 1, t_3 = 2, t_4 = 3, t_5 = 4, t_6 = 5, t_7 = 6, t_8 = 7, t_9 = 8$.

Slika prikazuje tudi njen kontrolni poligon, to je lomljena črta, ki povezuje kontrolne točke.



Slika 3: B-krivulja in njen kontrolni poligon.

B-krivulja je pravzaprav sestavljena iz dveh B-funkcij, kjer vsaka ustreza eni koordinati. Če dodamo še tretjo koordinato, dobimo B-ploskve, ki se jih prav tako uporablja pri oblikovanju.

Pomembna lastnost B-zlepkov, B-funkcij in B-krivulj je, da so zelo primerni za računalniške obdelave, saj je za njihovo rekonstrukcijo potrebno shraniti malo podatkov. Ena izmed njihovih glavnih prednosti je tudi, da sprememba enega odseka vpliva le na bližnje odseke.

Bezierove krivulje so najpogostejše uporabljene pri orodjih za računalniško podprto geometrijsko modeliranje. Pravzaprav gre za posebno obliko B-krivulj, ki sta jo v šestdesetih letih vpeljala Bezier in De Casteljau. Ime nosijo po Bezieru, saj De Casteljauovo delo ni bilo objavljeno. Definirane so kot

$$B_n[V_0, V_1, \dots, V_n; a, b; t] = \sum_{i=0}^n V_i E_i^n(t), a \leq t \leq b,$$

kjer so

$$E_i^n(t) = \binom{n}{i} \frac{(t-a)^i (b-t)^{n-i}}{(b-a)^n}, a \leq t \leq b, i = 0, 1, \dots, n$$

Bernsteinovi polinomi stopnje n , točke $V_0, V_1, \dots, V_n$ pa so kontrolne točke.

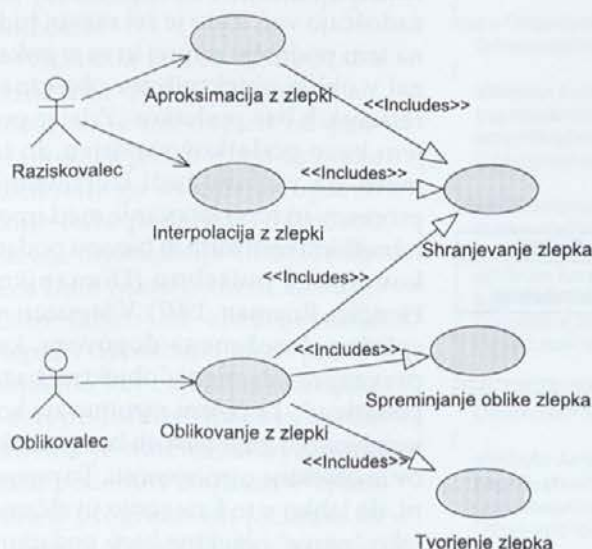
Oblikovanje s pomočjo B-krivulj ali Bezierovih krivulj (Farin, 1997) je enostavno, saj kontrolni poligon ponazarja obliko krivulje. Če premaknemo kontrolno točko, to vpliva na bližnje odseke krivulje. Operacije kot rotacija, premikanje ali spreminjanje velikosti lahko realiziramo tako, da operacijo izvedemo na kontrolnem poligonu in ponovno narišemo krivuljo. Če želene oblike ne moremo doseči s premikanjem kontrolnih točk, krivulji dodamo nove kontrolne točke s pomočjo delitvenih algoritmov (Farin, 1997), (Indihar, 1992), katerih namen je, da obstoječo krivuljo opišemo z večjim številom kontrolnih točk in tako dobimo več možnosti za spreminjanje njene oblike.

3. OBJEKTNI MODEL SISTEMA

Sistem sva modelirala s pomočjo standardnega jezika za objektno modeliranje UML (Unified Modeling Language), ki ga razvija organizacija OMG (Object Management Group), saj je tovrstne modele mogoče preslikati v enega izmed objektnih programskih jezikov, objektno bazo podatkov ali celo v tabele relacijske baze podatkov (Booch, Rumbaugh, Jacobson, 1999). Razviti model lahko služi za osnovo katerekoli programske rešitve, ki uporablja zlepke, saj sva modelirala samo tiste dele sistema, ki so neodvisni od konkretne programske rešitve. Model zaradi lastnosti objektno usmerjenosti zlahka razširimo s potrebami konkretne programske rešitve. Pomembno je, da model ne zajema samo podatkov o zlepkih, ampak tudi operacije za delo z njimi.

Ker je za predstavitev v bazi podatkov, tudi če je le-ta objektna, najpomembnejši statični vidik sistema, sva največ pozornosti posvetila razrednim diagramom. Te diagrame navajajo tudi v literaturi kot najpomembnejše pri modeliranju statičnega pogleda na sistem (Booch, Rumbaugh, Jacobson, 1999), (Blaha, Premierlani, 1998). Kljub temu pa sva predstavila tudi del dinamike sistema, ki vključuje zlepke. Predstavila sva jo z diagrami primerov uporabe in diagrami zaporedja, ki sestavljajo pogled na uporabniške zahteve. S pomočjo teh diagramov sva lažje določila potrebne operacije modeliranih razredov.

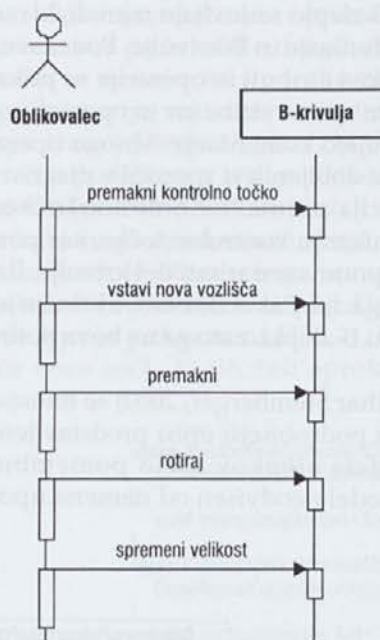
Slika 4 prikazuje diagram primerov uporabe sistema. Kot vidimo, lahko imamo dve vrsti uporabnika sistema, ki sta prikazana kot akterja. To sta Raziskovalec in Oblikovalec. Prvi je v interakciji s primeroma uporabe Aproksimacija z zlepkom in Interpolacija z zlepkom. Drugi pa s primeroma uporabe Oblikovanje z zlepkom in Tvorjenje zlepka.



Slika 4: Diagram primerov uporabe sistema.

ki, drugi pa s primerom uporabe Oblikovanje z zlepkom. Vsi trije omenjeni primeri uporabe vsebujejo (oziroma uporabljajo) primer uporabe Shranjevanje zlepkov, tretji pa še Spreminjanje oblike zlepka in Tvorjenje zlepka.

S pomočjo diagramov zaporedja sva podrobneje modelirala primer uporabe Spreminjanje oblike zlepka. Slika 5 prikazuje diagram zaporedja za B-krivulje, podoben diagram pa lahko narišemo tudi za Beziereve krivulje. Nahaja se v (Indihar Štemberger, 2000).



Slika 5: diagram zaporedja za B-krivulje.

Strukturo sistema sva modelirala s pomočjo razrednih diagramov. Pri tem sva si pomagala s sporočili iz diagramov zaporedja, ki se preslikajo v operacije ustreznih razredov. Slika 6 prikazuje glavni razredni diagram sistema, ki sva jih razporedila v štiri pakete.



Slika 6: glavni razredni diagram sistema.

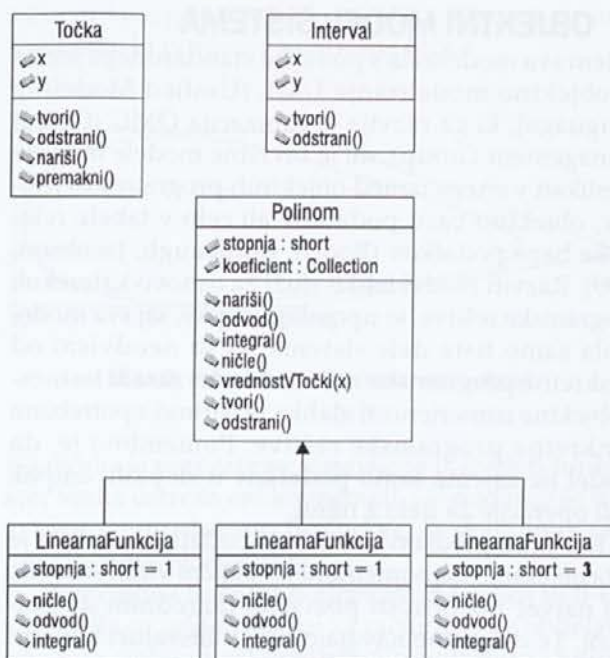
Paket Osnovni vsebuje razrede, ki predstavljajo osnovo za izgradnjo ostalih razredov, npr. Polinom, Interval in Točka. Kot je razvidno s slike 7, ki prikazuje glavni razredni diagram paketa, imajo vsi modelirani razredi konstruktor tvori in destruktor odstrani. Pomen ostalih atributov in operacij je razviden s slike. Paket vsebuje tudi razrede LinearnaFunkcija, KvadratnaFunkcija in KubičnaFunkcija, ki so najpogostejše

v rabi. Ti razredi podedujejo atribute in operacije od razreda **Polinom**, vendar so nekatere enostavnejše.

Paket **Polinomski** zlepki vsebuje razrede, ki so potrebni za predstavitev odsekoma polinomskih funkcij. Osrednji razred paketa se imenuje **OdsekomaPolinomskaFunkcija**, modeliranih pa je še več razredov, ki so vsi nasledniki tega razreda. Podrobnosti v zvezi z atributi in operacijami so razvidne s slike 8, ki prikazuje glavni razredni diagram paketa. Vsi razredi v paketu so zbirke polinomov, kjer je zbirka lahko seznam, polje ali drugega tipa.

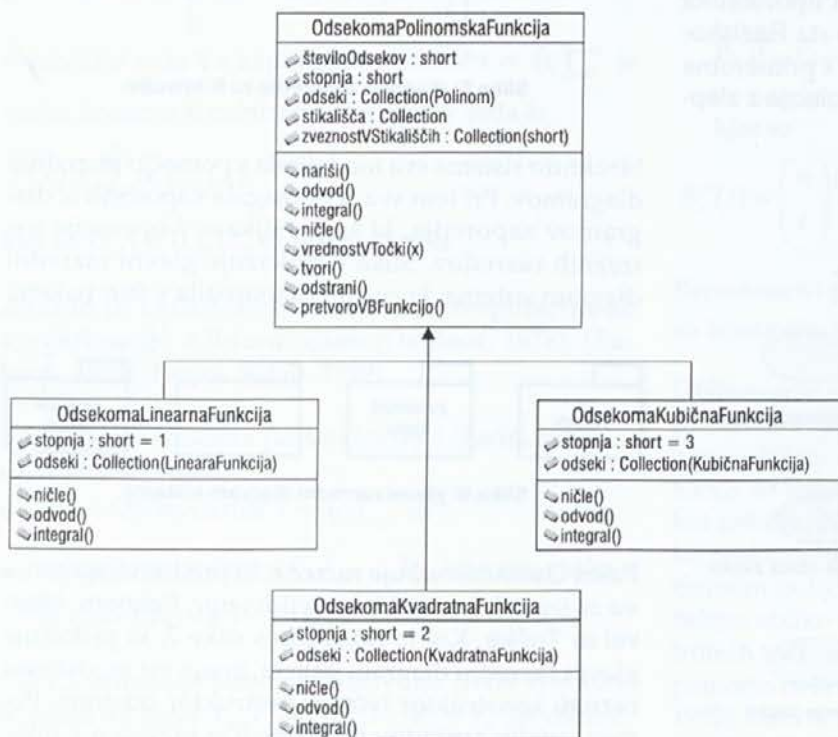
Paket **B-zlepki** sestavljajo razredi, ki modelirajo B-zlepke, B-funkcije in B-krivulje. Povezave med razredi, ter njihovi atributi in operacije so prikazani na sliki 9. Pomen večine atributov in operacij je očiten, zato ne potrebujejo komentarja. Mnogo operacij razreda **BKrivulja** je dobljenih iz sporočil v diagramu zaporedja. Operacija **premakniKontrolnoTočko** na primer služi premikanju kontrolne točke, kar povzroči, da je potrebno ponovno narisati del krivulje. Razrede povezuje agregacija. Paket **Bezierve** krivulje je zelo podoben paketu **B-zlepki**, zato ga ne bova podrobneje opisala.

V (Indihar Štemberger, 2000) se nahaja še več diagramov in podrobnejši opisi predstavljenega objektnega modela zlepkov. Zelo pomembno je, da je opisani model neodvisen od namena uporabe zlepkov



Slika 7: razredni diagram paketa Osnovni.

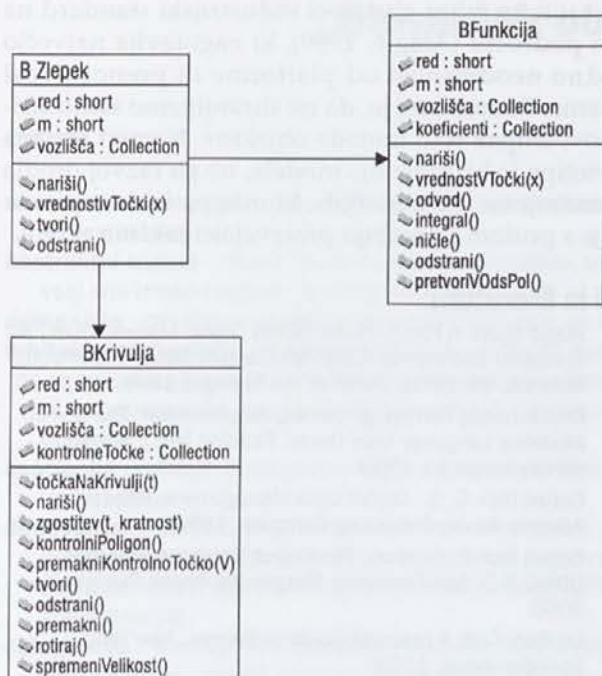
ov in lahko služi kot osnova sistema, kjer se zlepki uporabljajo za aproksimacijo ali interpolacijo ali pa sistema, ki omogoča oblikovanje z zlepki.



Slika 8: razredni diagram paketa Polinomski zlepki.

4. SHRANJEVANJE ZLEPKOV

Pri sodobnih informacijskih sistemih se srečujemo z vse kompleksnejšimi objekti, za katerih shranjevanje klasične relacijske baze podatkov ne zadoščajo več. Zato je šel razvoj tudi na tem področju naprej in se je pokazal v obliki objektnih ter objektno-relacijskih baz podatkov. Zdaj je pojem baze podatkov razširjen do te mere, da vsebuje tudi izvrševanje procesov in torej ločevanje med uporabniškimi rešitvami in bazami podatkov ni več potrebno (Domanjko, Heričko, Rozman, 1997). V literaturi ni splošno doseženega dogovora, kaj pravzaprav pomeni "objektna baza podatkov". Ta pojem razumevajo kot kombinacijo zmogljivosti baz podatkov in objektne usmerjenosti. To pomeni, da lahko v to kategorijo uvrščamo tako "prave" objektne baze podatkov kot objektne razširitve sistemov za upravljanje z relacijskimi bazami podatkov, kot je na primer Oracle.



Slika 9: razredni diagram paketa B-zlepki.

Za shranjevanje kompleksnih objektov, s katerimi se srečamo tudi pri programskih rešitvah, ki uporabljajo zlepke, so objektne baze podatkov prava rešitev. Na področju računalniške grafike se za shranjevanje nekaterih posebnih primerov zlepkov že uporabljajo objektne baze podatkov (Cattell, 1994), ne uporablja pa se jih pri orodjih za rudarjenje po podatkih.

Odločila sva se za uporabo standarda ODMG (Cattell et al., 2000), ki je neodvisen od programskega jezika in od uporabe konkretnega sistema za upravljanje baze podatkov. Standard razvija organizacija ODMG (Object Data Management Group). Zato lahko za shranjevanje zlepkov uporabimo tako "pravo" objektno kot objektno-relacijsko bazo podatkov. Po standardu ODMG je možno načrt sheme baze podatkov napisati v enem izmed programskih jezikov, za katerega obstaja jezikovna povezava (C++, Smalltalk in Java) ali v neodvisnem jeziku za definicijo objektov (angl. Object

Definition Language - ODL). Odločila sva se za slednjo možnost. Slika 10 prikazuje definicijo sheme za polinomske zlepke. Definicija sheme za vse ostale razrede, ki so del sistema, se nahaja v (Indihar Štemberger, 2000).

5. PROTOTIP SISTEMA

Izdelala sva tudi prototip sistema, za izgradnjo katerega sva uporabila obetavni programski jezik Java, ki se vse bolj uveljavlja. Prednost tega jezika je predvsem neodvisnost delovanja od strojne opreme in operacijskega sistema, ima pa še veliko drugih dobrih lastnosti. Prototip se nanaša na uporabo zlepkov na področju aproksimacije, saj so ravno na tem področju potrebe po shranjevanju zlepkov, ki jih kreirajo orodja za analizo podatkov, v bazi podatkov največje. Uporabniku omogoča vnos točk, ki jih želi aproksimirati, iskanje kubičnega aproksimacijskega zlepka ter njegovo shranjevanje. Razen tega je možno tudi ponovno prikazati že shranjene zlepke. Uporabniški vmesnik prototipa je prikazan na sliki 11. Uporabniku omogoča vnos točk, ki jih želi aproksimirati s

```

class OdsekomaPolinomskaFunkcija
{
    attribute short stopnja;
    attribute short številoOdsekov;
    attribute Array<Polinom> odseki;
    attribute array<float> stikališča;
    void tvori();
    void odstrani() raises(ni_funkcije);
    void nariši();
    float vrednostVTočki(in float x) raises(ni_definirana);
    bag<float> ničle();
    OdsekomaPolinomskaFunkcija odvod();
    OdsekomaPolinomskaFunkcija integral();
    BFunkcija pretvoriVBFunkcijo();
}

class OdsekomaLinearnaFunkcija extends
OdsekomaPolinomskaFunkcija
{
    attribute Array<LinearnaFunkcija> odsek;
    bag<float> ničle();
    array<float> odvod();
    OdsekomaKvadratnaFunkcija integral();
}

class OdsekomaKvadratnaFunkcija extends
OdsekomaPolinomskaFunkcija
{
    attribute Array<KvadratnaFunkcija> odsek;
    bag<float> ničle();
    OdsekomaLinearnaFunkcija odvod();
    OdsekomaKubičnaFunkcija integral();
}

class OdsekomaKubičnaFunkcija extends
OdsekomaPolinomskaFunkcija
{
    attribute Array<KubičnaFunkcija> odsek;
    bag<float> ničle();
    OdsekomaKvadratnaFunkcija odvod();
    OdsekomaPolinomskaFunkcija integral();
}

class KubičniInterpolacijskiZlepek extends
OdsekomaKubičnaFunkcija
{
    void interpoliraj(in Set<Točka> točke);
}

class KubičniAproksimacijskiZlepek extends
OdsekomaKubičnaFunkcija
{
    void aproksimiraj(in Set<Točka> točke);
}

class NaravniKubičniZlepek extends
KubičniInterpolacijskiZlepek
{
    void interpoliraj(in Set<Točka> točke);
}

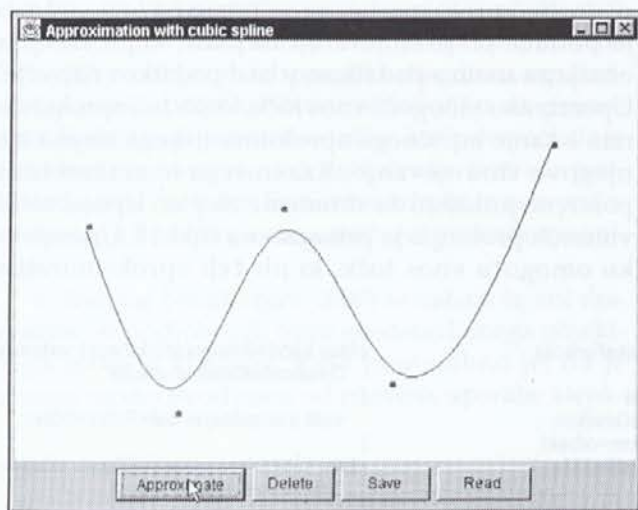
class KubičniZlepekZOdvod extends
KubičniInterpolacijskiZlepek
{
    void interpoliraj(in Set<Točka> točke;
in float odvZ, in float odvK);
}

class PeriodičniKubičniZlepek extends
KubičniInterpolacijskiZlepek
{
    void interpoliraj(in Set<Točka> točke);
}
  
```

Slika 10: definicija sheme za razrede OdsekomaPolinomskaFunkcija, OdsekomaLinearnaFunkcija, OdsekomaKvadratnaFunkcija in OdsekomaKubičnaFunkcija.

kubičnim zlepkom, iskanje aproksimacijskega zlepka po eni izmed metod opisanih v (Eubank, 1999), njegovo shranjevanje in branje shranjenih zlepkov. Razred, ki je rezultat aproksimacije, ustreza razredu *OdsekomaKubičnaFunkcija* in je naslednik tega razreda, saj je opremljen še z dodatnimi operacijami.

Odločila sva se za shranjevanje objektov v objektni bazi Poet, vendar bi bilo zaradi neodvisnosti razvitega modela od platforme, ki sva jo dosegla z uporabo standarda ODMG, mogoče uporabiti katerikoli SUBP, ki ima povezavo s tem standardom. Preko preslikave objektov v relacije (na primer Java Blend) bi bilo mogoče uporabiti tudi relacijsko bazo podatkov.



Slika 11: uporabniški vmesnik prototipa.

6. ZAKLJUČEK

Meniva, da predstavljeni model lahko služi kot "skupni imenovalec" sistema, ki uporablja zlepke. Na njem lahko temelji vsaka tovrstna programska rešitev, ki ponavadi model razširja z dodatnimi razredi, ki pa so nasledniki modeliranih razredov. Prepričana sva, da je izbira jezika UML pravilna, saj je primeren za veliko okolij. Uporaba standarda ODMG, pri razvoju katerega sodeluje večina proizvajalcev sistemov za upravljanje objektnih baz podatkov, je prav tako pravil-

na, saj je to edini obstoječi industrijski standard na tem področju (Alagič, 1999), ki zagotavlja največjo možno neodvisnost od platforme in prenosljivost sistema. Pomembno je, da ne shranjujemo samo atributov, ampak tudi metode objektov. Namen razvoja prototipa je bil preizkus modela, ne pa razvoj orodja za rudarjenje po podatkih. Morda pa lahko opisane ideje s pridom uporabijo proizvajalci takšnih orodij.

Viri in literatura

1. Alagič Suad: A Family of the ODMG Object Models, Third East European Conference ADBIS'99 (Lecture Notes in Computer Science, Vol 1691), Berlin et al.: Springer, 1999, str. 14-30.
2. Booch Grady, Rumbaugh James, Jacobson Ivar: The Unified Modeling Language User Guide. Reading [etc.]: Addison Wesley Longman, 1999.
3. Cattell Rick G. G.: Object Data Management. Reading: Addison-Wesley Publishing Company, 1994.
4. Cattell Rick G. G. et al.: The Object Database Standard ODMG 3.0. San Francisco: Morgan Kaufmann Publishers, 2000.
5. De Boor Carl: A practical Guide to Splines. New York: Springer-Verlag, 1978.
6. Domanjko Tomaž, Hričko Marjan, Rozman Ivan: Uporaba objektnih podatkovnih baz, COTL, 1 (1997), 2, [URL: <http://lisa.uni-mb.si/cot/cot/april97/index.shtml>].
7. Eubank Randall L.: Nonparametric Regression and Spline Smoothing (Statistics, Textbooks and Monographs, V. 157), Marcel Dekker, 1999.
8. Farin Gerald: Curves and Surfaces for CAGD. Boston: Academic Press, 1997.
9. Friedman J.H.: Multivariate Adaptive Regression Splines. Annals of Statistics, 19 (1991), str. 1-141.
10. Indihar Štemberger Mojca: Zlepki v okolju objektnih baz podatkov, Doktorska disertacija, Ljubljana: Univerza v Ljubljani, Ekonomska fakulteta, 2000.
11. Kim Won, urednik: Modern Database systems – The Object Model, Interoperability and Beyond. New York: Addison-Wesley Publishing Company, 1995.
12. Khoshafian Setrag, Abnous Razmik: Object Orientation. New York: John Wiley & Sons, 1995.
13. Object Data Management Group, [URL: <http://www.odmg.org/>].
14. Object Management Group, [URL: <http://www.omg.org/>].
15. Pagan Adrian, Ullah Aman: Nonparametric Econometrics, Cambridge: Cambridge University Press, 1999.
16. Poet, [URL: <http://www.poet.com/>].
17. Salford Systems, [URL: <http://www.salford-systems.com/>].
18. Schumaker Larry L.: Spline Functions: Basic Theory. Malabar, Florida: Krieger Publishing Company, 1993.

Dr. Mojca Indihar Štemberger je po osnovni izobrazbi matematik. Leta 1996 je magistrirala na Fakulteti za računalništvo in informatiko, leta 2000 pa doktorirala iz informacijsko upravljalskih ved na Ekonomski fakulteti, kjer je tudi zaposlena kot asistentka za informatiko. Ukvarja se s poslovno informatiko, optimizacijo in bazami podatkov.

Dr. Janez Grad je magistriral iz matematike na Univerzi v Birminghamu, Anglija, leta 1973 pa doktoriral iz matematičnih znanosti na Vseučilišču v Zagrebu. Od leta 1973 sodeluje kot učitelj za informatiko na Ekonomski fakulteti, najprej kot docent, od leta 1979 dalje kot izredni profesor, od 1985 pa kot redni profesor. Ukvarjal se je s programiranjem na računalniku in z reševanjem problema lastnih vrednosti in vektorjev matrik, v zadnjih letih pa se ukvarja z reševanjem problemov s področja operacijskega raziskovanja in s področja baz podatkov.