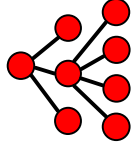


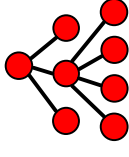
EXPRESS - jezik za modeliranje informacij

- modeliranje informacij neodvisno od strojne in programske opreme.
- **Informacijski model** je nedvoumen dogovor o načinu zapisa in interpretaciji podatkov z namenom izmenjave informacij.
- EXPRESS je bil razvit v okviru STEP standarda, kot mednarodni standard je bil sprejet leta 1994 pod oznako ISO 10303-11.
- EXPRESS omogoča opis izdelka skozi celoten življenjski cikel. Tehnični opis je zahteven.
- Ima tekstovno in grafično obliko predstavitve (EXPRESS-G). Tekstovna oblika je namenjena interpretaciji z računalnikom, medtem ko je grafična oblika preglednejša za človeka.

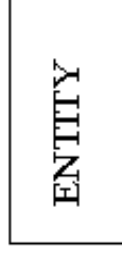
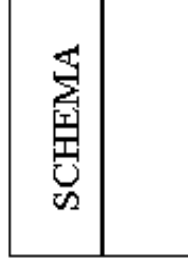


Zahteve pri jeziku za modelirane informacij:

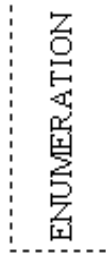
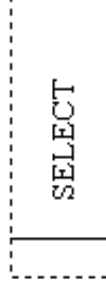
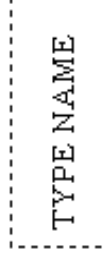
- berljiv za človeka in računalnik
- omogoča modeliranje objektov na določenem področju (napr. izdelki)
- omogoča definiranje omejitev in operacij nad objekti
- definiranje celotnih opisov
- možnosti razširitve
- spremenljivost definicij



simbol za shemo in objekt:



simboli za podatkovne tipe:



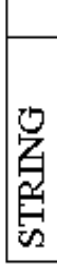
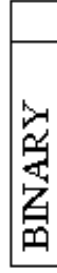
simboli za relacije:



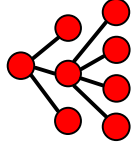
opcijski element

podtip (subtype)

simboli za osnovne podatkovne tipe:



Grafični simboli v modelirnem jeziku EXPRESS-G.

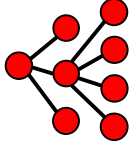


```
SCHEMA demo_schema;  
ENTITY tocka;  
  x: REAL;  
  y: REAL;  
  z: OPTIONAL REAL;  
END ENTITY;  
  
ENTITY krog;  
  sredisce: tocka;  
  radij: REAL;  
DERIVE  
  površina: REAL := PI * radij ** 2;  
END ENTITY;  
END_SCHEMA;
```

geometrija

Primer podatkovnega modela v EXPRESS jeziku.

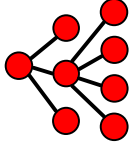
Schema (schema) združuje skupino EXPRESS elementov (objektov in omejitev), ki tvorijo zaključeno celoto. Elementi jezika se morajo vedno nahajati v določeni shemi. Pri oblikovanju sheme se lahko sklicujemo na posamezne objekte v drugih shemah.



```
ENTITY tocka;  
  x: REAL;  
  y: REAL;  
END_ENTITY;
```



Objekt (entity) je osnovni element express jezika. V najpreprostejši obliki je objekt poimenovana skupina atributov. Znotraj sheme se imena objektov ne smejo podvajati. Vsak atribut v objektu mora imeti ime in definiran podatkovni tip. Relacije med objekti tvorimo tako, da kot atribut definiramo določen objekt.

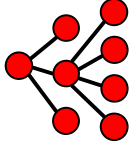


Atributom lahko pripišemo posebne lastnosti: Zahteva po **enkratnosti (unique)** določenega atributa postavlja omejitev, da nobena dva primerka objekta ne smeta imeti enake vrednosti.

```
ENTITY oseba;  
    ime: STRING;  
    priimek: STRING;  
UNIQUE  
    UR1: ime, priimek;  
END_ENTITY;
```

Izpeljanim (derived) atributom določajo vrednost ostali atributi v objektu:

```
ENTITY krog;  
    sredisce : tocka;  
    radij    : REAL;  
DERIVE  
    površina : REAL := PI * radij ** 2;  
    premer   : REAL := 2.0 * PI * radij;  
END_ENTITY;
```

**V EXPRESS jeziku definirani podatkovni tipi (TYPE):**

enostavni podatkovni tipi so podobni kakor v drugih programskih jezikih

- real
- integer
- binary
- string
- logical
- number
- boolean

sestavljene podatkovni tipi (aggregation types):

Pri modeliranju tehničnih objektov ne moremo mimo sestavljenih podatkovnih tipov, ki jih delimo na urejene (seznam in vektor) in neurejene (množica in torba).

- množica (set)

Množica je neurejena zbirka elementov, med katerimi ne sme biti ponovitev posameznih primerkov. Množica je poseben primer torbe. primer: *SET OF point* ; množica lahko vsebuje poljubno število [0 : ?] objektov point

- torba (bag)

Neurejena zbirka elementov med katerimi lahko pride do ponovitve.

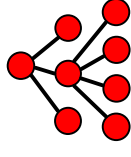
- seznam (list)

Urejena zbirka elementov, ki ima lahko postavljeno zahtevo, da ne sme biti podvajanja elementov. Do elementov v seznamu lahko dostopamo glede na njihov zaporedni položaj.

- polje (array)

Urejena zbirka oštevilčenih elementov.

primer: *ARRAY[1:10] UNIQUE REAL* - polje realnih števil z najmanj enim in največ desetimi elementi

**definirani podatkovni tipi**

Uporabnik lahko določi nove podatkovne tipe, ki jih lahko uporablja na enak način kot že vgrajene.

```
TYPE hour_in_day = INTEGER
WHERE
  WR1: { 0 <= SELF < 24 };
END_TYPE;
```

Pri naštevnikih tipih so vnaprej definirane vse možne variante (glej primer).

```
TYPE barva : ENUMERATION OF (rdeca, zelena, modra, rumena);
TYPE premik_vozila : ENUMERATION OF (levo, desno, naprej, nazaj);
```

Izbirni podatkovni tip omogoča, da podatkovni tip ni potrebno vnaprej definirati:

```
TYPE transport = SELECT(train, plane, car, bicycle);
```

```
ENTITY train;
```

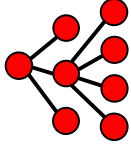
```
-- ...
```

```
END_ENTITY;
```

```
ENTITY plane;
```

```
-- ...
```

```
END_ENTITY;
```

Hierarhija podatkovnih tipov

Podatkovni tipi imajo lahko podrejene (subtype) in nadrejene (supertype) podatkovne tipe. Pri tem veljajo zakonitosti dedovanja kot je značilno za objektno orientiran pristop. Podrejeni tip podeduje od nadrejenega vse attribute, omejitve in morebiten način uporabe.

```
ENTITY curve
```

```
  SUPERTYPE OF (ONEOF(line, arc))
```

```
  color: color;
```

```
  line_type: INTEGER;
```

```
END_ENTITY;
```

```
ENTITY line
```

```
  SUBTYPE OF (curve);
```

```
  start_point: point;
```

```
  end_point: point;
```

```
END_ENTITY;
```

```
ENTITY arc
```

```
  SUBTYPE OF (curve);
```

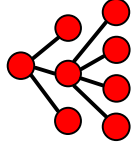
```
  start_angle: REAL;
```

```
  end_angle: REAL;
```

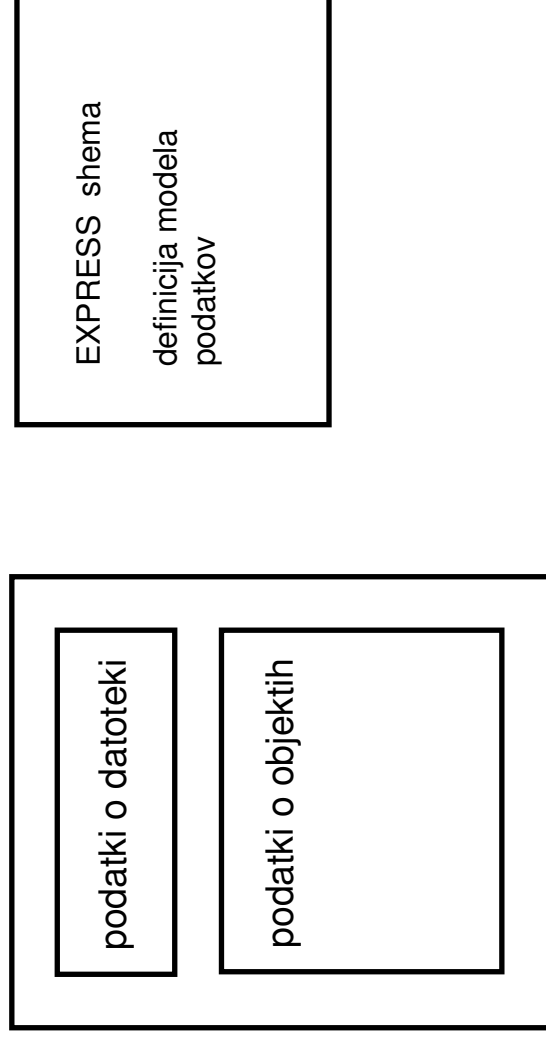
```
  radius: REAL;
```

```
  center_point: point;
```

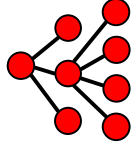
```
END_ENTITY;
```



Oblika zapisa podatkov v STEP datoteki po ISO 10303-21:



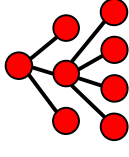
Standard predpisuje obliko nevtralnega zapisa podatkov v ASCII datoteki, ki so namenjene prenosu. Izmenjava temelji na definicijah objektov v EXPRESS informacijskem modelu. STEP datoteke običajno kreiramo s STEP procesorji.



Podatki v datoteki so razvrščeni v dva dela. Med besedama HEADER in ENDSEC se nahajajo informacije o datoteki kot so: kratek opis, avtor, čas in kraj nastanka... ter kot najpomembnejše **ime sheme** z definicijo vseh objektov v EXPRESS jeziku, ki v datoteki lahko nastopajo (struktura vhodne sekcije je definirana v shemi "header_section_schema").

V podatkovnem delu, ki je omejen z besedama DATA in ENDSEC, so zapisani vsi podatki in so zaporedno številčeni. Podatki v datoteki morajo biti zaključena celota. Tako se ne moremo sklicevati na objekte, ki v datoteki niso podani. V datoteki se nahajajo lahko le objekti in podatkovni tipi, ki so definirani v EXPRESS shemi imenovani v uvodnem delu.

Torej STEP datoteka ni samostojna, njen sestavni je shema, ki jo potrebujemo tako pri generiranju kakor branju podatkov v datoteki (ločitev definicije objektov in fizične datoteke). Če uporabljamo standardne sheme, jih ni potrebno prenašati skupaj s STEP datotekami.



EXPRESS shema z definicijo podatkov.

```
SCHEMA demo_shema;  
ENTITY TOCKA  
  x: REAL;  
  y: REAL;  
  z: OPTIONAL REAL;  
END_ENTITY;  
  
ENTITY KROG;  
  Sredisce: tocka;  
  Polmer: REAL;  
DERIVE  
  Area: REAL := PI * radij ** 2;  
END_ENTITY;  
ENDSCHEMA;
```

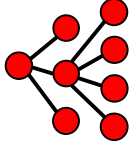
ISO-10303-21;
HEADER;
FILE_DESCRIPTION ('Datoteka vsebuje kratek primer STEP datoteke.', '1');
FILE_NAME ('demo STEP datoteka', '2000-02-11T11:45:30', ('Joze Tavcar'), ('LECAD'),
'processor version 2.4', 'CAD system AutoCAD 14.0', 'approval');
FILE_SCHEMA (('demo_shema'));
ENDSEC;

```
DATA;  
#1 = TOCKA (10.0, 50.0, $);  
#2 = TOCKA (12.0, 70.0, 20.0);  
#10 = KROG (#1, 20.0);  
#11 = KROG (#2, 30.0);  
ENDSEC;  
END-ISO-10303-21;
```

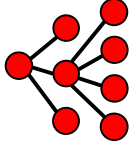
glava

Datoteka s podatki.

Primer EXPRESS sheme in STEP datoteke s podatki.

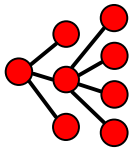
**The Instance Model: STEP exchange file format (ISO10303 Part 21 syntax)**

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION('', '2;1');
FILE_NAME('', '10.09.1999 14:18:28', ('n.n.'), (''), '', '');
FILE_SCHEMA({'PDM_SCHEMA {1.2}'});
ENDSEC;
DATA;
#10 = PRODUCT_CATEGORY_RELATIONSHIP('', $, #20, #30);
#20 = PRODUCT_RELATED_PRODUCT_CATEGORY('part', '', (#40));
#30 = PRODUCT_RELATED_PRODUCT_CATEGORY('detail', $, (#40));
#40 = PRODUCT('comp_part', 'a_part', '', (#60));
#50 = PRODUCT_CATEGORY_RELATIONSHIP('', $, #30, #100);
#60 = PRODUCT_CONTEXT('', #70, '');
#70 = APPLICATION_CONTEXT('');
#80 = APPLICATION_PROTOCOL_DEFINITION('version 1.2', 'pdm_schema', 2000,
#70);
#90 = PRODUCT_DEFINITION_FORMATION('1', '', #40);
#100 = PRODUCT_RELATED_PRODUCT_CATEGORY('spare part', $, (#40));
#110 = PRODUCT_CATEGORY_RELATIONSHIP('', $, #100, #120);
#120 = PRODUCT_RELATED_PRODUCT_CATEGORY('independent product', $, (#40));
ENDSEC;
END-ISO-10303-21;
```



Preslikava med EXPRESS shemo in STEP 21 datoteko:

```
ENTITY vozilo;  
  a : INTEGER;  
  b : STRING(2);  
  c : STRING(8);  
  d : BOOLEAN;  
  g : REAL;  
  h : LIST[1:?] OF INTEGER;  
END_ENTITY;  
  
#1 =VOZILO(100,'AB','passat',T.,254.32, (1,2,3,4))
```



Podatkovni model opisa izdelka po protokolu za uporabo AP 214.

