



Dinamičko izvršavanje procedura i funkcija preko konfiguracije

Stipe Aras-Gazić

O autoru

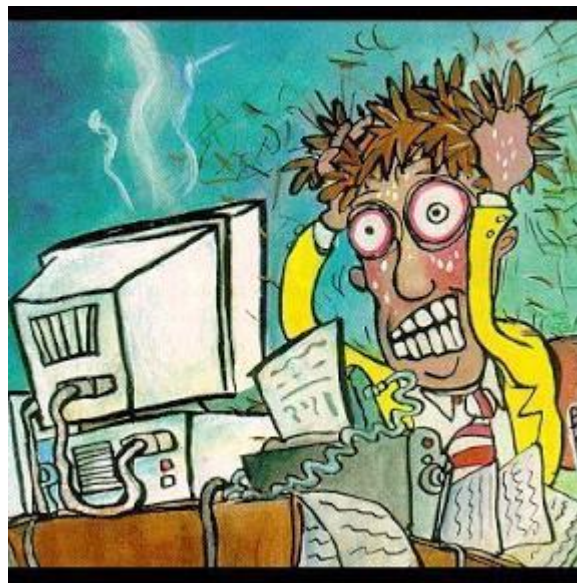


stipe.aras-gazic@multicom.hr

- Multicom d.o.o.
- Sektor razvoja
- Voditelj tima za BSS
- Projekti:
 - CRM
 - BillFormater
 - MDM

Razvoj aplikacija u PL/SQLu

- veliki i komplicirani sustavi – mnogo kôda
- nadogradnjama kôd raste
- spor kod izvršavanja
- težak za održavanje.



Razvoj aplikacija u PL/SQLu

- Kako izbjeći kaos?
- Pobojšati čitljivost koda?
- Efikasnost razvoja?
- Bolje performanse?



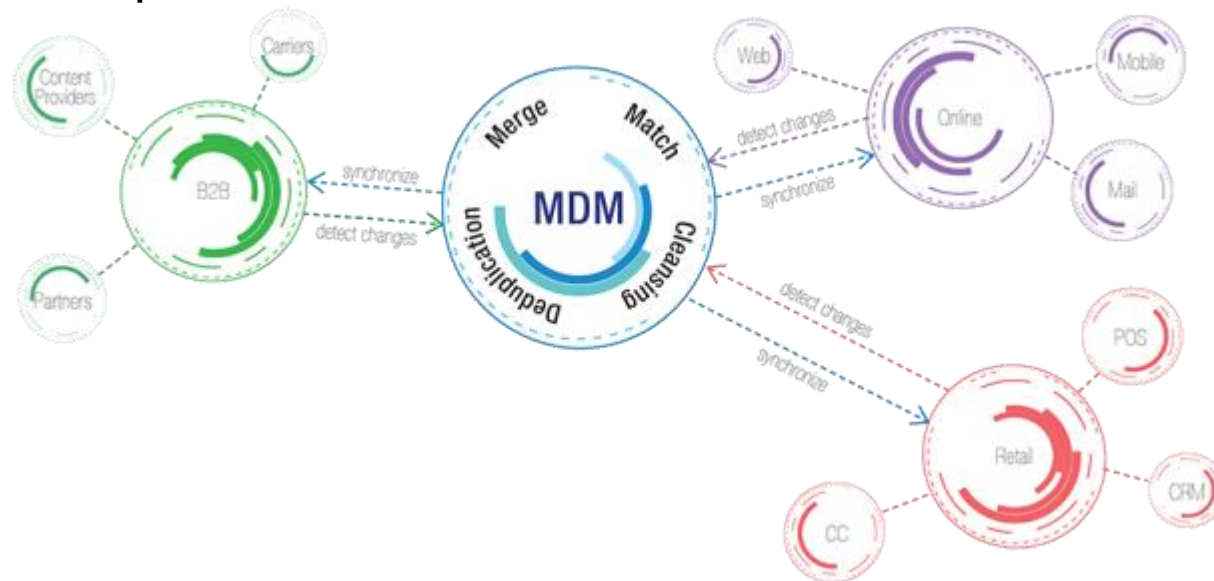
Razvoj aplikacija u PL/SQLu

- Jedno od rješenja:
 - univerzalne procedure/funkcije
 - dinamičko izvršavanje
 - konfiguracijske tablice



MDM

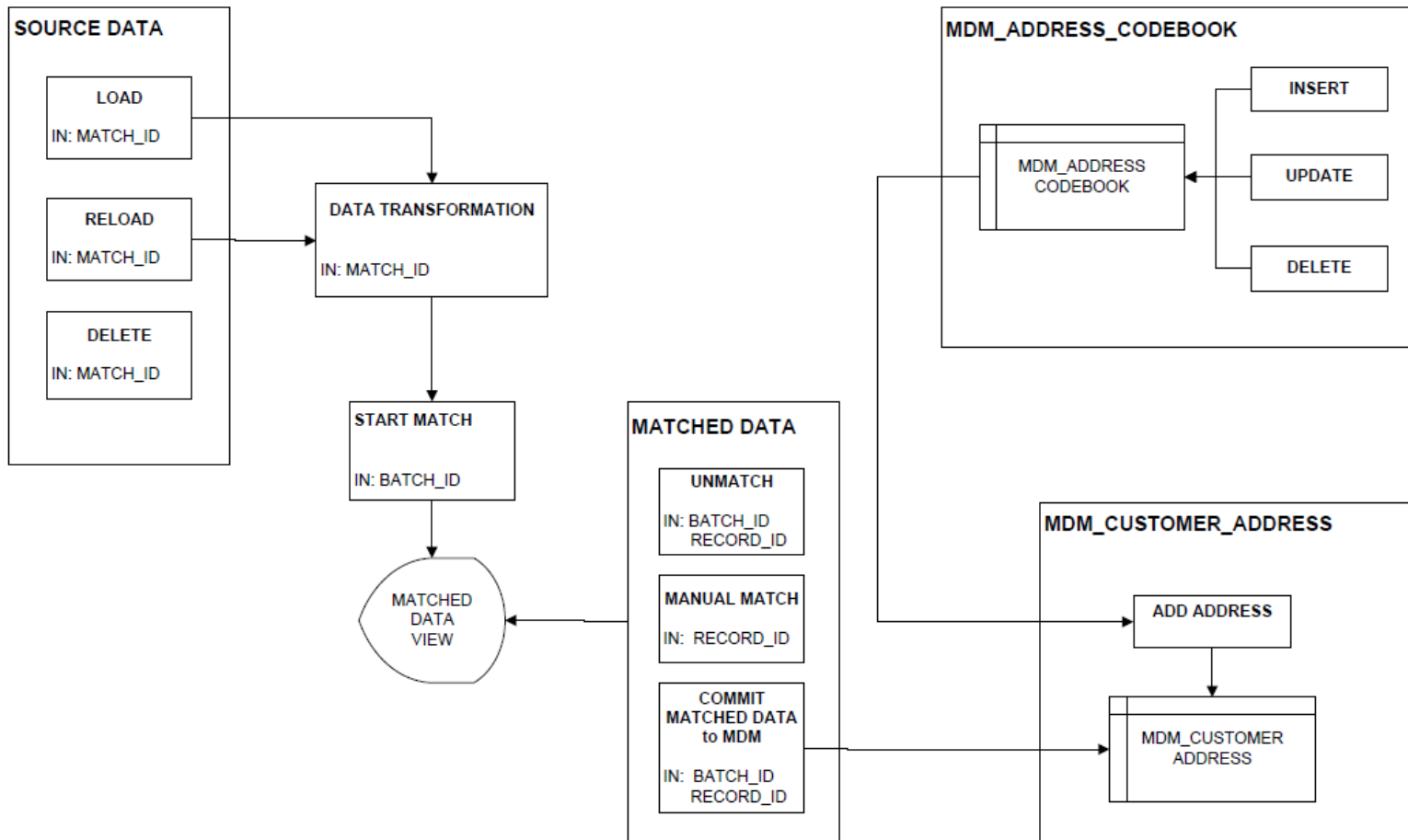
- MDM - Master Data Management
<http://www.multicom.hr/solutions/crm/customer-mdm/>
- Sustav koji se brine o matičnim podacima (kupaca, *customer*)
- Puno sustava u velikim firmama održava svoje zapise sa matičnim podacima (npr. OSS/BSS sustavi)
- Za razliku od ostalih sustava, svrha MDM sustava je upravljanje matičnim podacima



MDM

- Apstraktni model podataka kupca je podesiv (“konfigurabilan”) – vrlo je jednostavno izmjeniti strukturu pojedinih entiteta ili dodati relaciju među entitetima
- Dobre performanse
- Fleksibilnost prilikom definicije podatkovnog modela
- Jednostavno uklapanje u poslovne procese – konfiguriranje pravila

MDM Address matching module



MDM Address matching module – čemu služi

- Adrese se nalaze u više različitih izvornih tablica
- Proizvoljan način unosa adresa u raznim sustavima
- Bez jedinstvenog identifikatora
- MDM centralni šifarnik adresa
- Cilj - adresama iz izvornih sustava dodijeliti jedinstvene identifikatore koristeći razne algoritme za spajanje adresa

MDM Address matching module – kako radi

- Set procedura, funkcija i konfiguracijskih tablica
- uvođenje novog algoritma kroz administraciju konfiguracijskih tablica
- Nepotrebno posebno pisati procedure ili funkcije za svaki algoritam
 - smanjuje broj objekata u bazi
 - olakšava održavanje
 - olakšava nadogradnju modula s novim algoritmima

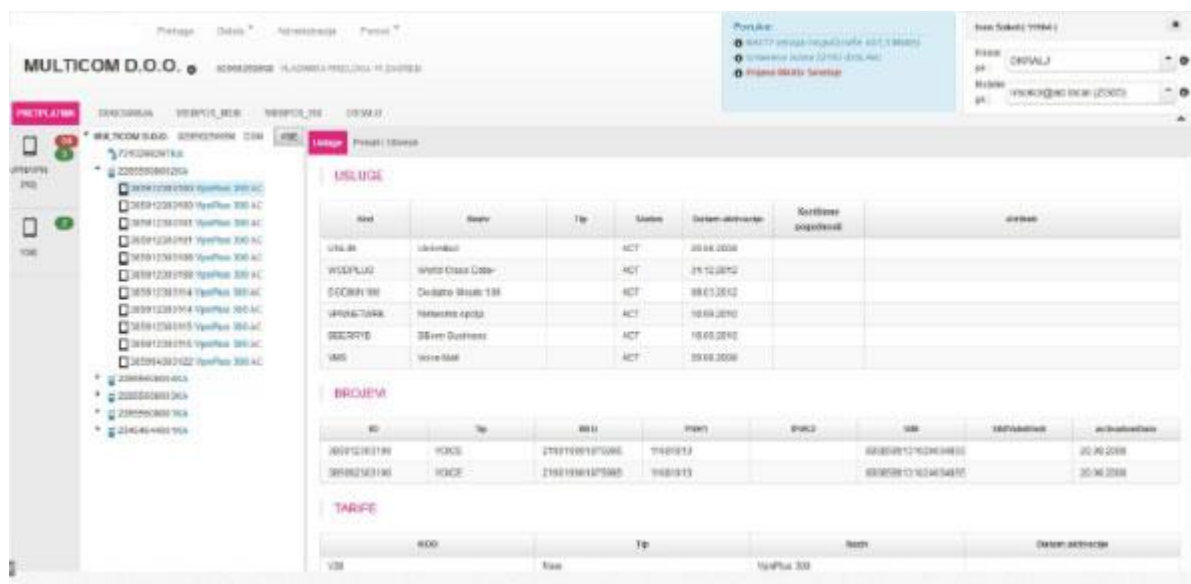
MDM Address matching module – kako radi

- 1. korak
 - Load izvornih podataka
 - Čišćenje ulaznih podataka (npr. REGEXP)
 - Sve preko konfiguracijskih tablica

COLUMN_ID /	COLUMN_ALIAS	COLUMN_EXPR	CUSTOM_EXPR
1	MESSAGE_ID	MESSAGE_ID_SEQ.nextval	
2	MESSAGE_TIMESTAMP	SYSDATE	
3	SOURCE	'1'	
4	SOURCE_ID	ID	
5	SRC_DISPLAY_ADDRESS	ADRESA	
6	SRC_STREET_NAME	NAZIV_ULICE	
7	SRC_STREET_NR	KUĆNI_BROJ ' ' KUĆNI_BROJ_DODATAK	
8	SRC_CITY	NAZIV_NASELJA	
9	SRC_ZIP_CODE	regexp_substr (s_poste, '[0-9]{5}')	
10	SRC_ZIP_NAME	NAZIV_POSTE	
11	SRC_COUNTY	NAZIV_ZUPANDE	
12	SRC_STREET_ID	S_ULICE	
13	SRC_PLACE_ID	S_NASELJA	
14	SRC_COUNTY_ID	S_ZUPANDE	
15	MDM_REC_CRE_DATE	SYSDATE	
16	MDM_REC_CRE_BY	USER	
17	SOURCE_CHANNEL	'2'	
18	ID	SOURCE_ID	
19	PHRASE	SRC_STREET_NAME	
20	ID	STREET_ID	

MDM Address matching module – kako radi

- 2. korak
 - Integracija izvornih podataka (MDM integration module)
 - Priprema za matching



MDM Address matching module – kako radi

- 3. korak
 - Pokretanje Matchinga
 - Funkcije za ocjenjivanje i validaciju spoja
 - Automatski spoj ili je potrebna dodatna manualna odluka

RULE_NAME	WEIGHT	LEVEL_NO	DESCRIPTION	CURSOR_SQL	STATE	DECISION_FUNCTION_ID	DECISION_THRESHOLD
PBR_NAS_UL	100	0	post + naselje + ulica (točna...	select distinct adr.SOURCE_ID, v.P...	1	100	
PBR_MJ_UL	100	1	post + post_mjesto + ulica (...	select distinct adr.SOURCE_ID, v.P...	1	100	
PBR_UL	80	1	post + ulica (točna ulica)	select distinct adr.SOURCE_ID, v.P...	1	100	
NAS_UL	80	0	naselje + ulica (točna ulica)	select distinct adr.SOURCE_ID, v.P...	1	100	
MJ_UL	80	1	post_mjesto + ulica (točna u...	select distinct adr.SOURCE_ID, v.P...	1	100	

MDM Address matching module – kako radi

- 4. korak
 - Integracija spojenih parova



MDM Address matching module – zaključak

- Završni rezultat je modul koji se sa svojom konfiguracijom može koristiti za spajanje adresa između bilo kojih izvornih sustava i internog referentnog šifarnika adresa

... HVALA!



Q&A