

ZAVOD ZA PRIMJENJENO RAČUNARSTVO
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

PROGRAMSKI JEZIK M (MICROSOFT)

Goran Kovačević

SEMINARSKI RAD IZ KOLEGIJA Programske paradigme i jezici

Zagreb, 2012.

Sadržaj

1. Uvod.....	1
2. Osnove M-a.....	2
3. Primjeri M programa.....	3
4. Zaključak.....	6
5. Literatura.....	7
6. Dodatak A.....	8

1. Uvod

U seminarskom radu obrađene su osnove karakteristike programskog jezika M. Cilj rada je prikazati korisniku mogućnosti programskog jezika M te usporediti funkcionalnosti i mogućnosti s drugim poznatim programskim jezicima. Kao referentni programski jezik uzet je C#. Rad je namijenjen studentima treće godine Fakulteta elektrotehnike i računarstva, studija programsko inženjerstvo te se podrazumijeva čitačeva upoznatost s osnovnim pojmovima u računarstvu i programiranju.

2. Osnove M-a

Puni naziv programskog jezika je *The Microsoft code name "M" Modeling Language* [1.]. Razvijen je u sklopu Oslo projekta koji nikada nije izdan. Dodatak A sadrži uputstva za instalaciju Oslo projekta i pripremu okoline za rad u programskom jeziku M. M je jezik za modeliranje domene, skup povezanih pojmova ili predmeta, koristeći tekst. Modeliranje domene sastoji se od odabira određenih osobina koje će se uključiti u model. One osobine koje se smatraju nevažnima se isključuju iz modela. M modelirane podatke pretvara u T-SQL[2.] Velika razlika između modeliranja pomoću teksta i modeliranja pomoću dijagrama je što je i programeru i računalu intuitivnije i lakše raditi s tekstom nego dijagramom. M je izgrađen na 4 osnovna koncepta:

- Jezik – skup pravila koja prepoznaju obični tekst i proizvode strukturirani prikaz relevantnih pojmova u tekstu.
- Podaci - oskudni tekstualni prikaz informacija pripremljenih za automatsko spremanje, oblikovanje i komunikaciju.
- Ograničenja - pravila koja prepoznaju specifičnu strukturu i odnose među podacima.
- Pretvorba - preslikavanje između izvornih podataka i rezultatnih podataka.

M koristi module (slično kao prostor imena u C#-u) za osiguranje jedinice enkapsulacije koja omogućuje kontrolu vidljivosti i opsega. Kao i u C#-u, doseg se određuje pomoću vitičastih zagrada. Moduli, kao i prostori imena, mogu uključivati druge module. Kako bi opisao oblik podatka M koristi ključnu riječ `type`. Ključne riječi u M-u su: `any`, `accumulate`, `by`, `empty`, `equals`, `error`, `export`, `false`, `final`, `from`, `group`, `id`, `identity`, `import`, `in`, `interleave`, `join`, `language`, `labelof`, `left`, `let`, `module`, `null`, `precedence`, `right`, `select`, `syntax`, `token`, `true`, `type`, `unique`, `value`, `valuesof`, `where`. Uz te korištene postoje i one rezervirane za još ne implementirane funkcionalnosti: `checkpoint`, `identifier`, `nest`, `override`, `new`, `virtual`, `partial`.

Tipovi literala su: `Binary`, `Date`, `DateTime`, `DateTimeOffset`, `Decimal`, `Guid`, `Integer`, `Logical`, `Null`, `Scientific`, `Text`, `Time`.

Atributi tipa se deklariraju na način da se ime atributa odvoji od vrste atributa operatorom `:"`. Vrsta atributa mogu biti svi opći literali i dodatni tipovi stvoreni modifikacijom literala. Dodatni su: `Any` - označava bilo koju vrijednost, `Collection` – ne poredana grupa vrijednosti, `Entity` – skup imenovanih varijabli, `General` – sve jednostavne vrste, `List` – poredana grupa vrijednosti, `Number` – bilo koja numerička vrijednost, `Unsigned` – bez predznačna vrijednost.

3. Primjeri M programa

Kod 3.1 prikazuje program napisan u programskom jeziku M. Program pokazuje stvaranje oblika podataka pomoću ključne riječi `type`. Unutar domene ključne riječi, odvojeno vitičastim zagradama, definiraju se atributi i tipovi atributa. Npr. `Name` je naziv atributa koji je tipa `Text`. Neki tipovi atributa imaju ugrađene metode, `constraints`, pomoću kojih se mogu definirati dodatne opcije ili ograničenja nad atributima. U ovom primjeru atribut `Name` ne smije imati više od 20 znakova jer je tako definirano pomoću metode `Count` i ključne riječi `value` koja označava da se radi o trenutnom podatku. Nakon definiranja mogu se definirati primarni ključevi T-SQL-a. Ključna riječ `identity` to omogućava. M ima mogućnost definiranja relacija između objekata određenog tipa. U danom primjeru definirano je da može postojati nula ili više objekata tipa `Language`. Nakon definiranja modela, upisani su neki osnovni podaci. Kao i sve u M-u, tako i upis podataka se vrši unutar bloka odvojenog vitičastim zagradama. To je osobina koja karakterizira sve Microsoft-ove programske jezike, npr. `C#`. Još jedna zajednička stvar M-a i `C#`-a je obvezno definiranje kraja linije pomoću znaka `;`.

```
module PPiJ
{
    type Language
    {
        Name : Text where value.Count <= 20;
        Year : Integer16;
        Creator : Text;
    } where identity Name;

    Languages : Language* {
        { Name = "C#", Year = 2001, Creator = "Hejlsberg"},
        { Name = "C++", Year = 1983, Creator = "Stroustrup"},
        { Name = "Haskell", Year = 1990, Creator = "Peyton-Jones"},
        { Name = "Python", Year = 1991, Creator = "van Rossum"},
        { Name = "F#", Year = 2005, Creator = "Syne"},
        { Name = "M", Year = 2008, Creator = "Langworthy"}
    }
}
```

Kod 3.1. Jednostavni program u M-u

Pokretanjem Repository T-SQL Prewiev opcije iz M mode padajućeg izbornika u editoru, pojavljuje se T-SQL kod za zadani primjer u posebno odvojenom prozoru. Na ispisu 3.1. se nalaze dijelovi T-SQL koda za primjer 3.1 koji pokazuju stvaranje i definiranje tablica te `update` naredbu za jedan par vrijednosti.

```

create table [PPIJ].[LanguagesTable]
(
    [Name] nvarchar(20) not null,
    [Creator] nvarchar(max) not null,
    [Year] smallint not null,
    constraint [PK_Languages] primary key clustered ([Name])
);
.....
insert into [PPIJ].[Languages] ([Name], [Year], [Creator])
        values (N'Python', 1991, N'van Rossum');
....

```

Ispis 3.1. Dio T-SQL koda za primjer 3.1.

Korištenjem prevoditelja stvara se .sql datoteka koju je moguće koristiti s bilo kojim od dostupnih alata za rad s bazama podataka.

Kao i u C#-u, i u M-u postoji enumeracija. Kod 3.2 prikazuje primjer zatvorene enumeracije. Definiranjem tipa `ColorsEnum` uz ključnu riječ `type` onemogućuje se mijenjanje vrijednosti tako da atribut `Color` u tipu `ToyCar` ne smije imati niti jednu drugu vrijednost osim tri navedene u `ColorsEnum`. Ukoliko se `ColorsEnum` definira bez ključne riječi `type` onda atribut `Color` može poprimiti bilo koju vrijednost. Time bi se dobila otvorena enumeracija.

```

module ClosedEnumerationSample {
    type ColorsEnum {
        "Red",
        "Blue",
        "Yellow",
    }
    type ToyCar {
        Color : ColorsEnum;
        // useful work
    }

    ToyCars : ToyCar*;
}

```

Kod 3.2. Primjer enumeracije

Još jedna sličnost M-a s C#-om je implementacija LINQ-a. Kod 3.3 prikazuje primjer jednostavnog modula s LINQ upitom. Sintaksa je ista kao u C#-u. Definirana je lista brojeva i spremljena u atribut `Numbers`. Funkcija `LowNums` se sastoji od LINQ upita koji iterira po elementima liste i uzima samo one koji su manji od 5.


```
module Lin
{
    Numbers : Integer32* { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0}

    LowNums() {
        from n in Numbers
        where n < 5
        select n
    }
}
```

Kod 3.2. Primjer LINQ upita

4. Zaključak

Programski jezik M, kao i bilo koji drugi jezik za modeliranje baze podataka, ima svojih prednosti nad standardnim uređivačima baze podataka. Budući da M nije nikada zaživio u potpunosti teško je pronaći bilo koji važniji primjer njegove primjene. Odustajanje od razvoja i izdavanja jezika je razumljivo jer gotov sve mogućnosti koje M ima sadrži i C#. Gledajući koliko sve proširenja i okvira za C# postoji, izdavanje potpuno novog jezika s ograničenom funkcionalnosti se čini kao unaprijed propali projekt.

5. Literatura

1. The Microsoft code name "M" Modeling Language Specification – Introduction, URL: <http://msdn.microsoft.com/en-us/library/dd285271.aspx> (14/6/12)
2. Transact-SQL – Wikipedia, the free encyclopedia URL: <http://en.wikipedia.org/wiki/Transact-SQL> (14/6/12)

6. Dodatak A

Paket Oslo projekta potreban za pokretanje M modula može se besplatno skinuti s Microsoft-ove internet stranice: <http://download.microsoft.com/download/5/4/B/54BF5C12-20B8-462E-802E-0BCA2BCBF72C/OsloSDKJan2009CTP.msi>. Skinuti SDK (engl. Software Development Kit) se može pokrenuti samo u operacijskom sustavu Windows, XP edicija ili novije. Instalacija se vrši praćenjem jednostavnih uputstava u čarobnjaku za instalaciju. U sklopu instalacijskog paketa dolazi i tekstualni editor Intellipad za pisanje M modula. Nakon pokretanja editora potrebno je u gornjem desnom kutu prozora promijeniti način rada iz "Standard Mode" u "M Mode" kako bi editor bio spreman za rad s M modulima.