

Uporaba standarda LandXML in razširitev PmcXML

Matej Gomboši, Borut Žalik

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, Smetanova 17, 2000 Maribor

Danijel Rebolj

Fakulteta za gradbeništvo, Univerza v Mariboru, Smetanova 17, 2000 Maribor

matej.gombosi@uni-mb.si, zalik@uni-mb.si, rebolj@uni-mb.si

Izvleček

Standard LandXML predstavlja razširljiv jezik XML za potrebe informatizacije na področjih gradbenega inženirstva in izmenjave podatkov predvsem v procesih gradbene in transportne industrije. Omogoča prenos podatkov med ponudniki in uporabniki ter ponuja podatkovni format za dolgoročno shranjevanje ter standardni format za elektronsko posredovano načrtovanje. LandXML omogoča opis prostorskih elementov, kot so cesta, parcela, model terena, ter njihove spremljajoče infrastrukture. Za naše potrebe smo shemo standarda LandXML razširili z dodatnimi elementi in tako dobili razširjeno shemo PmcXML.

Abstract

The Usage of Standard LandXML and the Extention PmcXML

Standard LandXML represents an extensible XML language. It was designed for informatization of civil engineering and survey data used in construction and transportation industries. It enables transfer of engineering data between suppliers and users and provides a format suitable for long term archiving and a standard format for electronic exchange of design data. LandXML enables description of spatial elements as roads, parcels, digital terrain models including accompanying infrastructure. We extended LandXML specification with additional elements to obtain a new specification named PmcXML.

1 Uvod

LandXML je zasnoval Autodesk decembra 1999 kot industrijsko podprti odprti standard XML za izmenjavo prostorskih podatkov (Kvamme 1997). S tem se je približal potrebam gradbenikov, geodetov, razvijalcev programske in strojne opreme in ponudnikov storitev s področja geografskih informacijskih sistemov (GIS). Prva shema standarda LandXML je bila izpeljana iz predhodnega standarda EAS-E (Engineering and Surveying - Exchange) za izmenjavo podatkov v formatu ASCII.

Strukture podatkov v standardu LandXML omogočajo:

- prenos prostorskih podatkov in meritev med izvajalci in uporabniki,
- podatkovni format za dolgotrajno shranjevanje podatkov,
- standardni format za uradno elektronsko poslovanje.

Podatki LandXML se lahko uporabljajo tudi kot (LandXML 2002):

- osnova za ocenitev stroškov in zagona projektov,
- osnova za izračune in poročila po meri uporabnika,
- izmenjava podatkov iz merilnih naprav,

- format za iskanje in dodajanje informacij v baze GIS,
- standard za prenos inženirskih podatkov med različnimi aplikacijami.

V letu 2002 je precej računalniških programov (Microsoft Office, AutoCAD) ter baz podatkov (Microsoft SQL 2000, IBM DBMS in Oracle) privzelo standard LandXML in s tem tudi vse podprte značilnosti.

LandXML je specializiran XML format (LandXML 2002) za prostorske podatke, ki se na ta način lahko uporabljajo na različne načine v poslovnih in tehničnih aplikacijah in bazah podatkov, ki podpirajo XML.

V prispevku bomo opisali uporabo standarda LandXML in njeno razširitev za opis modela ceste.

2 Osnovna shema LandXML

LandXML v svoji najnovejši verziji temelji na shemi standarda XML konzorcija W3C (World Wide Web Consortium). Podatki XML so organizirani z uporabo sheme XML (Marchal 2000), ki je standard W3C za opis podatkovnih struktur in tipov podatkov v formatu XML (Birbeck 2001, St. Laurent 1998).

LandXML opisuje prostorske podatkovne strukture kot so merski sistem, podatki o točkah, geometrija cest, geometrija parcel, podatki meritev na terenu in podobno. Spodnji primer prikazuje preprost dokument LandXML, ki opisuje parcelo.

```
<LandXML version="1.0" date="2001-02-08" time="08:46:45">
  <Project name="subdivision"/>
  <Units>
<Imperial linearUnit="foot" areaUnit="squareFoot"
volumeUnit="cubicYard"/>
  </Units>
<Application name="Land Development Desktop"
manufacturer="Autodesk">
  <Author createdBy="Good Boy"/>
</Application>
  <Parcels>
    <Parcel name="Lot #16" area="65788.267">
      <Center>3144.170572 2097.339412</Center>
      <CoordGeom>
        <Line length="142.8500">
          <Start>3221.037433 2253.014210</Start>
          <End>3297.983320 2132.660352</End>
        </Line>
        <Curve rot="cw" radius="280.0000" length="61.3300">
          <Start>2987.138471 1988.824807</Start>
          <Center>2707.150400 1986.240192</Center>
          <End>2979.887278 2049.601027</End>
        </Curve>
      </CoordGeom>
    </Parcel>
  </Parcels>
</LandXML>
```

V tem primeru lahko enostavno najdemo podatke o projektu, uporabljene merske enote, informacije o programu, ki je ustvaril podatke, in podatke o parceli.

Vendar pa LandXML ne opisuje samo točk. Tabela 1 prikazuje podatkovni model sheme LandXML na najvišjem nivoju:

V tabeli 1 in v prikazanem primeru vidimo, da LandXML ne določa, kako naj točke, črte, parcele ali drugi elementi izgledajo na prikazovalniku. Podatke prikažemo s pomočjo aplikacije, ki jo uporabljamo.

LandXML vključuje vse prednosti standarda XML:

- standard podpira W3C,
- postaja standardni metajezik za prenos podatkov v računalniški industriji,
- je objektno orientiran in s tem podpira aktualne programske koncepte,
- je čitljiv in
- je razširljiv.

Z uporabo XML za opis LandXML dobimo samostojne podatke, ki ne potrebujejo dodatnega opisa. To pomeni, da so sami podatki uporabni tudi zunaj konteksta aplikacije, ki jih je kreirala. To omogoča

Projekt	Ime in opis projekta
CoordinateSystem	Kartezijski ali georeferenčni koordinatni sistem
Units	Dolžinske, kotne in ploščinske enote
CgPoints	Zbirka točk geometrije
Survey	Zbirka podatkov in parametrov iz meritev
Parcels	Parcele
Alignments	Zbirka podatkov za cestne osi
PipeNetworks	Zbirka podatkov o omrežjih
CoordGeom	Urejen seznam geometrije (črte, krivulje, spirale)
CrossSect	Prečni prerezi
PlanFeatures	Druge značilnosti načrta (ograje, drevesa, svetilke)
Surfaces	Elementi za opis modela površja

Tabela 1: Shema LandXML na najvišjem nivoju

dobra strukturiranost in preglednost XML jezika. Tako se podatki v drugih aplikacijah lahko uporabijo in prikažejo na različne načine.

3 PmcXML

Pri računalniškem opisovanju cest bi radi pogosto opisali ne samo osnovne podatke o cesti, ampak tudi podatke o spremljajočih objektih, kot so zidovi, mulde, nosilci ipd. Za opis cest in pripadajočih objektov smo v sodelovanju s Centrom za gradbeno informatiko na Fakulteti za gradbeništvo Univerze v Mariboru razširili shemo standarda LandXML in jo poimenovali PmcXML.

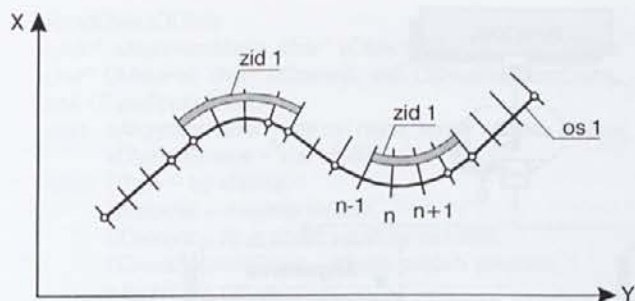
3.1 Predstavitev modela

Shema PmcXML predstavlja nadgradnjo sheme LandXML. V obstoječi shemi LandXML namreč ni strukture, ki bi celostno opisovala 3D elemente ceste. Pri tem mislimo na objekte, ki se vzdolž ceste pojavljajo v večjih prečnih presekih in so v LandXML predstavljeni kot množica nepovezanih elementov prečnih presekov. Slika 1 kaže primer ceste in dveh sten (wall), ki sta postavljeni vzdolž osi ceste (axis).

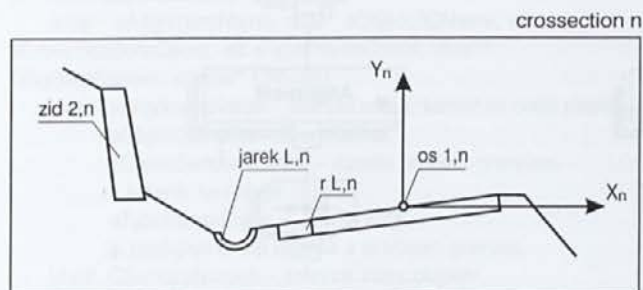
Vidimo, da se obe steni pojavljata v več prečnih prerezih. Objekt predstavimo kot množico povezanih elementov prečnih presekov. Slika 2 kaže primer prečne prereza na poziciji n iz slike 1.

Za opis novih struktur v shemi PmcXML smo morali uvesti nekaj novih pojmov v shemo LandXML:

- *Novi element Object3D*. Ta predstavlja posamezni 3D element ceste.



Slika 1: Tloris cestnega odseka



Slika 2: Prečni prerez cestnega odseka

- *Novi element Object3DType.* Ker so elementi tipizirani, jih lahko s pomočjo vnaprej definiranih tipov lažje klasificiramo. S tem tudi enostavno opišemo podatke o posameznem tipu objekta.
- *Integracija novih elementov v shemo LandXML.* Nove elemente je bilo treba skladno s pravili LandXML uvrstiti na ustrezna mesta v obstoječo strukturo, da bi omogočili čim boljše in enostavnejšo integracijo.

Poglejmo sedaj naštetje spremembe v zgradbi sheme LandXML. Na sliki 3 sta prikazani obe shemi (levo LandXML, desno PmcXML). Obe shemi sta okrnjeni samo na osnovne elemente, koristne za ponazoritev razlik. Spremembe smo izvedli tako, da je nadgradnja LandXML v OmcXML čim bolj smiselna.

Na sliki smo prikazali samo podatke o osi ceste (Alignment) in podrejenih podatkih, ker le v tej strukturi pride do sprememb. Kot je razvidno iz slike, smo na najvišjem nivoju dodali element Object3DTypes, ki predstavlja seznam za elemente Object3DType. Ta element se lahko uporablja v vseh strukturah PmcXML, zato je zaradi dostopa uvrščen na vrh sheme. To pomeni, da se lahko elementi določenega tipa pojavljajo v različnih oseh ceste.

```
<PmcXML>
...
<Object3DTypes>
  <Object3DType name="tip 1" points="10"/>
  <Object3DType name="tip 2" points="6"/>
  <Object3DType name="tip 3" points="5"/>
</Object3DTypes>
...
</PmcXML>
```

Vsak Object3DType ima svoj identifikator, prek katerega se Object3D sklicuje nanj. Ta identifikator je shranjen v obliki parametra v enem in drugem elementu in nam predstavlja glavni ključ za povezovanje elementov Object3D in Object3DType.

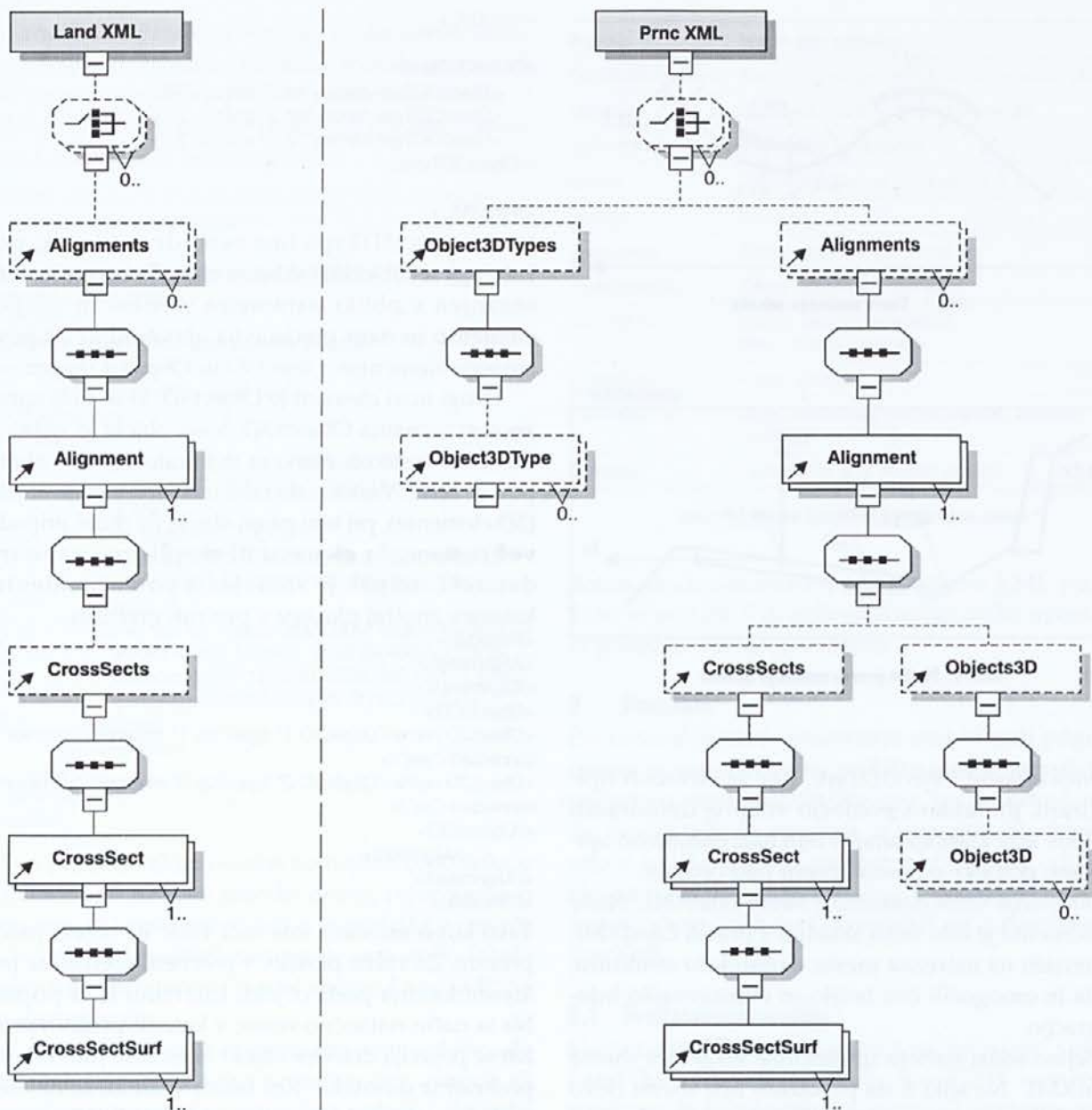
Drugi novi element je Object3D, ki se tudi nahaja znotraj seznama Objects3D. Vsak objekt je vezan na določeno cesto oz. njeno os in je zato dodan v shemo pod os ceste. Vsaka cesta tako ima lahko svoje objekte (3D elemente), pri tem pa en objekt ne more pripadati več cestam. Ta element ni eksplicitno zapisan v datoteki, ampak je shranjen s pomočjo identifikatorjev znotraj ploskev v prečnih prerezi.

```
<PmcXML>
<Alignments>
<Alignment>
<Objects3D>
<Object3D name="Objekt3D 1" type="tip 1" material="concrete"
connected="yes"/>
<Object3D name="Objekt3D 2" type="tip 3" material="aluminium"
connected="no"/>
</Objects3D>
</Alignment>
</Alignments>
</PmcXML>
```

Tako kot cesta sama ima tudi vsak 3D objekt prečne prereze. Za vsako ploskev v prečnem prerezu se prek identifikatorja poda objekt, kateremu le-ta pripada. Na ta način natančno vemo, v katerih prečnih prerezi se pojavlja določen objekt in ga zato tudi ni težko prebrati iz datoteke. Kot lahko vidimo, se dejanski podatki o objektih shranijo v enem najnižjih nivojev sheme PmcXML.

3.2 Implementacija

Za zapis podatkov v PmcXML smo napisali funkcije, ki sprejmejo ustrezne podatke o cestnem modelu in jih sproti zapisujejo v strukturo PmcXML. Ko so vsi podatki vnešeni, se zapišejo v datoteko XML. Celotna implementacija je bila izvedena v okolju Windows s programskim orodjem Visual C++ NET. Program je shranjen v knjižnici DLL, ki jo je mogoče uporabljati tudi v drugih programskih okoljih (Delphi, Visual Basic, C++ Builder).



Slika 3: Primerjava LandXML (levo) in PmcXML (desno) sheme

Opise bomo podali v obliki, kot se pojavljajo v programu, torej v C++ deklaracijah. Uporabljene so naslednje funkcije:

- AddObject3DType**
 (char* aObject3DTypeName, int aNumOfPoints)
 vhod: aObject3DTypeName – ime oz. id tipa,
 aNumOfPoints – število točk, ki opisujejo tip.

Funkcija ustvari in doda nov tip objekta.

- AddObject3D**
 (char* aAlignmentName, char* aObject3DName, char* aObject3DType, char* aMaterialType, bool aAlignmentConnection)
 vhod: aAlignmentName – ime osi ceste, kamor se doda objekt,
 aObject3Dname – ime oz. id objekta,
 aObject3DType – ime oz. id tipa objekta,
 aMaterialType – material objekta,
 aAlignmentConnection – ali je objekt vezan na os.

Funkcija ustvari nov objekt in ga doda v ustrezno os ceste.

- **ReadObject3DInfo**
(char^{*} aAlignmentName, char^{*} aObject3DName, char^{*} OType, char^{*} OMaterial, char^{*} OConnect, int& OCrossSectionsCount, int& OTypePointsCount)
vhod: aAlignmentName – ime osi ceste, kamor se doda ploskev, aObject3Dname – ime objekta.
izhod: OType – tip objekta, OMaterial – material objekta, OConnect – ali je objekt vezan na os ceste, OCrossSectionsCount – število prečnih prerezov, v katerih se nahaja objekt, OTypePointsCount – število točk, ki opisujejo ta tip objekta v prečnem prerezu.

Funkcija prebere informacije o objektu.

- **ReadObject3DData**
(char^{*} aAlignmentName, char^{*} aObject3DName, int aCrossSectionsCount, int aTypePointsCount, char^{**} OSurfaceNames, double^{*} OPoints)
vhod: aAlignmentName – ime osi ceste, kamor se doda ploskev, aObject3Dname – ime objekta, aCrossSectionsCount – število prečnih prerezov, v katerih se nahaja objekt, aTypePointsCount – število točk, ki opisujejo ta tip objekta v prečnem prerezu.
izhod: OSurfaceNames – seznam imen ploskev, ki tvorijo objekt, OPoints – seznam točk objekta.

Funkcija prebere dejanske podatke o objektu. Imena ploskev so urejena glede na stacionažo prečnih prerezov. Prav tako so urejene tudi točke.

- **ReadObjects3DCount**
(char^{*} aAlignmentName)
Funkcija vrne število objektov v določeni osi ceste.
- **ReadObject3DNames**
(char^{*} aAlignmentName, int aLength, char^{**} OObjects3DNames)
vhod: aAlignmentName – ime osi ceste, kjer iščemo objekte, aLength – število objektov,
izhod: OObjects3DNames – seznam imen objektov.

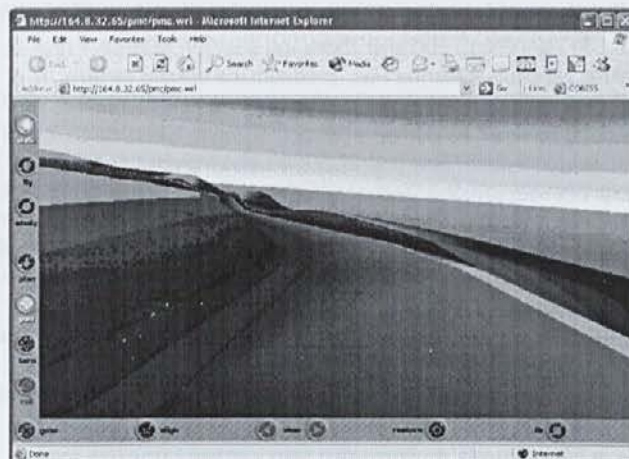
Funkcija prebere imena objektov v določeni osi ceste.

- **ReadObject3DTypesCount()**
Funkcija vrne število definiranih tipov objektov.
- **ReadObject3DTypesData**
(char^{**} OTypeNames, int^{*} OPointsCount)
izhod: OTypeNames – polje nizov imen, OPointsCount – polje števil točk.

Funkcija prebere imena in število točk, ki definirajo posamezni tip objekta.

4 Zaključek

V prispevku smo predstavili standard LandXML, ki predstavlja razširljiv jezik XML za potrebe gradbenega inženirstva in izmenjave podatkov. Videli smo, kako je mogoče opis elementov, potrebnih v gradbeni informatiki, spremeniti in dodati nove elemente, ki nam omogočajo popolnejši opis, s tem pa tudi višjo stopnjo interpretacije podatkov (primer prikazuje slika 4).



Slika 4: **Avtomatsko generirani 3D model cestnega telesa**

5 Literatura

- [1] Birbeck, M. et al. (2001). Professional XML 2nd edition, Wrox Press, UK.
- [2] Kvamme, K. (1997). Geografski informacijski sistemi, Znanstveno raziskovalni center SAZU, Ljubljana.
- [3] LandXML shema 1.0 (2002). <http://www.landxml.org>.
- [4] Marchal, B. (2000). XML by example, Que Programming, US.
- [5] St. Laurent, S. (1998). XML a primer, MIS Press, US.

Mag. Matej Gomboši je asistent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Diplomiral je leta 1999 in magistriral leta 2002. Kot asistent se od leta 1999 ukvarja z algoritmi računalniške geometrije in uporabo teh v sistemih GIS.

Dr. Borut Žalik je vodja laboratorija za geometrijsko modeliranje in algoritme multimedijev na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Doktoriral je leta 1993 na takratni Tehniški fakulteti v Mariboru. Na raziskovalnem področju se ukvarja z geometrijskim modeliranjem, računalniško geometrijo, sistemi GIS in tehnikami stiskanja podatkov.

Dr. Danijel Rebolj je diplomiral na področju gradbeništva in magistriral na področju računalništva in informatike. Doktorat tehniških znanosti je dosegel na Tehniški univerzi v Gradcu s področja gradbene informatike, s katero se ukvarja na znanstvenem in izobraževalnem področju. Od leta 1995 je na Fakulteti za gradbeništvo predstojnik Centra za gradbeno informatiko, od leta 1999 pa tudi prodekan za izobraževalno dejavnost.