

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Špilak

**Upravljanje z naročili in integracija z zalednimi
sistemi v telekomunikacijskem podjetju**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Mojca Ciglarič

Ljubljana, 2013

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 01892/2013

Datum: 07.01.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANDREJ ŠPILAK**

Naslov: **UPRAVLJANJE Z NAROČILI IN INTEGRACIJA Z ZALEDNIMI SISTEMI
V TELEKOMUNIKACIJSKEM PODJETJU**

**ORDER MANAGEMENT AND ITS INTEGRATION IN A
TELECOMMUNICATIONS COMPANY INFRASTRUCTURE**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:


Preučite specifično upravljanje poslovnih procesov, zlasti pa upravljanje z naročili v telekomunikacijskem podjetju. Opredelite osnovne pojme, predstavite zanimive tehnologije in opišite uporabljane pristope v sodobnih vsebinsko podobnih sistemih. Povzemite priporočila in standarde TM Forum-a in pojasnite pomen standardizacije na tem področju. Nato opišite projekt vzpostavljanja informacijske podpore upravljanju naročil v telekomunikacijskem podjetju: ključne sisteme v podjetju, arhitekturo rešitve, načrtovanje generičnega podatkovnega modela in sledenje standardom pri izmenjavi podatkom. Pri slednjem pojasnite tudi pozitivne učinke na razvoj nadaljnjih integracij. Nazadnje kritično ovrednotite narejeno in podajte predloge za nadaljnje delo.

Mentor:


doc. dr. Mojca Ciglaric



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU diplomskega dela

Spodaj podpisani Andrej Špilak,
z vpisno številko 63010149,

sem avtor diplomskega dela z naslovom:

Upravljanje z naročili in integracija z zalednimi sistemi v telekomunikacijskem podjetju

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 18.04.2013

Podpis avtorja: _____

Zahvala

Na prvem mestu bi se rad zahvalil staršema, da sta mi omogočila študij.

Sodelavcem iz podjetja Crea se zahvaljujem za veliko pridobljenega znanja, prav tako pa gre zahvala mentorici doc. dr. Mojci Ciglarič, ki mi je pomagala, da sem svoje izkušnje uspešno zil v tole diplomsko delo.

Na zadnje še iskrena zahvala Svetlani, ki me je bodrila, ko je bilo najtežje.

Kazalo

Povzetek	1
Abstract.....	2
1. Uvod.....	3
1.1. Uvod v problematiko.....	3
1.2. Namen in cilji.....	4
2. Opredelitev tehnologij	5
2.1. Priporočila in ogrodja izpod okrilja TM Foruma.....	5
2.1.1. Frameworkx	5
2.1.2. eTOM.....	6
2.1.3. Osnovna umestitev modela SID in njegov namen	8
2.1.4. Informacijski model.....	8
2.1.5. Podatkovni model.....	9
2.1.6. Povezava SID z eTOM.....	10
2.2. Spletne storitve	11
2.2.1. Storitveno usmerjena arhitektura	11
2.2.2. Standard za spletne storitve SOAP	12
2.2.3. Jezik za opisovanje spletnih storitev	13
2.3. Poslovni procesi in njihovo upravljanje.....	14
2.3.1. Opredelitev poslovnega procesa	14
2.3.2. Upravljanje poslovnih procesov	14
2.3.3. Sistemi za upravljanje poslovnih procesov.....	15
3. Podroben pregled ogrodja SID	17
3.1. Domene delovanja.....	18
3.2. Agregirane poslovne entitete	18
3.2.1. ABE za domeno produkta.....	19
3.2.2. ABE za domeno storitev.....	20
3.3. Agregirane systemske entitete	21
3.4. Uporaba programskih vmesnikov in modela SID	21
3.4.1. Motivacija za uporabo programskih vmesnikov, temelječih na SID.....	21
3.4.2. Implementacija programskih vmesnikov, temelječih na SID	23
3.5. Načrtovalni pristopi SID	24
3.5.1. Specifikacija entitete/entiteta	24
3.5.2. Entiteta/vloga entitete	25
3.5.3. Kompozitno/atomarno	25
3.5.4. Entiteta specifikacija karakteristik/entiteta karakteristik.....	25
3.5.5. Primer na specifikaciji produkta	26
4. Sistemi telekomunikacijskega operaterja in arhitektura OMS.....	29
4.1. Razvoj sistema za upravljanje z naročili in arhitekture B/OSS.....	29
4.2. Sistem za upravljanje z naročili (OMS).....	31
4.2.1. Arhitektura rešitve in tehnologije.....	31
4.2.2. Podpora poslovnim procesom.....	31
4.3. Zaledni sistemi in način povezovanja z OMS.....	34
4.3.1. Produktni katalog	34
4.3.2. Sistem za upravljanje s storitvami in strankami.....	37
4.3.3. Sistem za zagotavljanje pred-provizioniranja	39
4.3.4. Sistem za upravljanje subvencij.....	39
4.4. Primer novega naročila v OMS z opisom odvisnosti.....	39

5. Podatkovna plast OMS.....	41
5.1. OrderContract	43
5.2. OrderItem.....	43
5.3. Characteristic.....	45
5.4. Price.....	46
5.5. Customer	46
5.6. Globalen šifrant.....	46
6. Zaključek	48
Literatura.....	49
Slike.....	51
Tabele.....	52

Seznam uporabljenih kratic in simbolov

ABE	Aggregated Business Entity
OSS	Operations Support System (sistemi za podporo poslovnih operacij)
BSS	Business Support System (sistemi za podporo poslovanja)
NGOSS	New Generation Operations Systems and Software
OMS	Order Management System (sistem za upravljanje z naročili)
PLM	Product Lifecycle Management (upravljanje življenjskega cikla proizvoda)
CIM	Common Information Model (skupen informacijski model)
IP	Internet Protocol
eTOM	enhanced Telecom Operations Map
TAM	Telecom Applications Map
TNA	Technology Neutral Architecture
WSDL	Web Services Schema Definition
POS	Point Of Sale (prodajna točka)
BO	Back Office
WCF	Windows Communication Foundation
API	Application programming interface
TOCP	Telecom Operations Content Pack
ADSL	Asymmetric Digital Subscriber Line
DVB-T	Digital Video Broadcasting – Terrestrial
GSM	Global System for Mobile Communications (orig. Groupe Spécial Mobile)
VoIP	Voice over IP
CFS	Customer Facing Service
RFS	Resource Facing Service

Povzetek

Telekomunikacijska industrija se nenehno razvija, rezultat tega pa je vedno večja ponudba storitev končnim uporabnikom. Trenuten trend ponudbe paketov trojčkov in četverčkov, ki vsebujejo storitve, realizirane na različnih tehnologijah, pomeni tudi kompleksnejšo informacijsko podporo ponudnikov storitev.

Diplomsko delo obravnava sistem za upravljanje z naročili (OMS), ki ima nalogo omogočiti vnos naročila in upravljati celoten proces naročanja storitev vse do realizacije naročila. Z namenom doseganja čim večje stopnje avtomatizacije poslovnega procesa se OMS povezuje z velikim številom drugih sistemov, ki skrbijo vsak za svojo domeno delovanja. Zaradi kompleksnosti informacijske podpore v sodobnem telekomunikacijskem podjetju je delitev na domene delovanja nujna, kar pa otežuje obvladljivost podatkov. Zato je združenje TM Forum razvilo priporočila za razvoj informacijskih sistemov in priporočila za upravljanje poslovnih procesov in informacij.

Delo v prvem delu predstavi tehnologije, ki jih je treba poznati za uspešen razvoj sistema za upravljanje z naročili v telekomunikacijski domeni. Najprej je predstavljen skupek ogrodij TM Forum, ki so zbrana pod skupnim imenom Framework. Delo predstavi tudi spletne storitve SOAP kot tehnologijo za izmenjavo podatkov, predstavi pa tudi osnove sistemov za upravljanje poslovnih procesov, ki so potrebne za implementacijo procesnega dela OMS.

V tretjem poglavju se delo podrobneje posveti predstavitvi ogrodja SID, komponente Framework, ki predstavlja standarden, deljen model za predstavitev informacij. Razdela predvsem domeni produkta in storitev, ki sta najpomembnejši z vidika upravljanja naročil.

Osrednji del diplomskega dela predstavi OMS, njegovo arhitekturo in pester razvoj sistema v dveh letih produkcijske uporabe, prav tako pa predstavi tudi ključne sisteme, s katerimi se OMS povezuje.

Za konec je predstavljena implementacija podatkovnega modela in podana razlaga, kako OMS uparja podatke, pridobljene preko spletnih storitev produktnega kataloga in sistema za upravljanje storitev, katerih shema vmesnikov SOAP sledi priporočilom ogrodja SID.

Ključne besede:

Sistem za upravljanje z naročili, informacijski model v telekomunikacijah, SID, BPM

Abstract

The telecommunications industry is constantly evolving, which has resulted in a growing number of services for end-users. The current trend of triple- and quadruple-play offers that include services using different technologies have caused the IT support for the IT service providers to become increasingly complex.

This diploma paper deals with the order management system (OMS) whose purpose is to support order entry and order handling as an end to end process. In order to achieve the greatest possible degree of business process automation, OMS performs integrations with numerous other systems that are responsible for other domains.

Due to the complexity of IT in modern telecommunications companies, the fragmentation of IT into specific domains is required, but this makes the data management within an organization much more difficult. Therefore, the TM Forum has developed frameworks for the organization of IT and recommendations on how business processes should be defined and how enterprise information model should be implemented.

In its first part, this thesis presents the technologies that are required for the successful implementation of the order management system in telecommunications. First, the TM Forum's Framework is described, followed by SOAP web services. The business process modeling systems basics are introduced, since these are required for the implementation of the order handling business process.

The third chapter is devoted to the SID framework analysis, a Framework component that deals with shared information and data model. It focuses mostly on product and service domains, since these are the most important from the order handling perspective.

The main part of the thesis then presents the OMS, its architecture and a rich history of optimizations of the system in the last two years since its initial production deployment. The main systems that OMS is integrated with are presented as well.

Finally, the implementation of the OMS data model is presented. The chapter also explains how data obtained from the product catalog and service management system is being paired in the OMS.

Keywords:

Order management system, information model in telecommunications, SID, BPM

1. Uvod

1.1. Uvod v problematiko

Prehod na omrežja novejšje generacije, ki temeljijo na internetnem protokolu (IP), pomeni nov izziv za telekomunikacijsko industrijo. Ta omogoča ponudnikom storitev, da razširijo svojo ponudbo in svojim strankam ponudijo pakete storitev, t. i. trojčke ali v morebitnem primeru vključitve mobilne telefonije v svoj portfelj ponudbe celo četverčke.

Po predvidevanjih [1] bodo storitve, kot so VoIP, IP-TV, mobilna TV idr., nadomestile starejše tehnologije. Konvergenca, ki temelji na IP-ju, je neizogibna, njene posledice pa bodo nove storitve in nove potrebe po storitvah na tržišču. Za ponudnike storitev to pomeni, da bo morala biti njihova informacijska podpora dovolj fleksibilna, da bo omogočala hitro uvedbo novih storitev na tržišče, hkrati pa učinkovita pri upravljanju skozi njihov celoten življenjski cikel.

Podobna je tudi situacija telekomunikacijskega podjetja, predstavljenega v tem delu, ki je nastalo z združitvijo dveh operaterjev. Prvi operater je pokrival storitve mobilne telefonije, drugi pa področje fiksnih storitev (fiksna telefonija, internet in televizija) in združeno podjetje je naenkrat imelo vse možnosti ponujanja raznovrstnih storitev v enotnih paketih.

A pri taki združitvi in povečani ponudbi na račun raznolikosti storitev se posledično povečuje kompleksnost informacijske podpore in podsistemov, ki jo sestavljajo.

Zato je združeno podjetje šlo skozi več faz informacijske prenove, kjer so v prvi, najnujnejši, fazi zamenjali sistem za upravljanje z GSM naročniškimi sistemi, saj je bil lastnik sistema istočasno tudi lastnik konkurenčnega operaterja na njihovem trgu. To je omogočalo konkurenčnemu podjetju vpogled v podatke o naročniških paketih, kar je bilo seveda nesprejemljivo. Hkrati pa je v tej fazi združeno podjetje uvedlo nov sistem za upravljanje z naročili v kombinaciji s produktnim katalogom in poudarkom na optimizaciji poslovnih procesov. Sistem za upravljanje z naročili je razvilo podjetje, v katerem sem zaposlen.

Sledilo je še pet večjih projektov informacijske prenove, ki so, nekateri bolj, drugi manj, zadevali naš sistem za upravljanje z naročili. Vse prenove so se zgodile v obdobju 2 let, kar priča o tem, kako pomembne so prilagoditve informacijskih sistemov v telekomunikacijah z namenom prilagoditve tržišču in kako pomembna je njihova dobra tehnična zasnova, ki mora omogočati, da so podobni posegi čim manj boleči. Tako učinkovitost najlažje dosežemo s sledenjem priporočilom in standardom, ki so bili razviti na podlagi dobrih praks, za kar na telekomunikacijskem področju skrbi TeleManagement Forum (v nadaljevanju: TM Forum).

V diplomskem delu bom zato prikazal, kako je zasnovan sistem za upravljanje z naročili (angl. *Order Management System*, v nadaljevanju tudi OMS) in kakšna priporočila za telekomunikacijsko domeno predvideva TM Forum z vidika načrtovanja podatkov. Ker za kompleksno telekomunikacijsko informacijsko podporo po navadi skrbi veliko sistemov, si bomo ogledali, kako so izvedene učinkovite integracije med sistemi v telekomunikacijski domeni.

1.2. Namen in cilji

V prvem, teoretično-tehnološkem, delu diplomskega dela želim opredeliti osnovne pojme, tehnologije in pristope, ki so potrebni za razumevanje in uspešno implementacijo informacijskega sistema ali modula v telekomunikacijski domeni. Podrobneje bom pregledal priporočila TM Foruma s poudarkom na modelu SID (*Shared Information/Data model*), ki je uporaben tudi za razvoj vmesnikov spletnih storitev.

V četrtem poglavju bom predstavil arhitekturo rešitve OMS in nekatere ključne sisteme v telekomunikacijskem podjetju, s katerimi se OMS povezuje. Analiziral bom vmesnike zunanjih sistemov in ugotavljal njihovo sledenje modelu SID ter poskušal ugotoviti, kaj je ključni doprinos pri uporabi deljenega informacijskega modela.

Za konec bom predstavil, kako je načrtovan podatkovni model OMS z namenom, da bi bil čim bolj generičen in zato omogočal čim enostavnejšo implementacijo zahtev ter prilagoditev sistema. Predstavil bom tudi, kako OMS rešuje mapiranje podatkov v strukturo SID.

2. Opredelitev tehnologij

2.1. Priporočila in ogrodja izpod okrilja TM Foruma

TM Forum je neprofitna organizacija, ki ima za poslanstvo ponudnikom storitev omogočati boljšo odzivnost in povečevati inovativnost. Organizacijo sestavljajo ponudniki storitev na področju telekomunikacij kot tudi njihovi dobavitelji. V letu 2013 organizacija šteje nekaj več kot 900 članov. [2]

2.1.1. Framework

Do pred kratkim znan tudi kot NGOSS (*New Generation Operations Systems and Software*) je celovito ogrodje za razvoj, vzpostavitev in uvedbo operacijskih in poslovno podpornih sistemov (OSS¹/BSS²) ter njihove programske opreme.

Sestavljajo ga štiri razvojna ogrodja, ki vsako s svojega vidika skrbi za boljšo poslovno in informacijsko organiziranost:

- **eTOM** (*enhanced Telecom Operations Map*) – razvojno ogrodje s priporočili za izvajanje in podporo poslovnih procesov v telekomunikacijah,
- **model SID** (*Shared Information/Data Model*) – deljen informacijski model, neodvisen od platforme, jezika ali protokolov,
- **TAM** (*Telecom Applications Map*) – ogrodje, ki se osredotoča na vloge in funkcionalnosti z vidika posameznih aplikacij,
- **TNA** (*Technology-Neutral Architecture*), ki definira interakcije med poslovnimi procesi in podatki s pomočjo formalnih definicij, primerov uporabe in priporočil za razvoj komponent.

Cilj Frameworka je spopasti se s številnimi izzivi, s katerimi se srečujejo organizacije s komunikacijskega in informacijskega področja. Ti cilji so naslednji:

- zmanjšati čas, potreben za lansiranje ponudbe na tržišču (angl. *time to market*),
- zmanjšati stroške za integracijo poslovnih procesov in programskih modulov,
- zmanjšati čas za upravljanje in stroške, povezane z upravljanjem,
- pospešiti uvajanje novih tehnologij in storitev, ki temeljijo na teh tehnologijah,
- podpreti uvedbo številnih aplikativnih tehnologij.

Uvedba novih tehnologij in storitev, ki so provizionirane³ na teh tehnologijah, lahko pomeni velik strošek za organizacijo. Ta strošek in čas uvedbe je z uporabo Frameworka možno zmanjšati, poleg tega pa je možno postopek ponoviti ob uvedbi novih tehnologij na podlagi skupno razvitega ogrodja.

S tehničnega vidika temelji Framework na naslednjih petih ključnih principih:

- ločevanje poslovnih procesov od implementacije komponent,

¹ *Operations Support System* – skupek sistemov telekomunikacijskega podjetja, katerih namen je zagotavljanje storitev (predvsem na nivoju omrežja, storitev, inventarja ...)

² *Business Support System* – skupek sistemov, ki jih uporablja telekomunikacijsko podjetje za vodenje poslovanja in upravljanja poslovnih procesov s strankami

³ angl. »*provisioning*«, je proces zagotovitve omrežja in virov za priključek storitve. Terminološki slovar informatike (www.islovar.org) priporoča uporabo besede »oskrbovanje«, vendar v svojem delu zaradi jasnosti izraza raje uporabljam izraz »provizioniranje«

- šibko sklopljeni porazdeljeni sistemi,
- skupen informacijski model,
- skupna komunikacijska infrastruktura,
- nespremenljive definicije vmesnikov.

Ti zagotavljajo, da sistemi učinkovito upravljajo področje (zgolj) svoje domene in omogočajo enostavno zamenljivost v primeru, da po tem nastopi potreba.

2.1.2. eTOM

Je večnivojsko ogrodje za razvoj poslovnih procesov, ki se osredotoča na različne nivoje poslovnih procesov v telekomunikacijskem podjetju glede na njihovo pomembnost za uspešno poslovanje. Ogrodje je definirano zelo generično z namenom, da je organizacijsko, tehnološko in storitveno neodvisno.

Na vrhnjem, konceptualnem, nivoju ogrodje predstavlja tri glavna procesna področja:

- upravljanje **operacij**, ki pokrivajo ključna dnevna opravila,
- upravljanje in določanje življenjskega cikla **strategije, infrastrukture in produktov**,
- **upravljanje organizacije** z vidika podpornih procesov.

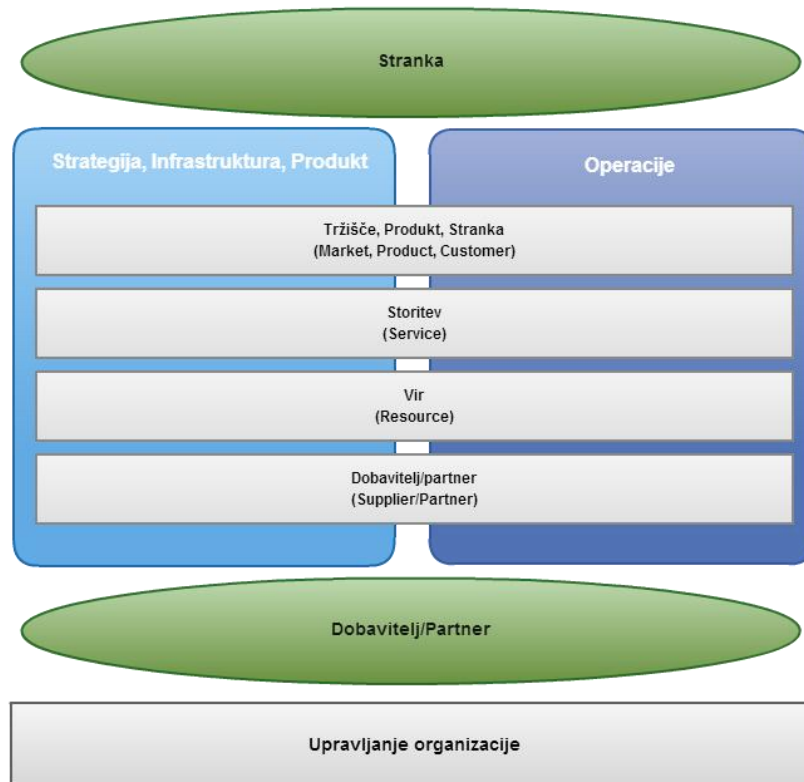
Posamezna procesna področja se nato na naslednjih nivojih razčlenijo podrobneje. Upravljanje **operacij** se na nivoju 1 deli na:

- **Izpolnjevanje** (*Fulfillment*) je proces izpolnjevanja naročil, ki po navadi izhajajo iz prodajnih točk⁴
- **Zagotavljanje storitev** (*Assurance*) obsega proces vključevanja, delovanja in sprememb na storitvah
- **Zaračunavanje** storitev in upravljanje dohodkov (*Billing*)
- **Podpora delovanju in pripravljenost** (*Operations Support and Readiness*) obsega procese v zalednem poslovanju, ki omogočajo podporo in avtomatizacijo prejšnjim trem točkam.

Procesna vertikala **Strategije, infrastrukture in produktov** pa se na nivoju 1 precej predvidljivo deli na [3]:

- **Strategijo in poslovno zavezovanje** (*Strategy and Commit*), ki definira procese za podporo infrastrukture in zagotavljanje pod-procesov za upravljanje življenjskih ciklov. Obsega tudi pod-procese za zagotavljanje podpornih strategij za vsa področja delovanja (tržišča, produkti, stranke, storitve ...) in opredeljuje, kako slediti uspešnosti strategije ter uvajati potrebne izboljšave.
- **Upravljanje življenjskega cikla proizvoda** (*Product Lifecycle Management, PLM*) je proces, ki podpira temeljne operacije za generiranje ponudbe na tržišču in zadovoljevanje potreb strank. Definira procese razvoja in uvajanja storitev, neposredna uspešnost procesov življenjskega cikla pa se odraža na procesu zadržanja strank (angl. *customer retention*).
- **Upravljanje življenjskega cikla infrastrukture** (*Infrastructure Lifecycle Management*) je podobno kot PLM proces razvoja in uvajanja, vendar za infrastrukturo, na kateri so implementirane storitve.

⁴ POS (angl. Point of Sale)



Slika 2-1: Model nivoja 0 ogrodja eTOM z entitetami, ki se raztezajo horizontalno skozi procese eTOM

Slika 2-1 prikazuje navedena področja na vrhnjem nivoju (nivoju 0) ogrodja eTOM skupaj z domenami delovanja (tržišče, produkt, stranka, storitev, vir, dobavitelj/partner), ki se raztezajo horizontalno skozi procese organizacije in so podrobneje predstavljene v poglavju 3.1. Te entitete namreč predstavljajo podatkovni, statični vidik ogrodja Frameworx – model SID.

Sedaj, ko smo spoznali, kako je ogrodje eTOM organizirano in katere funkcionalnosti se razširjajo v njegovi horizontali, pa pogledjmo, kateri so ključni doprinosi njegove uporabe: [4]

- uvaja standardno strukturo, terminologijo in klasifikacijo za opisovanje poslovnih procesov in predstavitev njihovih gradnikov,
- predstavlja temelj za vzpostavitev discipline pri razvoju poslovnih procesov v celotni organizaciji,
- predstavlja osnovo za razumevanje in upravljanje portfeljev aplikacij informacijske tehnologije s stališča zahtev poslovnih procesov,
- omogoča implementacijo konsistentnih in kvalitetnih procesnih tokov z možnostjo rezanja stroškov ali izboljšave implementacij in ponovne uporabe obstoječih procesov ter sistemov
- njegova uporaba v industriji povečuje možnost, da so že serijsko izdelane aplikacije razvite tako, da omogočajo organizaciji enostavno integracijo v svojo informacijsko platformo, kar je seveda ceneje od po meri razvite programske opreme,
- strukturirana narava omogoča detekcijo vrzeli v poslovnih procesih in podvajanja aktivnosti, kar omogoča enostavnejšo optimizacijo poslovnih procesov. Optimizacija lahko poteka tudi na način, ki ga ne predpisuje ogrodje, ampak je izvedena kot podaljšek ogrodja na nižjih nivojih (npr. na dekompoziciji tretjega ali četrtega nivoja).

2.1.3. Osnovna umestitev modela SID in njegov namen

Kot je že iz imena razvidno, je *Shared Information/Data model* na prvem mestu informacijski model, iz katerega kasneje razvijemo podatkovni model, kar bomo podrobneje spoznali v naslednjih podpoglavjih.

SID je neodvisen od platform, programskih jezikov in protokolov. V začetku leta 2013 v trenutno aktualni verziji XII. vsebuje 1364 razredov, organiziranih v 282 paketov (vključno s paketi z metapodatki). [5]

Kot pravi TM Forum [6], je bilo ogrodje SID razvito z namenom, da se sorodni podatkovni koncepti lahko enostavno aplicirajo na poslovne procese, ki potekajo v organizaciji in omogočajo vpogled na deljene podatke brez redundance.

Ker je SID informacijski model, so po TM Forumu osnovni nameni zanj enaki kot za uvedbo in uporabo informacijskega modela [7]:

- uporabi se kot začetna točka pri modeliranju podatkovnih shem, načrtovanju aplikacij in sporočilih med programskimi moduli,
- pomaga definirati skupno terminologijo poslovne domene,
- pomaga pri transformaciji poslovnega procesa, da je ta čim hitrejša in čim cenejša,
- pomaga pri razumevanju konceptov poslovanja in relacij med poslovnimi procesi,
- je vir navdiha – za svež pogled na tradicionalne prakse.

Dodatno pa je SID uporaben tudi za formalne specifikacije za izgradnjo komponent, ki so skladne s Frameworkom.

2.1.4. Informacijski model

V kompleksnih organizacijskih sistemih z več področji delovanja, kjer poteka veliko poslovnih procesov skozi različne domene, je predstavitev informacijske domene in izdelava načrta informacijskih sistemov zahtevna naloga, ki jo rešimo z definicijo informacijskega modela.

Informacijski model je predstavitev konceptov, relacij, omejitev, pravil in aktivnosti za določitev podatkovne semantike izbranega področja obravnave. Določa relacije med posameznimi tipi podatkov, lahko se pa osredotoča tudi na posamezne instance in predpisuje stabilno, organizirano in deljivo arhitekturo podatkov ali znanja za neko domeno. [8]

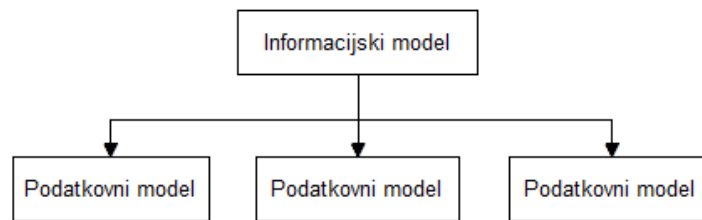
Informacijski model se od podatkovnega razlikuje v tem, da je abstrakcija, ki ponuja širši pogled na problematiko. Raje kot globino problema poskuša predstaviti celoten okvir delovanja.

Poleg modela SID je med bolj znanimi informacijskimi modeli še *Common Information Model* (CIM) organizacije *Distributed Management Task Force*. CIM je v bistvu splošna množica priporočil, iz katere so izpeljani bolj specifični informacijski modeli za posamezne domene. Na tem mestu velja omeniti, da SID ne izhaja iz CIM, ampak je bil razvit ločeno.

2.1.5. Podatkovni model

Podatkovni model eksplicitno determinira strukturo podatkov in definira objekte, ki so predmet načrtovanja na nižjem nivoju abstrakcije ter vsebujejo konkretnije implementacijske podrobnosti, kot so npr. konceptualni ali logični model baze podatkov in načini izmenjevanja podatkov. Nekateri podatkovni modeli tako predpisujejo attribute, kot je npr. število maksimalnih dostopov ali kje je potrebno dodati indeksiranje na podatkovni bazi [9]. Iz navedenega je razvidno, da je na podlagi enega informacijskega modela možno razviti več različnih podatkovnih modelov, kar ponazarja tudi slika 2-2.

Podatkovni modeli so po navadi realizirani s pomočjo jezikov za podatkovno modeliranje (npr. ER diagrami).

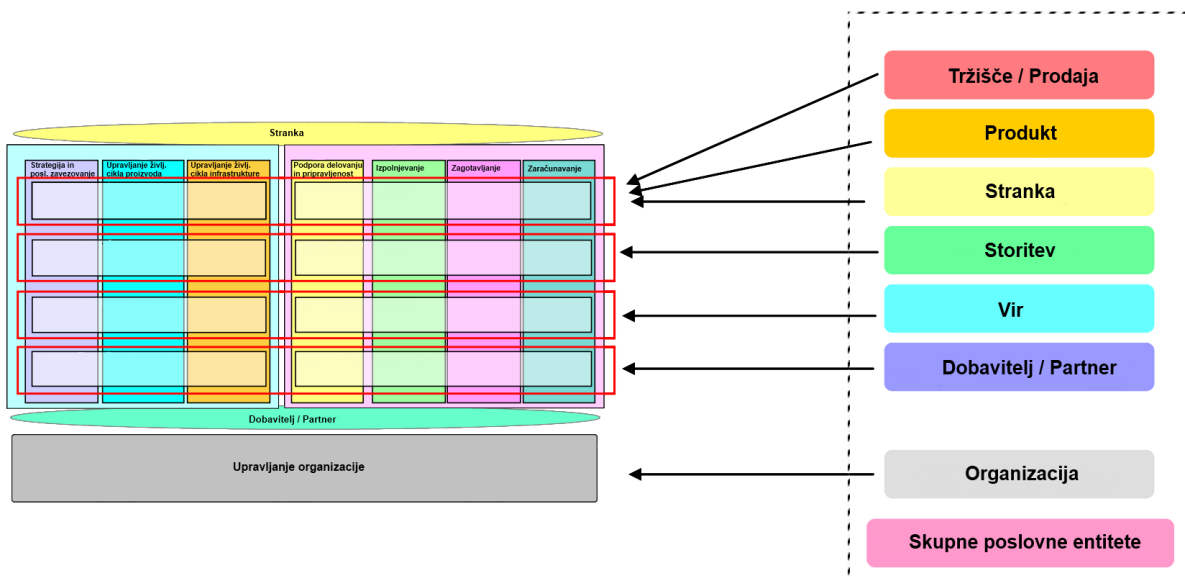


Slika 2-2: Razmerje med informacijskim in podatkovnim modelom

2.1.6. Povezava SID z eTOM

SID in eTOM predstavljata modeliranje iz dveh različnih zornih kotov. eTOM se osredotoča na poslovni proces, torej dinamični vidik poslovanja, medtem ko se SID analitični model osredotoča na statični vidik poslovanja – tj. na entitete, ki se pomikajo skozi poslovni proces, ter na njihove relacije in karakteristike. [7]

Večnivojsko ogrodje SID na vrhnjem nivoju vsebuje **skupek domen**, ki so usklajene z ogrodjem eTOM za poslovne procese, kot to prikazuje spodnja slika. Na levi strani je vidna dekompozicija sedmih eTOM vertikal na nivoju 1 (podrobnejši prikaz kot na sliki 2-1, kjer je prikazan zgolj nivo 0), na desni strani pa so prikazane domene ogrodja SID, ki so predmet obravnave v procesih, ki jih definira eTOM in katere bom podrobneje razdelal v poglavju 3. *Podroben pregled ogrodja SID.*



Slika 2-3: Povezava med eTOM procesnimi vertikalami (levo) in domenami poslovnega pogleda SID (desno) [10]

2.2. Spletne storitve

W3C⁵ opredeljuje spletne storitve kot programsko opremo, ki podpira medsebojno združljivost delovanja (angl. interoperability) med sistemi v omrežju. Sistem gosti spletno storitev preko vmesnika, ki je opisan v standardiziranem formatu, razumljivem odjemalcu – v jeziku za opisovanje spletnih storitev (angl. *Web Service Definition Language* ali WSDL). Sistemi se povezujejo s spletnimi storitvami s pomočjo sporočil SOAP, najpogosteje preko protokola HTTP s pomočjo XML serializacije⁶ in v kombinaciji z ostalimi spletnimi standardi. [11]

V nadaljnjih podpoglavjih bom razdelal ključne, že omenjene, tehnologije, ki jih je potrebno razumeti v povezavi s spletnimi storitvami. Vir za spodnja podpoglavja je knjiga Matjaža B. Juriča in drugih avtorjev [12] in specifikacij W3C za SOAP [13] ter WSDL [14].

2.2.1. Storitveno usmerjena arhitektura

Običajno informacijski sistemi na področju telekomunikacij sestojijo iz več aplikacij. Te so po izvoru lahko zgrajene rešitve po meri znotraj podjetja, po meri zgrajene aplikacije zunanjih izvajalcev ali komercialni sistemi, vsi pa lahko uporabljajo različne arhitekturne pristope, tehnologije ali so implementirani v različnih programskih jezikih.

Posledično je seveda malo verjetno, da so bili vsi sistemi zasnovani kot del enotne arhitekture in podatkovnega modela, ki bi omogočala enovit in fleksibilen informacijski sistem. Ker je poslovanje podjetja strogo odvisno od teh aplikacij samih, je njihova hitra zamenjava nemogoča. Tako je potrebno najti način, kako izmenjevati podatke med različnimi sistemi in odgovor na ta izziv je servisno orientirana arhitektura (angl. *Service Oriented Architecture*), v nadaljevanju SOA.

SOA omogoča šibko sklopljenost arhitekture sistemov in zato omogoča vzporedni razvoj na nivoju podsistemov, kar zmanjšuje potreben odzivni čas na spremembe v poslovnih procesih in omogoča doseganje večje fleksibilnosti.

Z vidika oddelkov IT so ključni doprinosi SOA sledeči:

- zaradi nenehnih sprememb v poslovnih procesih je lažje zagotoviti kontinuirane spremembe informacijskih sistemov in možno zmanjšati negativne posledice sprememb (npr. napake v programski kodi),
- zagotavlja nam tehnološko neodvisnost,
- omogoča uporabo že razvitih rešitev drugim sistemom,
- omogoča, da se znebimo podvojenih podatkov v sistemih tako, da se za vsako domeno uvede nosilni sistem, ki ponuja podatke odjemalcem,
- omogoča, da se znebimo podvojenih funkcionalnosti v posameznih aplikacijah tako, da izpostavimo funkcionalnosti kot spletne storitve,
- s povečevanjem naprav in platform (osebni računalniki, tablice, mobilni telefoni ...), ki dostopajo do storitev, je mogoče ponuditi storitve skozi različne kanale,
- lažje ločevanje odgovornosti, ko aplikacije razvijajo zunanji podizvajalci,
- možnost razvoja storitvenega omrežja, ki lahko nudi storitve tudi izven organizacije, kar odpira nove možnosti za uporabo informacijskih tehnologij za optimizacijo poslovanja,
- omogoča osredotočanje na vsebino storitev, namesto na tehnologije.

⁵ World Wide Web Consortium – konzorcij za svetovni splet, ki skrbi za standardizacijo tehnologij na svojem področju.

⁶ Proces pretvorbe podatkovne strukture ali objekta v format, ki ga je možno shraniti (npr. v datoteko) ali poslati preko omrežja

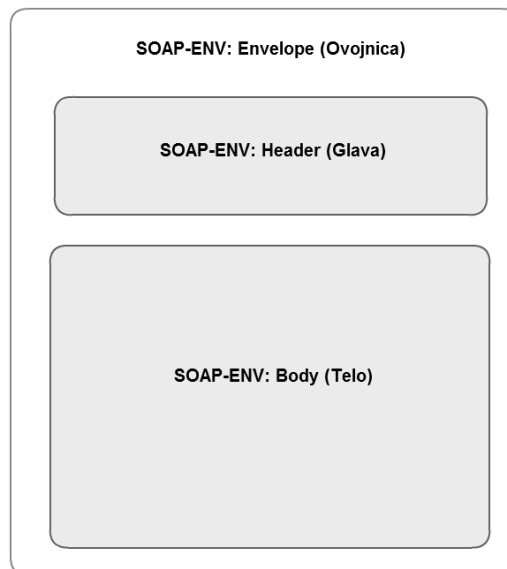
2.2.2. Standard za spletne storitve SOAP

Simple Object Access Protocol (SOAP) je protokol za izmenjavo podatkov za spletne storitve. Podatki za izmenjavo so predstavljeni v strukturiranem formatu XML⁷, standard za komunikacijo pa je HTTP protokol, čeprav ga je mogoče zamenjati s katerim izmed ostalih visokonivojskih protokolov (npr. SMTP⁸, JMS⁹).

Po [15] sta za izbiro protokola HTTP ključna naslednja razloga:

- pri uporabi protokola HTTP nimamo veliko težav s požarnimi zidovi, saj je načeloma promet zanj dovoljen,
- HTTP in XML sta odprtokodna standarda, kar omogoča veliko orodij (tudi brezplačnih) za delo in upravljanje z njima.

Sporočilo SOAP je sestavljeno iz naslednjih delov: ovojnice SOAP, glave SOAP in telesa SOAP, kar je prikazano na spodnji sliki. Telo SOAP lahko vsebuje opsijski element o morebitni napaki SOAP, v primeru, da je prišlo do nje.



Slika 2-4: Struktura sporočila SOAP

Ovojnica SOAP (angl. SOAP envelope) je obvezen del sporočila SOAP, v katerem je podan imenski prostor dokumenta XML. Ovojnica kot najbolj zunanji element sporočila XML vsebuje glavo in telo.

Glava SOAP (angl. SOAP header) je neobvezen element in služi informacijam o specifičnih aplikacijah ali informacijam o varnosti. V primeru, da je glava prisotna, mora biti prvi pod-element ovojnice. Glava ima tudi nekatere predhodno definirane attribute. Tak je npr. *mustUnderstand*, ki v primeru, da je njegova vrednost enaka 1, označuje, da ga mora odjemalec spoznati in upoštevati. V nasprotnem primeru je rezultat sprejetja sporočila napaka.

Telo SOAP (angl. SOAP body) je dejansko sporočilo, ki je namenjeno končnemu prejemniku sporočila.

⁷ XML – angl. Extensible Markup Language, (slo. »razširljivi označevalni jezik«)

⁸ SMTP – Simple Mail Transfer Protocol

⁹ JMS – Java Message Service

Napaka SOAP (angl. SOAP fault) je opcijski element telesa SOAP, ki se lahko pojavi zgolj enkrat. Vsebuje sporočilo ob morebitni napaki. V primeru, ko prejemnik ne razume SOAP sporočila, za katerega je bilo to zahtevano, bo odjemalec vrnil napako.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
http://www.operater.si/FindResource/findResourceRequest</Action>
    <Security s:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-13">
        <wsse:Username>OMS</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">xxxxx</wsse:Password>
        <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-security-1.0#Base64Binary">wfz4pQ+EB6igJV0yB1pLEg==</wsse:Nonce>
      </wsse:UsernameToken>
    </Security >
  </s:Header>
  <s:Body>
    <findResource xmlns="http://www.mobitel.si/extension/inventory/message/interface">
      <resource xmlns="">
        <resource xsi:type="ql:LogicalResource">
          <Name>MSISDN</Name>
          <CharacteristicValue>
            <Characteristic>
              <Name>ROWCOUNT</Name>
            </Characteristic>
            <Value>5</Value>
          </CharacteristicValue>
          <CharacteristicValue>
            <Characteristic>
              <Name>PREFIX</Name>
            </Characteristic>
            <Value>2</Value>
          </CharacteristicValue>
          <CharacteristicValue>
            <Characteristic>
              <Name>RESERVE</Name>
            </Characteristic>
            <Value>False;2;Burton;1661566;20e4f2b5-bd97-4124-9f04-c0ecf142c5a1</Value>
          </CharacteristicValue>
          <dealerId>1661566</dealerId>
        </resource>
      </resource>
    </findResource>
  </s:Body>
</s:Envelope>
```

Primer 2-1: SOAP ovojnica s svojimi pod-elementi za spletno storitev *findResource*

2.2.3. Jezik za opisovanje spletnih storitev

Jezik za opisovanje spletnih storitev WSDL (angl. *Web Services Description Language*) je skupek definicij o spletni storitvi, ki jih potrebuje odjemalec zato, da zna uporabljati storitev. Te definicije se običajno generirajo avtomatsko, z orodji, ki so razvita v ta namen.

WSDL je razvit na osnovi XML in definira predpisane elemente, ki odjemalcem služijo za razumevanje spletne storitve. Korenski element XML je *definitions*, ki vsebuje ostalih 6 glavnih elementov:

- *types* – definira podatkovne tipe, ki so predmet izmenjave podatkov (bodisi vhodni ali izhodni podatki). V tem elementu najdemo definicije ali povezave do definicij XSD¹⁰,
- *message* – definicija sporočila, ki ga sprejme ali odpošlje spletna storitev preko operacije,

¹⁰ XML Schema Definition je jezik za implementacijo omejitev v dokumentih XML, iz katerih je možno generirati razrede za programske jezike (C#, Java ...), ki nato služijo implementaciji spletnih storitev

- *portType* – opisuje nabor podprtih operacij določene spletne storitve z opisom vhodnih in izhodnih sporočil,
- *binding* – specificira dejanski protokol in format podatkov za posamezni *portType*,
- *port* – označuje naslov končne točke (angl. endpoint) spletne storitve,
- *service* – se uporablja za združevanje sorodnih elementov *port*.

2.3. Poslovni procesi in njihovo upravljanje

2.3.1. Opredelitev poslovnega procesa

Eden prvih in vidnejših pionirjev ekonomije, Adam Smith, je svojo ugotovitev, kako lahko delitev dela poveča storilnost v organizaciji, predstavil na primeru v manufakturi bucik:

En mož žico razteguje, drugi jo ravna, tretji reže, četrti ostri, peti jo brusi pri vrhu, kjer bo pritrjena glavica, katere izdelava terja dva ali tri različne postopke; še zabadanje v papir je obrt zase; zahtevno izdelovanje bucik je tako razdeljeno na osemnajst ločenih postopkov, ki jih v nekaterih manufakturah vse po vrsti opravljajo različni delavci, čeprav v drugih dva ali tri izmed njih včasih izvaja en sam mož. [16]

Smith je pravzaprav vsebinsko že predstavil *poslovni proces* (čeprav ga ni poimenoval tako), v sodobnejšem času pa je sledila množica definicij različnih avtorjev. Zanimivejša je recimo definicija po Khanu [17], ki pravi, da je poslovni proces skupek zaporednih in vzporednih nalog, ki jih izvajajo ljudje ali aplikacije z namenom, da bi dosegli skupni cilj.

Različni avtorji opredeljujejo poslovni proces različno, a kar je skupno vsem definicijam, je to, da ima poslovni proces vhod in izhod ter da je namen ustvarjanje neke dodane vrednosti [18].

2.3.2. Upravljanje poslovnih procesov

V devetdesetih letih prejšnjega stoletja, ko se je raven informacijske podpore znatno dvignila, je veliko podjetij naletelo na težave pri uvajanju izboljšav v informacijsko podporo lastne organizacije. Kot upravljavski predlog za rešitev teh težav se je pojavil pristop, imenovan *prenova poslovnih procesov* (angl. *Business process reengineering*) ali BPR. Prenova poslovnih procesov temelji na celoviti prenovi poslovanja in zajema prenavo poslovnih procesov na vseh nivojih organizacije. Tak pristop se je sčasoma izkazal za izredno tveganega in leta 1994 je bilo ugotovljeno, da kar 70 odstotkov projektov BPR ne uspe. [15]

Takšna ugotovitev je seveda terjala ponoven razmislek o samem pristopu k izboljšavi informacijske podpore z vidika podpore poslovnim procesom in tako je nastal pristop, imenovan *upravljanje poslovnih procesov* ali BPM (iz angl. *Business process management*). Ta je manj radikalen kot BPR – temeljna razlika med njima je namreč to, da BPM predvideva izboljšave v več iteracijah. Bistvene razlike so razvidne iz naslednje tabele.

Tabela 2-1: Razlike med BPR in BPM [19]

Dejavniki	Prenova poslovnih procesov	Upravljanje poslovnih procesov
Raven sprememb	korenite – celotni procesi	celoten poslovni cikel
Izhodiščna točka	neobremenjeno s preteklostjo	novi ali obstoječi procesi
Pogostnost sprememb	enkratne ali občasne	enkratne, občasne, stalne ali razvojne
Čas izvajanja	dolg	v realnem času
Izvajanje	prelomno, takojšnja in korenita prenova	postopno
Sodelovanje in izvedba	od vrha navzdol	oboje – od vrha navzdol in od spodaj navzgor
Področje obravnave	široko, medfunkcijsko	celovito upravljanje s procesi organizacije
Usmeritev	v prihodnost	v preteklost, sedanost in prihodnost
Tveganje	visoko	nizko
Poglavitni pospeševalec	informacijska tehnologija	procesna tehnologija
Orodja	modeliranje procesov	različna
Izvajalci prenove	splošni poznavalci poslovanja	specialisti za prenovu procesov, z vsemi zaposlenimi
Izvedba sprememb	proces	proces in poslovna praksa

Upravljanje poslovnih procesov (angl. *Business process management*) je torej celovit pristop k prilagoditvi poslovnih procesov potrebam in zahtevam strank in je tako proces, ki zagotavlja nenehno izboljševanje poslovanja podjetja. Programska oprema, ki nudi podporo temu procesu, se imenuje **sistem za upravljanje poslovnih procesov**.

2.3.3. Sistemi za upravljanje poslovnih procesov

Kot v delu [20] ugotavlja avtor, je osrednja značilnost sistemov za upravljanje poslovnih procesov (angl. *Business process management system*) sposobnost avtomatizacije procesov, kar je bistveni razlog za zmanjšanje t. i. »mrtvega« časa v poslovnem procesu. Mrtvi čas je čas, ko izvedba opravil v poslovnem procesu stoji.

Pri tipičnih poslovnih procesih je mrtvega časa kar do 90 % trajanja celotnega poslovnega procesa, z uporabo BPMS pa lahko tega bistveno zmanjšamo, saj naloge hitreje prihajajo do odgovornih oseb v izvrševanje. Za te naloge je tudi znano, v katerem koraku so, kako dolgo traja kateri korak v procesu in do kdaj ga je potrebno zaključiti. Vpogled v omenjene dejavnike omogoča boljši nadzor nad poslovnim procesom in lahko bistveno pripomore k njegovi optimizaciji.

Z avtomatizacijo poslovnega procesa lahko določena ročna opravila povsem avtomatiziramo in jih prepustimo za to namenjeni aplikaciji. To avtomatizacijo opravil dosežemo z integracijami med sistemi ter z uporabo spletnih storitev in storitveno usmerjene arhitekture.

Korake v poslovnem procesu, v katerih se odvijajo integracije in za njihov zaključek ni potrebna interakcija z uporabnikom, imenujemo *avtomatski koraki*, tiste, kjer je potrebno uporabnikovo posredovanje, pa imenujemo *človeški koraki*. Avtomatski koraki se od človeških razlikujejo v tem, da jih je običajno možno zaključiti brez večjih zakasnitev zaradi čakanja na izvedbo, zato večje število avtomatskih korakov načeloma pomeni tudi hitrejšo izvedbo poslovnega procesa. Implementacija sistema z velikim številom avtomatskih korakov lahko pomeni bistveno hitrejšo izvedbo poslovnega procesa.

Glede na omenjeno je precej odvisno, s kakšnimi poslovnimi procesi imamo opravka, saj so različni BPMS različno učinkoviti glede na potrebe. Tako poznamo *človeško naravnane BPMS*, ki stremijo predvsem k izboljšanju kakovosti, učinkovitosti in hitrosti človeškega dela, nekatere podskupine teh sistemov pa pridejo posebej v poštev v primeru, ko pravila za izvajanje akcij uporabnika niso dobro definirana. Na drugi strani imamo *integracijsko naravnane BPMS*, ki učinkoviteje zadovoljujejo potrebe po integraciji našega poslovnega procesa z drugimi sistemi in njihovimi procesi [15].

V telekomunikacijski domeni imamo v procesu naročanja dobro definirana pravila za potek poslovnega procesa naročanja storitev in stremimo k čim večji avtomatizaciji poslovnih procesov, zato je načeloma boljša izbira integracijsko naravnane BPMS.

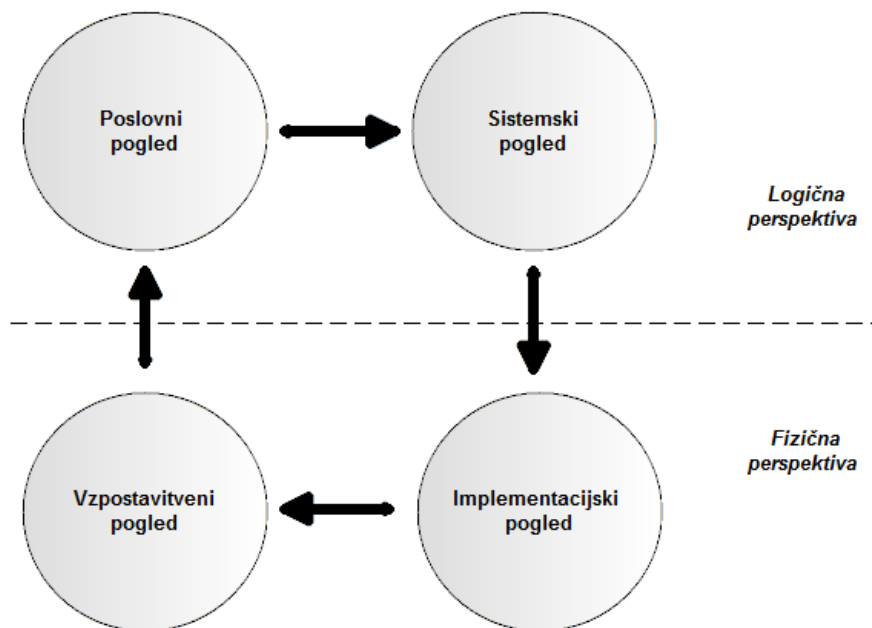
3. Podroben pregled ogrodja SID

Ogrodje SID je komponenta Frameworkxa, ki ponuja standarden model za predstavitev informacij, ki so v uporabi znotraj organizacije, med ponudniki storitev in njihovimi poslovnimi partnerji.

SID ponuja standardni (referenčni) model in slovar izrazov za informacijski model, ki je potreben za implementacijo eTOM procesov. Tako zmanjšuje kompleksnost procesa načrtovanja in razvoja informacijskih sistemov za nudenje storitev ter integracij med njimi na tak način, da je sprejemljiv za vse vpletene.

Frameworkx pristop k razvoju modela SID je prikazan na spodnji sliki. Vsak pogled se drugače osredotoča na skupek SID artefaktov, ki se razvijajo iz pogleda v pogled:

- **Poslovni pogled** (angl. Business View) opisuje konceptualni ali logični načrt informacijske arhitekture s poslovne perspektive.
- **Sistemske pogled** (angl. System View) dodaja vedenjske vidike in začenja definirati, kako bodo entitete sodelovale kot enota.
- **Implementacijski pogled** (angl. Implementation View) opisuje dejanske podrobnosti implementacije, kot so definicije na nivoju podatkovne baze in vmesnikov API.
- **Vzpostavitveni pogled** (angl. Deployment View) prikazuje vpeljavo informacijske arhitekture kot funkcionalne storitve (npr. podatkovne baze ali vmesniki API).



Slika 3-1: NGOSS pogledi

Iz slike 3-1 je razvidna tudi ločnica med logično in fizično perspektivo. Ločnica med njima in njena identifikacija sta pomembni zaradi dejstva, da je iz enega logičnega modela SID možno izpeljati več fizičnih implementacij. To pomeni, da mora v neki točki razvoja logične perspektive arhitekt SID sprejeti odločitev, ali je vredno narediti spremembe, ki so specifične za implementacijo, ali pa želi, da je implementacija čim bližje čisti logični perspektivi in je zato bolj generična. Ločnica med logično in fizično perspektivo načeloma pomeni tudi prehod iz informacijskega modela v podatkovnega.

V nadaljnjih podpoglavjih bomo predstavili poslovni in sistemski pogled ter pomembne koncepte okrog njiju. Implementacijski in vzpostaviteni pogled pa bomo obdelali v praktičnem delu (5. poglavje). Naslednja tri podpoglavja so povzeta po knjigi TM Foruma [10].

3.1. Domene delovanja

Domene, ki sestavljajo ogrodje SID, so pravzaprav zbirka entitet, ki so povezane z določenim področjem upravljanja v organizaciji. Te domene so konsistentne s koncepti procesov, kot jih definira eTOM na nivoju 0:

- **Tržišče in prodaja** (*Market/Sales*) – se osredotoča na prodajne in marketinške aktivnosti, ki so potrebne za pridobivanje in zadržanje strank.
- **Produkt** (*Product*) – predstavlja ponudbo na tržišču, kar zajema razvoj ponudbe, oblikovanje cen in akcij, oblikovanje paketov in določanje njihovega življenjskega cikla ter merjenje učinkovitosti prodaje produktov.
- **Stranka** (*Customer*) – podpora vodenju in administraciji fizičnih in pravnih oseb, zaračunavanje strankam ter prejemanje in odločanje o prejetih zahtevkih strank.
- **Storitev** (*Service*) – tehnična domena; vsebuje entitete, ki se ukvarjajo z definicijami, razvojem in upravljanjem storitev telekomunikacijskega podjetja.
- **Vir** (*Resource*) – tehnična domena, ki zagotavlja operativnost. Ta domena zagotavlja izvedbo storitve, ki jo nudi podjetje, ter povezuje vire s storitvami in produkti.
- **Dobavitelj in partner** (*Supplier/Partner*) – upravljanje s partnerji, dobavo in planiranjem v zvezi z dobavitelji in poslovnimi partnerji.
- **Organizacija** (*Enterprise*) – podpora interakcijam in procesom znotraj organizacije. Ta domena je še v razvoju in še nima povsem definiranih agregiranih poslovnih entitet.
- **Skupne poslovne entitete** (*Common Business*) – niso posebej definirane v eTOM procesih, ampak jih uvaja ogrodje SID z namenom, da določi entitete, ki so povezane z vsaj dvema domenama. Kot primer skupne entitete lahko navedemo pogodbo (*Agreement*), ki je uporabljena v povezavi z domenami stranke, dobaviteljev in partnerjev.

Da lahko dobro identificiramo potrebe po informacijsko-podatkovnem modelu, je znotraj vsake domene treba identificirati t. i. **agregirane poslovne entitete** (*Aggregate Business Entities*), v nadaljevanju tudi ABE. Domena delovanja je pravzaprav skupek ABE, ki so povezane z določenim področjem v organizaciji in so bile identificirane v okviru poslovnega pregleda organizacije.

3.2. Agregirane poslovne entitete

Poslovne entitete predstavljajo entitete, ki so lahko:

- stvarne, npr. stranka (*Customer*),
- aktivne, npr. naročilo stranke (*CustomerOrder*),
- konceptualne, npr. uporabniški račun (*CustomerAccount*).

Poslovne entitete imajo attribute in sodelujejo z ostalimi poslovnimi entitetami, njihovi primerki pa se navadno premikajo skozi nek vnaprej definiran življenjski cikel.

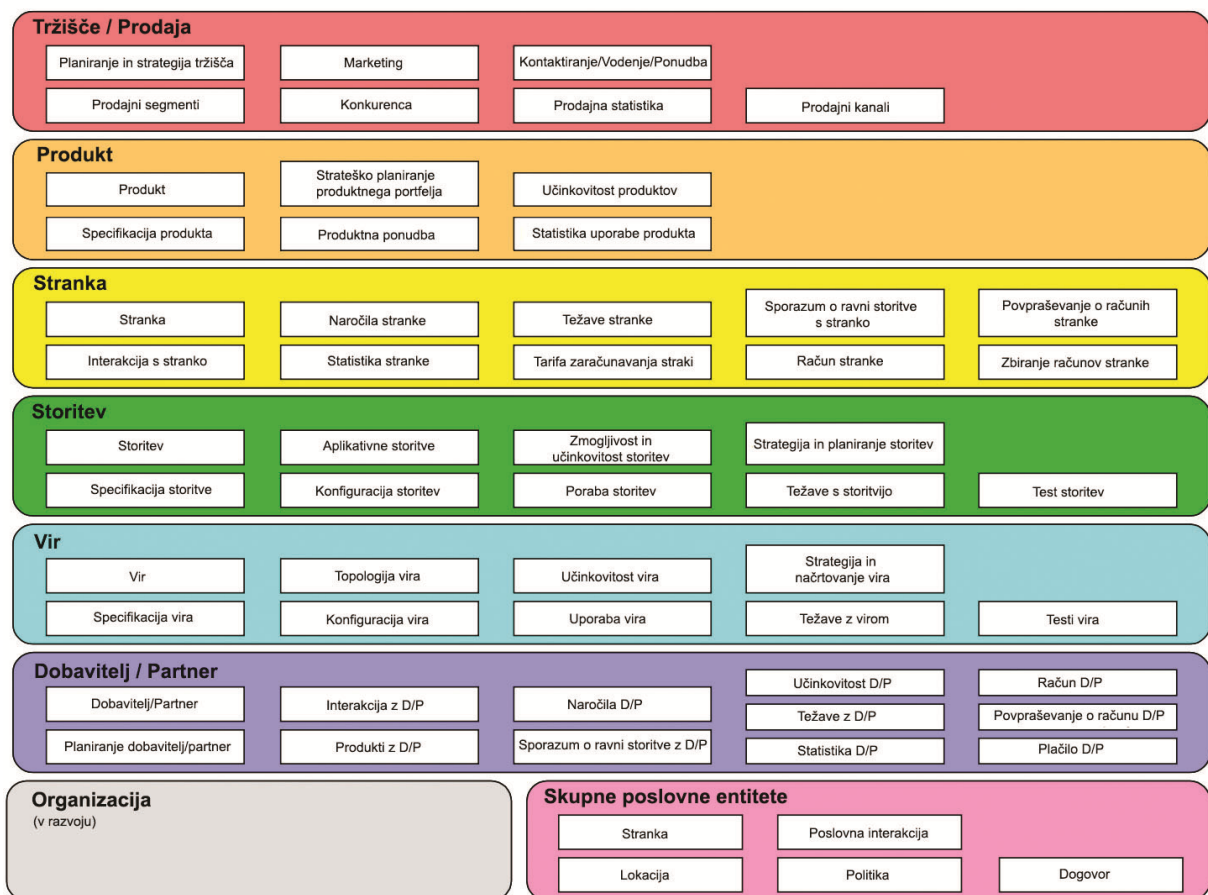
Agregirana poslovna entiteta je dobro definirana množica informacij in operacij, ki opisuje visoko povezano, a šibko sklopljeno množico poslovnih entitet. To omogoča segmentacijo celotnega obsega poslovnih problemov, hkrati pa omogoča, da so storitve in viri osredotočeni na točno določeno domeno delovanja.

To lahko ponazorimo s primerom opreme, torej (fizičnega) vira, s katerim upravlja aplikacija za inventar. Ta nudi podatke ostalim odjemalcem, ki skrbijo za druge vidike poslovanja. Tako

podatke o opremi uporablja tudi sistem za upravljanje z naročili v procesu oddaje naročila stranke, ki mora preveriti, ali je serijska številka pravilna in če je fizično na prodajnem mestu, ki želi opraviti prodajo. Sistem za upravljanje storitev pa mora v procesu provizioniranja aktivirati opremo na podlagi njene serijske številke in tako omogočiti uporabo storitve.

V navedenem primeru je šibka sklopljenost med podsistemi razvidna iz primera opreme (domena vira) in strankinega naročila (domena stranke), ki sta upravljana vsak v ločenem sistemu, visoka stopnja kohezije pa je razvidna iz dejstva, da je provizioniranje neke storitve neposredno odvisno od same opreme.

V ogrodju SID so ABE uporabljene z namenom prikaza poslovnih konceptov. Slika 3-2 prikazuje agregirane poslovne entitete ogrodja SID na najvišjem nivoju, umeščene v posamezne domene.



Slika 3-2: domene SID in agregirane poslovne entitete prvega nivoja za vsako domeno, povzete po TM Forumu [21]

Za namen tega diplomskega dela sta posebej zanimivi domeni produkta in storitev.

3.2.1. ABE za domeno produkta

V prejšnjem poglavju smo v grobem predstavili, kaj so odgovornosti domene produkta, sedaj pa bomo odgovornost bolj podrobno razdelali in predstavili glavne poslovne entitete te domene.

1. **Strateško planiranje produktnega portfelja** se osredotoča na razvoj ponudbe na tržišču z analizo tega, katerim segmentom tržišča ponuditi katere produkte, ter ugotavlja, kdaj in kako umakniti produkte s tržišča ter jih zamenjati z novimi.
2. **Specifikacija produkta** (*Product Specification*) definira funkcionalnosti in karakteristike produktne ponudbe, ki je ustvarjena za tržišče.

3. **Produktna ponudba** (*Product Offering*) predstavlja stvarne in nestvarne produkte in storitve, ki se ponujajo na tržišču za določeno ceno v produktnem katalogu.
4. **Produkt** predstavlja instanco produktne ponudbe, na katero je prijavljena dejanska stranka, ima določene vrednosti karakteristik in lokacijo uporabe. Ta ABE tudi hrani storitve in vire, ki so uporabljeni za njegovo realizacijo.
5. **Statistika uporabe produkta** skrbi za spremljanje trendov produktov iz prejšnje točke.
6. **Učinkovitost produktov** se ukvarja s spremljanjem realizacije poslovnih ciljev in ugotavlja, ali je bilo tem ciljem zadoščeno. Lahko se ukvarja tudi z merjenjem zmogljivosti in odkrivanjem morebitnih težav.

3.2.2. ABE za domeno storitev

1. ABE za **specifikacijo storitve** vsebujejo entitete, ki definirajo karakteristike in posledično njihovo obnašanje. To omogoča, da so iz ene specifikacije storitve lahko izpeljane mnoge, torej specifikacija storitve služi kot predloga za izpeljavo **storitev**. Te entitete skrbijo tudi za povezljivost s produktom (*Product*) in jih omogočajo viri (*Resource*).
2. **Storitev** (*Service*) je izpeljana iz specifikacije storitev, njene ABE pa predstavljajo različne informacije za pregled, analizo, konfiguracijo in nadzor storitev. Ker vsebuje entitete, ki so pomembne za razumevanje nadaljnjih poglavij, si pogledjmo še **drugi nivo ABE za storitev**:
 - a. **Storitve k stranki** (*CustomerFacingService, tudi CFS*): abstrakcija, ki definira karakteristike in obnašanje neke storitve, kot jo vidi stranka, kar pomeni da stranka »kupi« te storitve in se jih neposredno zaveda. Primeri storitev k stranki so *internet ADSL, televizija DVB-T* ali *telefonija GSM*. Nasprotje »storitev k strankam« so »storitve k virom«.
 - b. **Storitve k virom** (*ResourceFacingService, tudi RFS*): abstrakcija, ki definira karakteristike in obnašanje storitev, ki so zanemarljive za stranko in so načeloma vezane na vire oz. so nestvarna storitev, ki je vezana na sistem za oskrbovanje storitev (provizioniranje). Kot primer RFS lahko navedemo *modem* (za CFS *internet*) ali pa *blokada dohodnih klicev* (za CFS *telefonija GSM*). Kot je razvidno iz primerov, so RFS lastnost CFS.
3. **Aplikativne storitve** definirajo ABE za realizacijo storitev kot aplikacij. Te storitve so lahko zelo različnih tipov, zato so ABE povezane z velikim številom preostalih entitet. Primeri takih aplikacij so konfiguracija za zagotavljanje kakovosti storitev, VPN ali celo aplikacije za zagotavljanje storitev VoIP.
4. **Konfiguracija storitev** vsebuje entitete, ki skrbijo za definicijo strukture storitev (CFS in RFS) in predpisujejo pravila, kako se ta struktura lahko spremeni. Predpisujejo hierarhijo tega, kaj je storitev in kaj so njene pod-storitve.
5. Skupek entitet za nadzor **zmogljivosti in učinkovitosti storitev** skrbi za zbiranje in povezovanje podatkov, ki se uporabljajo v statistične namene, za nadzor sistema in generiranje poročil. Rezultati teh se uporabljajo za primerjavo z zastavljenimi cilji, primerjavo s trendi, merjenje napak ipd. Ker te entitete hranijo podatke za posamezno storitev (bodisi CFS ali RFS), so zelo uporabne za spremljanje na novo uvedenih tehnologij.
6. **Test storitev** vsebuje ABE, odgovorne za shranjevanje podatkov, ki so namenjeni testiranju CFS ali RFS, recimo pred uvedbo storitve pri stranki, ali ugotavljanju napak.
7. **Težave s storitvijo** definira skupek ABE, ki so potrebne za upravljanje z napakami, alarmi in izpadi storitev (fizičnih ali logičnih, CFS ali RFS).
8. **Poraba storitev** je skupek entitet, ki beležijo uporabljanje storitev (in uporabe omrežja) in izpostavljajo te informacije ostalim poslovnim entitetam, recimo stranki oz. njenemu računu (*CustomerAccount*) za namen zaračunavanja.

9. **Strategija in planiranje storitev** vsebuje entitete, ki so uporabljane za izboljšave in uvajanje novih storitev. Je tesno povezana z ABE produkta in virov.

Z navedenimi entitetami se pretežno ukvarja sistem za upravljanje s storitvami kot tudi sistemi za provizioniranje storitev (teh je več, saj so različni za različne tehnologije, kot npr. GSM ali IP). Za namen povezovanja z OMS so najpomembnejši pojmi (entitete), opisani v prvih dveh točkah tega poglavja.

3.3. Agregirane sistemske entitete

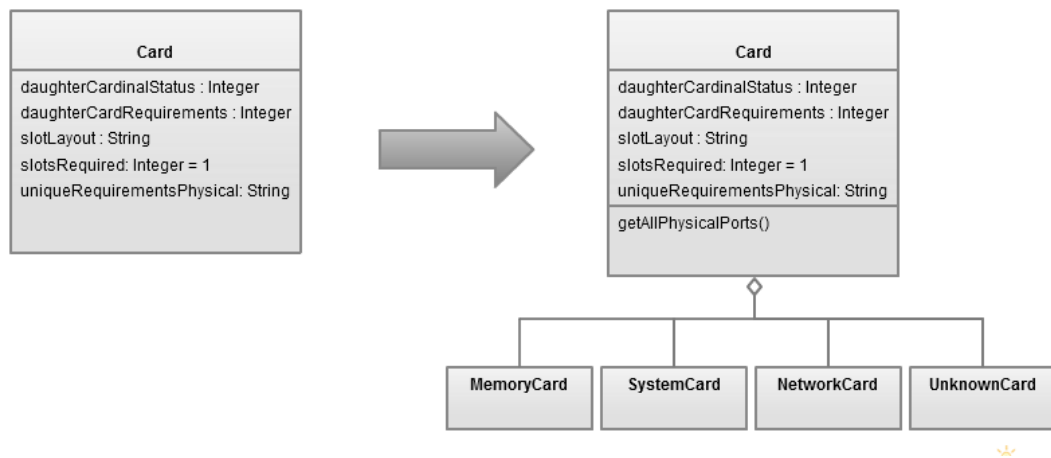
V sistemskem pogledu je osmim domenam na najnižjem, ničtem, nivoju poslovnega pogleda (nivo 0 in 1 sta razvidna iz slike 3-2) dodana deveta – arhitekturna. V splošnem so agregirane sistemske entitete zgolj podaljšek ABE v ustrezni poslovni domeni.

Kot navaja [22], so podaljški lahko različnih tipov, in sicer lahko:

- dodajajo attribute, ki so pomembni na sistemskem nivoju,
- dodajajo sistemske metode,
- transformirajo povezavo v povezovalni razred, ki vsebuje več semantike,
- dodajajo omejitve (omejitve podatkovnih struktur, omejitve operacij in sprožilcev).

Naslednja slika prikazuje primer za entiteto kartice, ki je bila definirana na poslovnem nivoju, in to, kako so ji bili na sistemskem nivoju dodani:

- atributi,
- metoda za pridobivanje prostih vrat (*getAllPhysicalPorts()*),
- dedujoči podrazredi, ki še niso bili pomembni na poslovnem nivoju.



Slika 3-3: Primer podaljška na sistemskem nivoju

3.4. Uporaba programskih vmesnikov in modela SID

3.4.1. Motivacija za uporabo programskih vmesnikov, temelječih na SID

Bistveni doprinos uporabe modela SID za programske vmesnike API je zmanjšanje časovne zahtevnosti za implementacije integracij, kar zmanjša stroške izmenjave podatkov med sistemi.

Ključni razlog za pocenitev tiči v uporabi enake strukture podatkov in podatkovnih tipov ter v dejstvu, da je izrazoslovje med sistemi poenoteno, kar poenostavi komunikacijo med analitiki ali integratorji sistemov, ki so predmet integracij.

V primeru, ko nek sistem uporablja model SID tudi za svoje interno delovanje, uporaba programskih vmesnikov, temelječih na modelu SID, bistveno poenostavi nalogo mapiranja podatkov in podatkovnih struktur.

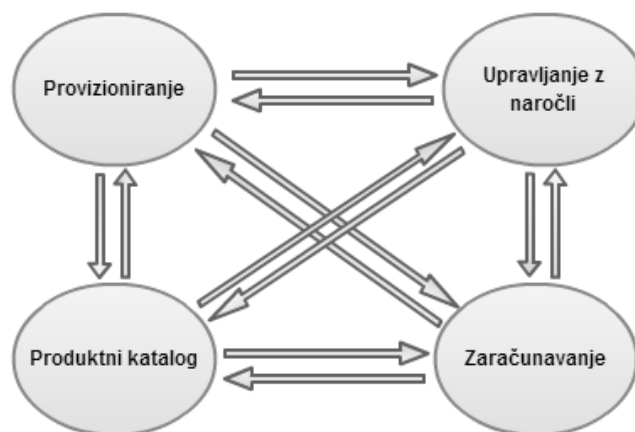
Seveda pa je zelo verjetno, da nek sistem za svoje interno delovanje (podatkovne strukture, podatkovni model,) ne uporablja ogrodja SID ali pa je to zelo razširjeno s specifikami sistema. To ne pomeni, da tak sistem ne more nuditi preostalim sistemom (odjemalcem) programskih vmesnikov, ki temeljijo na ogrodju SID. Ravno nasprotno – TM Forum to spodbuja, saj se v primeru uporabe modela SID interne spremembe v aplikaciji ne odražajo v spremembah na programskem vmesniku. To bi se sicer lahko zgodilo s slabo načrtovanim lastnim vmesnikom, ki odstopa od priporočil ogrodja SID.

Uvedba take arhitekture poveča stroške pri implementaciji integracij zaradi translacije iz in v model SID v samem sistemu, preprečuje pa, da bi spremembe znotraj sistema povzročale spremembe vmesnikov, kar bi v primeru spletnih storitev lahko pomenilo dodatno delo na strani vseh odjemalcev. Dodatno je zelo pomembno, da je med takimi sistemi terminologija poenotena, saj to zmanjšuje kompleksnost izmenjave podatkov med sistemi.

V tem primeru se bo s povečevanjem obsega uporabe modela SID v sistemih telekomunikacijskega podjetja strošek integracij zmanjšal, ta strošek pa seveda ne more povsem izginiti zaradi specifik sistemov in podaljškov modelu SID, ki so implementirani zaradi specifik aplikacij.

Spodnja slika ponazarja štiri sisteme iz telekomunikacijske domene; iz nje je razvidno, da je pri štirih sistemih lahko kar 12 povezav med sistemi. Sicer do maksimalnega števila povezav načeloma ne pride, npr. produktni katalog običajno ne zahteva podatkov od drugih sistemov, ampak le izpostavlja svoje podatke, vendar vseeno pogledjmo, koliko je maksimalnih možnih povezav in koliko jih je približno v povprečju – v primeru, da upoštevamo povezave kot enosmerne.

Da bi ponazorili doprinos omenjene arhitekture programskih vmesnikov, temelječih na ogrodju SID, bomo povzeli besede Andrewa McFadyena [23], ki je predstavil uporabo modela SID kot analogijo z uporabo esperanto jezika.



Slika 3-4: Prikaz maksimalnega števila povezav (12) med štirimi sistemi

Če imamo recimo Angleža in Francoza, ki se želita pogovarjati, potrebuje vsaj eden od njiju (lahko pa oba!) slovar. Če bomo dodali v skupino Nemca, bomo prvotnemu angleško-francoskemu slovarju morali dodati francosko-nemškega in angleško-nemškega. Če se pridruži še Italijan, bo število potrebnih slovarjev naraslo še za 3 (angleško-italijanski, francosko-italijanski, nemško-italijanski) – na skupno 6 slovarjev ali 12 v primeru, da vsakemu damo slovar za njegov jezik. Kot je že razvidno iz vzorca, je z vsakim novim prišlekom (novim vključenim

sistemom) število potrebnih slovarjev večje za število članov skupine pred prihodom novega člana oziroma za dvakratnik tega, če je povezava dvosmerna.

V primeru, ko za medsebojno komunikacijo uvedemo jezik esperanto in prepustimo prevajanje vsakemu posamezniku, je število potrebnih slovarjev enako številu sodelujočih, kar je razvidno iz spodnje tabele.

Tabela 3-1: Število potrebnih translacij za komunikacijo med več sistemi

Število udeležencev	2	3	4	5	6	7	8	9	10
Število enosmernih slovarjev	1	3	6	10	15	21	28	36	45
Število dvosmernih slovarjev	2	6	12	20	30	42	56	72	90
Število esperanto slovarjev	2	3	4	5	6	7	8	9	10

Z dodajanjem sistemov strošek narašča sorazmerno s številom potrebnih translacij. Maksimalno število potrebnih translacij, kjer sistemi ne upodabljajo skupnega podatkovnega modela, kot je npr. SID, torej narašča po formuli

$$T_{max} = N * (N - 1),$$

kjer N predstavlja število sistemov.

3.4.2. Implementacija programskih vmesnikov, temelječih na SID

Za implementacijo programskih vmesnikov, temelječih na modelu SID, so najpogostejše uporabljene tehnologije programski jezik Java, shema XML in spletne storitve.

TM Forum ponuja skozi svoj program *TM Forum and OSS through Java* odprtokodno ogrodje, razvito v programskem jeziku Java, ki je na voljo zastonj.

To pa ne pomeni, da je Java nujna izbira v primeru, da bi želeli uporabiti model SID. Kot alternativo ponuja TM Forum vmesnike v shemi XML (angl. XML Schema, ali XSD). Iz teh pa je mogoče generirati podatkovne strukture v kateremkoli programskem jeziku.

Tako generirani razredi modela SID bodo potem uporabljeni kot vhodni parametri za programske vmesnike, ki so v poslovni domeni najpogostejše spletne storitve, temelječe na protokolu SOAP.

OMS se skozi generirane razrede na podlagi shem XSD in WSDL povezuje do spletnih storitev, temelječih na protokolu SOAP, sam sistem OMS pa je na strežniku v celoti implementiran v programskem jeziku C# in na ogrodju .NET.

Na tem mestu velja omeniti še konkretno implementacijo ogrodij Frameworkx, ki jo uporablja naše telekomunikacijsko podjetje. Gre za implementacijo podjetja IBM, imenovano *WebSphere Telecom Operations Content Pack* (v nadaljevanju **TOCP**), ki je sestavljeno iz več prepletajočih se modulov, kjer je ključni tisti, ki implementira shemo *SID Telecom Business Object Model*. Ta je po proizvajalčevih podatkih razvit na podlagi standardov SID in razširjen z najboljšimi praksami podjetja IBM.

3.5. Načrtovalni pristopi SID

V naslednjih podpoglavjih si bomo ogledali osnovne načrtovalne pristope, ki so uporabljeni pri realizaciji sheme SID in so povzeti po knjigah TM Foruma [24] in [25].

3.5.1. Specifikacija entitete/entiteta

Pristop *specifikacija entitete/entiteta* (angl. *Entity specification/Entity*) je v uporabi skozi celotno SID ogrodje, predvsem pri podpori eTOM vertikalne procesov prvega nivoja, pri upravljanju življenjskega cikla proizvoda (PLM). Ta se osredotoča na tipe produktov, storitev in virov, ki so del ponudbe in infrastrukture ponudnika storitev.

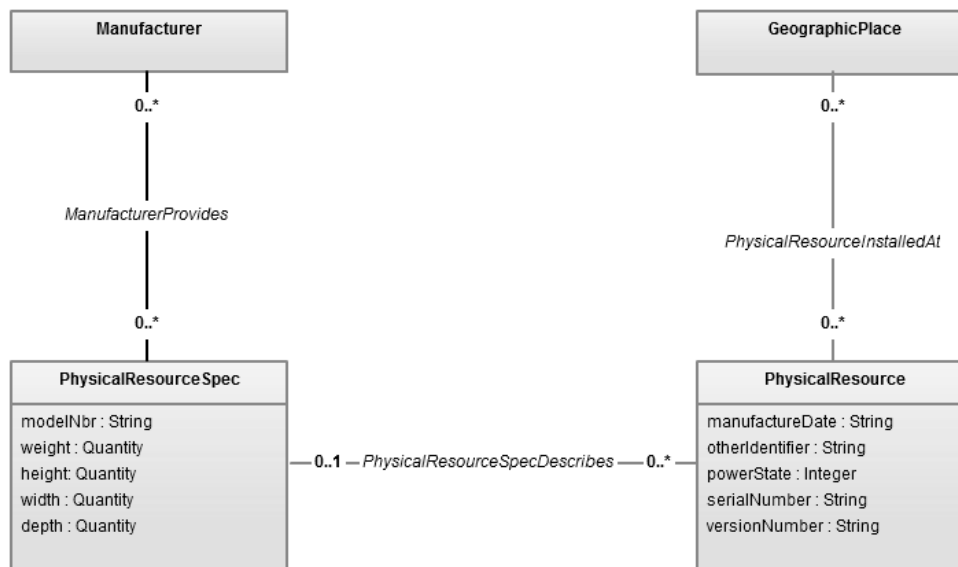
Gre za zelo osnoven pristop in bi ga lahko opisali kot seznam specifikacij ali kot podatkovnik proizvoda. Entiteta ima lahko nič ali več takih specifikacij.



Slika 3-5: Modelirni vzorec »specifikacija entitete/entiteta«

Kot netrivialen, a jasen primer iz modela SID bi lahko navedli opremo, ki je predstavljena kot fizični vir (*PhysicalResource*). Primer tega fizičnega vira ima lahko specifikacijo ali skupek lastnosti. Te so npr. številka modela, velikost, teža in celo relacija s proizvajalcem (*Manufacturer*) ter so predstavljene kot specifikacija fizičnega vira (*PhysicalResourceSpec*). Ta skupek lastnosti je nespremenljiv za vsako instanco opreme, ki se sklicuje na eno specifikacijo.

Vrednosti za dejstva, ki so spremenljiva za vsako instanco opreme, npr. serijska številka, lokacija ipd., pa so v SID neposredno lastnost entitete. Torej v podanem primeru fizični vir sovpada z entiteto, specifikacija fizičnega vira pa s specifikacijo entitete.



Slika 3-6: Primer uporabe modelirnega vzorca specifikacija entitete/entiteta v domeni vira na poslovni entiteti *PhysicalResource* iz sheme SID

3.5.2. Entiteta/vloga entitete

Skozi svoj življenjski cikel lahko igrajo entitete več vlog glede na to, s katerimi entitetami se povezujejo. Posameznik je tako lahko stranka in tudi zaposlenec pri ponudniku storitev, kar je primer entitete z več vlogami.

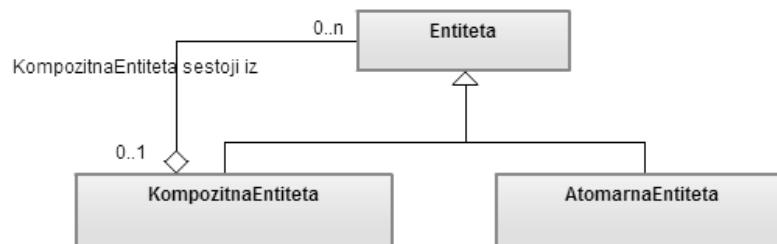
Kot drugi primer igranja vloge glede na povezavo z drugimi entitetami pa lahko navedemo primer produkta, ki je lahko (med drugim) povezan s stranko ali pa uporabniškimi računom stranke, kjer v prvem primeru stranka *naroči* storitev, storitev pa *se zaračuna* uporabniškemu računu.

Entiteta/vloga entitete (angl. *Entity/Entity role*) je pristop, ki se uporablja največ v splošni, produktni, storitveni in domeni virov. Ta pristop omogoča entiteti vključevanje dodatnih vlog brez potrebe večjih strukturnih sprememb modela SID.

3.5.3. Kompozitno/atomarno

Je najpogosteje uporabljen načrtovalni vzorec v SID in omogoča modeliranje paketov in skupin entitetnih tipov, definiranih v hierarhičnih ali matrično orientiranih strukturah.

Hierarhične strukture so omogočene, kadar je povezava med kompozitno poslovno entiteto in poslovno entiteto ena – mnogo, matrične pa takrat, kadar sta v odnosu mnogo – mnogo.



Slika 3-7: Modelirni pristop kompozitno/atomarno

Na sliki lahko vidimo, kako *AtomarnaEntiteta* predstavlja entiteto, ki je ni možno razbiti na manjše podenote, *KompozitnaEntiteta* pa je kompozicija večjega števila entitet tipa *Entiteta*.

Predvsem hierarhične strukture so izredno značilne za telekomunikacijsko domeno, saj so storitve ponudnika organizirane kot hierarhija produktov in storitev, kar bomo videli v nadaljnjih poglavjih o produktnem katalogu in podatkovnem modelu OMS.

3.5.4. Entiteta specifikacija karakteristik/entiteta karakteristik

Dandanes je ponudba telekomunikacijskega podjetja sestavljena iz toliko različnih storitev in proizvodov, da je praktično nemogoče vnaprej predvideti vse dinamične in statične attribute vseh storitev ali produktov, ki okarakterizirajo neko poslovno entiteto.

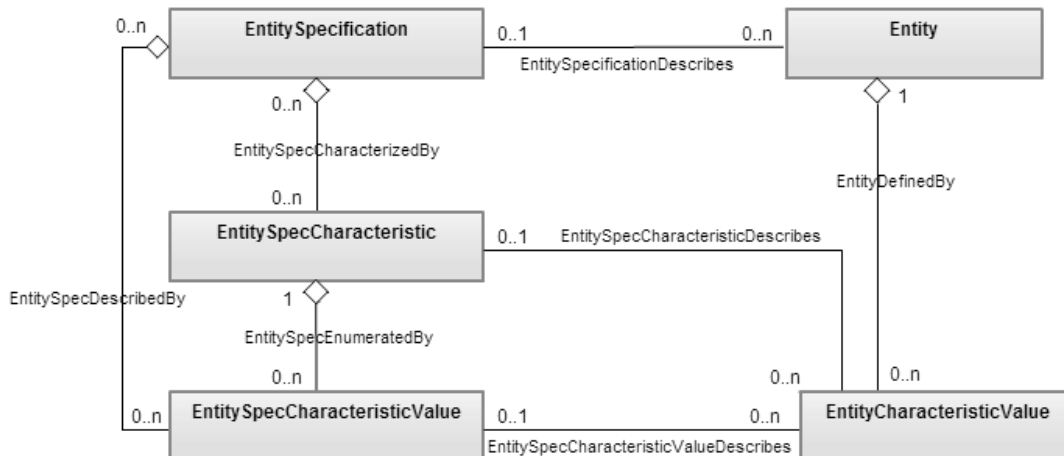
Oprema za različne storitve ima denimo različne karakteristike. Če npr. primerjamo modem za širokopasovne storitve in televizijski vmesnik (angl. *set-top box*), lahko ugotovimo, da imata nekaj skupnih lastnosti (npr. serijska številka) ter nekaj povsem nekompatibilnih (npr. število priključkov za storitev VoIP). Še več je takih razlik, če ponudnik internetnih storitev začne ponujati GSM storitve in želi v obstoječem podatkovnem modelu hraniti podrobnosti storitve brez večjih posegov vanj. To je pravzaprav realen primer ponudnika storitev iz te diplomske naloge in kopice njegovih konkurentov.

Pristop Entiteta specifikacija karakteristik/entiteta karakteristik (angl. *Entity Specification Characteristic/Entity Characteristic*) se loteva omenjene podpore generičnosti in omogoča, da:

- A. nam ob morebitnem dodajanju nove karakteristike ni potrebno spreminjati sheme,

B. v shemi ne potrebujemo vnaprej definiranih polj, ki bi v nekaterih primerih bila prazne (ko za entiteto taka lastnost ni relevantna).

Poglejmo še konkretnější primer na podlagi slike 3-8. Entiteti *EntitySpecCharacteristic* in *EntitySpecCharacteristicValues* obstajata z namenom, da definirata celotno množico, ki je možna za entiteto *EntitySpecification*. Če je nabor karakteristike »BARVA« (*EntitySpecCharacteristic*) lahko [»rdeča«, »rumena«, »modra«, »črna«] (*EntitySpecCharacteristicValues*), je s spodnjo shemo možno podpreti to, da je za nek vnos (npr. mobilni telefon) v *EntitySpecification* ta nabor omejen zgolj na [»rumena«, »črna«]. To omogoča mnogo – mnogo povezava, ki jo ima *EntitySpecification* z omenjenima tabelama.

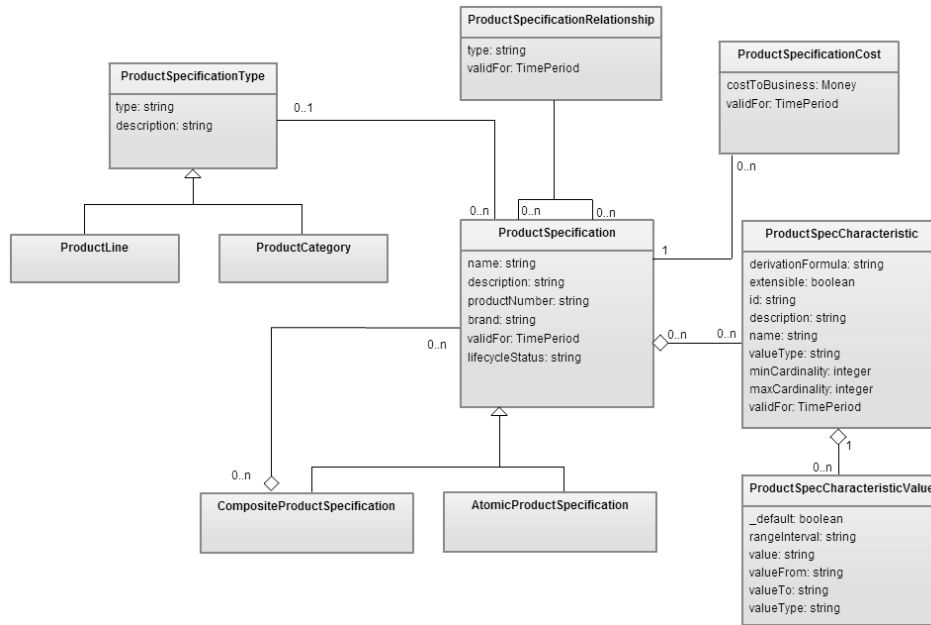


Slika 3-8: Entiteta specifikacija karakteristik/entiteta karakteristik

Kot bomo videli v nadaljevanju, pride pristop s karakteristikami izredno v poštev v storitveno usmerjeni arhitekturi, saj sprememba vmesnikov ob vpeljavi novosti zaradi točke A ni potrebna. V primeru, da bi uporabili pristop *Specifikacija entitete/entiteta*, bi to bilo neizogibno.

3.5.5. Primer na specifikaciji produkta

Za zaključek pregleda modelirnih pristopov bomo okvirno predstavili shemo SID na primeru agregirane poslovne entitete specifikacija produkta (*ProductSpecification*). Slika 3-9 predstavlja entitete iz drugega nivoja razgradnje sheme SID. Nekatere entitete so namenoma izpuščene z namenom, da je shema jasnejša za analizo pristopov načrtovanja iz prejšnjih poglavij, prav tako pa ni nobene povezave na entitete drugih ABE (npr. produkta, ki je pravzaprav instanca specifikacije produkta).



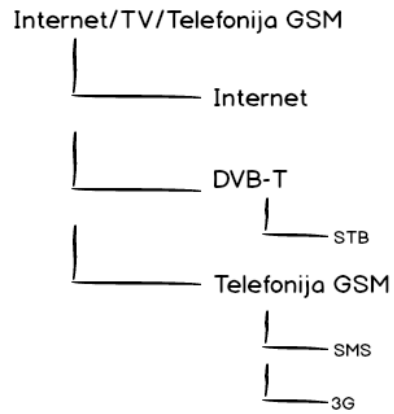
Slika 3-9: Prikaz pristopov načrtovanja na ABE specifikacija produkta (ProductSpecification)

Slika precej jasno prikazuje znane vzorce, predstavljene v predhodnih poglavjih. Če smo bolj eksplicitni:

- Entiteta *ProductSpecificationType* določa, katerega tipa je *ProductSpecification*, torej gre za pristop **entiteta/entiteta vloga**.
- *ProductSpecification* je lahko kompleksna (kompozicija) ali enostavna entiteta, saj je s pomočjo entitet *CompositeProductSpecification* in *AtomicProductSpecification* uporabljen pristop **kompozitno/atomarno**.
- *ProductSpecification* ima lahko tudi več karakteristik, te pa več vrednosti, kar je v principu pristop **entiteta specifikacija karakteristik/entiteta karakteristik**.
- Edini pristop od predstavljenih, ki nam manjka, je **specifikacija entitete/entiteta**. Čeprav sta entiteti *ProductSpecification* in *ProductSpecificationCost* v enakem razmerju, kot ga predvideva specifikacija entitete/entiteta, ne gre povsem za enako idejo – kreiranje instanc entitet na podlagi njenih specifikacij.

Zgornji primer ni dober samo za prikaz načrtovalnih pristopov znotraj ene same ABE prvega nivoja, ampak je bil njegov namen predstaviti, kako je videti specifikacija produktov v produktnem katalogu. Ta je v principu hierarhična struktura specifikacij produktov s karakteristikami in njihovimi vrednostmi. Prednost pristopa s karakteristikami je ta, da jih lahko za vsako specifikacijo produkta vnašamo neodvisno od sprememb sheme, saj je vsak atribut nov vnos oz. gledano z vidika podatkovne baze nova vrstica, namesto nov stolpec.

Podajmo še praktičen primer specifikacije produktov za zgornjo shemo, da razblinimo morebitne nejasnosti. Spodnja slika prikazuje hierarhijo specifikacije produkta *Internet/TV/Telefonija GSM*. Njegova kompozicija obsega specifikacije produktov *Internet*, *DVB-T* in *Telefonija GSM*. Prvi je enostavna specifikacija produkta, saj ni sestavljen iz drugih specifikacij produktov, medtem ko je del specifikacije produkta *DVB-T* tudi *STB* (angl. *set-top-box*), oprema, ki skrbi za dekodiranje signala. V primeru da ima telekomunikacijsko podjetje v ponudbi na voljo več različnih oprem *STB*, te ne bo dodalo v specifikacijo produkta *DVB-T* opreme, ampak jo bo ponudilo kot *dodatno storitev*. Koncept *ponudbe*, *storitev* in *dodatnih storitev* si bomo pogledali v poglavju o produktnem katalogu.



Slika 3-10: Hierarhična struktura, ki jo omogoča specifikacija produkta

Kot lahko vidimo, so produkti, ki tvorijo trojček *Internet/TV/Telefonija GSM*, precej različni, kar pomeni, da bodo specifikacije produktov imele tudi precej različne karakteristike, torej vnose v tabeli *ProductSpecCharacteristic*. Če primerjamo zgolj ključne identifikatorje, ki so pomembni za oskrbovanje teh storitev, so to *uporabniško ime* (internet), *serijska številka vmesnika* (TV) in *telefonska številka* (telefonija GSM).

4. Sistemi telekomunikacijskega operaterja in arhitektura OMS

4.1. Razvoj sistema za upravljanje z naročili in arhitekture B/OSS

Telekomunikacijsko podjetje, ki ga bom opisal, je večje podjetje iz tujine, ki je nastalo z združitvijo dveh podjetij in od takrat šlo skozi več faz informacijske prenove. Namen sprememb je bil vpeljati lastne sisteme ter zagotoviti učinkovitejšo in enostavnejšo informacijsko arhitekturo z deljenim informacijskim modelom, ki sledi priporočilom Frameworkx, obenem pa zagotoviti večjo avtomatizacijo poslovnih procesov. Večina prenov je v veliki meri zadevala sistem za upravljanje z naročili, ki ga je razvilo podjetje, pri katerem sem zaposlen. Nekatere spremembe so se bolj dotikale samega OMS, druge so bile potrebne zaradi ukinjanj, nadgrajevanj, uvajanj ali združevanj drugih sistemov, pri večini pa je bil skupni imenovalec optimizacija poslovnega procesa naročanja storitev.

OMS se namreč povezuje z velikim številom sistemov (kar 13 vključno z BPMS) z namenom avtomatizacije opravil skozi celoten poslovni proces: od naročila nove storitve ali spremembe, preko zagotavljanja predpogojev za realizacijo izvedbe naročila (npr. zahteva spremembe pri nacionalnem operaterju na najeti liniji ali zahtevanje prenosa telefonske številke od drugega operaterja), pa vse do vklopa storitev in vnosa podatkov o reverzih, subvencijah in dokumentih; seveda vse to ob upoštevanju poslovnih pravil podjetja.

Sistem OMS je bil nameščen v produkcijsko okolje z začetkom leta 2011 v okviru večje informacijske prenove, od takrat pa je bilo izvedenih še precej posegov v informacijsko arhitekturo. Predstavimo jih lahko takole:

0. Zamenjava sistema za upravljanje s storitvami GSM in uvedba sistema OMS s poudarkom na optimizaciji poslovnih procesov

Kot že omenjeno v uvodu, je v projektu korenite informacijske prenove združeno podjetje zamenjalo sistem za upravljanje storitev GSM in tako onemogočilo konkurenčnemu podjetju, ki je bilo lastnik sistema, vpogled v svoje podatke o naročniških razmerjih. S tem projektom pa je nastopila tudi potreba po učinkovitem sistemu za upravljanje z naročili, saj je bil prejšnji sistem za vnos naročil tesno povezan s sistemom za upravljanje storitev, hkrati pa je bil manj učinkovit z vidika podpore poslovnim procesom in upravljanja življenjskega cikla produktov.

1. Uvedba hitre avtorizacije naročil

Ta sprememba je pomenila spremembo poslovnega procesa naročanja storitev, saj je naročnik želel vnos naročila na prodajnem mestu (POS) izvesti kot dvokoračno opravilo, namesto v enem koraku, kot je to podpirala prvotna implementacija OMS. Namesto avtorizacije naročila po končanem vnosu celotnega naročila na prodajnem mestu, sedaj uporabnik POS vnese le podatke o stranki in zaključi opravilo. Takoj za tem prejme v svoj nabiralnik opravilo uporabnik v zaledju, ki ima pravico odločanja o odobritvi ali zavrnitvi naročila. Uporabnik opravi avtorizacijo na podlagi podatkov, ki so na voljo o stranki (obstoječe subvencije, zadnji računi, črna lista neplačnikov ipd.). V primeru, da je naročilo odobreno, lahko POS uporabnik nadaljuje z vnosom parametrov naročila. Avtorizacija naročila je potrebna, ker prodajno mrežo POS sestavlja veliko število pooblaščenih prodajalcev, ki ne spadajo pod okrilje telekomunikacijskega podjetja (npr. nacionalna pošta ipd.) in zato ne morejo odločati o morebitni zavrnitvi sklepanja naročniškega razmerja. Hkrati pa se tako podjetje seveda izogiba poslovnim škodam.

2. Sprememba v procesu in integracijah za namen **prenosljivosti številke fiksne in mobilne telefonije** zaradi sprememb na nivoju nacionalnega medoperaterskega portala. Ta je na podlagi zakonodaje podprl poslovna pravila, za katera mora operater, ki je prejel zahtevek za izključitev in prenos telefonske številke k drugemu operaterju, odgovoriti v določenem časovnem obdobju (4 ure v primeru, da je zahtevek prejet dopoldne, sicer pa najkasneje naslednji dan do 12h), v nasprotnem primeru se zahteva za prenosljivost smatra za odobreno. To je poleg spremenjene integracije v OMS pomenilo tudi spremembo implementacije poslovnega procesa v BPMS.

3. **Združitev sistemov za upravljanje storitev in uvedba sistema za pred-provizioniranje**

V času projekta prenove iz tč. 0 je bilo malo časa za migracijo vseh storitev v en sistem, zato se je podjetje odločilo, da migrira zgolj storitve telefonije GSM v povsem nov sistem, ostale fiksne storitve (fiksna telefonija, internet in televizija) pa obdrži v sistemu, ki ga je samo razvilo. To je bilo seveda precej nerodno, saj so storitve neke stranke (npr. paket trojček) lahko bile razpršene v obeh sistemih, kar je pomenilo tudi dvojno hranjenje podatkov o stranki in posledično oteževalo zaračunavanje storitev. Za namen podpore procesom sprememb je tako OMS moral spraševati oba sistema, ali poznata stranko glede na vnesen identifikator in katere so njene storitve, nato pa je te moral še pravilno uvrstiti v paket na podlagi strukture produktnega kataloga.

Podatki sistema fiksnih storitev so tako bili migrirani v sistem z mobilnimi storitvami, sistem fiksnih storitev pa ukinjen. Pri podjetju so tako imeli dovolj resursov, da so lahko znotraj projekta razvili tudi *sistem za pred-provizioniranje*, tj. avtomatizacijo preverjanja zmogljivosti in najemanja linij nacionalnega telekoma, saj operater nima svojih linij po celotni državi.

Dolgoročno gledano je bil to nujen poseg v sisteme ponudnika.

4. **Zamenjava sistema za oskrbovanje s predplačniškimi storitvami**

Ker je krovno podjetje, v čigar lasti je naš ponudnik telekomunikacijskih storitev, uporabljalo drugo platformo za oskrbovanje predplačniških storitev, je prehod na enako platformo, kot jo uporablja krovno podjetje, bil smiseln s finančnega vidika. V tem delu informacijske prenove je OMS prevzel prodajni del z vsemi podrobnostim paketov (ki so bili skonfigurirani v PLM), kar je prej opravljala interno razvita aplikacija. Ker gre pri ponudniku za veliko število prodanih predplačniških paketov vsak dan, OMS po vnosu naročila avtomatsko proži provizioniranje preko spletnih storitev vmesnikov TOCP.

5. **Zamenjava sistema BPMS za upravljanje poslovnih procesov**

Ker je naše telekomunikacijsko podjetje del večje skupine operaterjev, je iz podobnega razloga kot v prejšnji točki bila izvedena tudi migracija platforme za upravljanje poslovnih procesov. Prav v času pisanja tega diplomskega dela je bila izvedena migracija iz Oracle BPM na IBM BPM.

Med vsemi naštetimi večjimi informacijskimi prenovami je bilo implementiranih še ogromno dodatnih funkcionalnosti, sprememb in optimizacij v sistemu OMS (skupaj 62 namestitev novih verzij OMS do časa pisanja), ki so bodisi podprle bodisi avtomatizirale dele poslovnega procesa ali kako drugače olajšale delo zaposlenim. Vsi navedeni posegi so bili narejeni v roku dveh let, kar bi bilo nemogoče s slabo zasnovano sistema OMS.

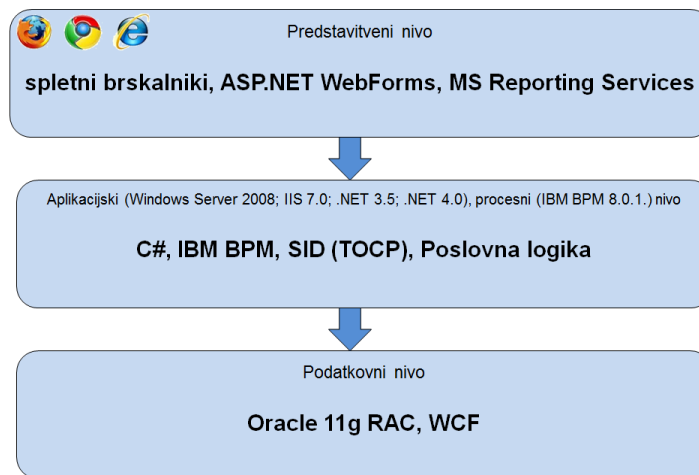
4.2. Sistem za upravljanje z naročili (OMS)

4.2.1. Arhitektura rešitve in tehnologije

Ker je sistem za upravljanje z naročili tak, da ga uporablja veliko prodajnih mest, med katerimi so tudi zunanji posredniki, in ker se prodajna mreža lahko vedno širi, je bila najbolj očitna izbira za razvoj predstavitvenega dela sistema spletna aplikacija. Rešitev OMS obsega naslednje module, ki so na predstavitvenem nivoju vsi spletne aplikacije:

- Modul za naročila
- Modul za zadržanje strank
- Modul za arhiviranje
- Modul za vnos pritožb

Moduli rešitve so na strežniškem delu implementirani na ogrodju .NET v programskem jeziku C#, torej so spletne forme implementirane na delu ogrodja .NET za spletne forme – ASP.NET, za asinhrono osveževanje uporabniškega vmesnika pa je uporabljen tudi del ogrodja ASP.NET AJAX.



Slika 4-1: Plasti rešitve OMS

Na aplikacijskem oz. nivoju poslovne logike je pomembno omeniti BPMS, ki skrbi za to, da je procesni del ločen od samega OMS. Na tem mestu je potrebno še omeniti intenzivno uporabo spletnih storitev, za kar je uporabljen API ogrodja .NET, imenovan *Windows Communication Foundation (WCF)*, s katerim so tudi generirani klienti za dostop do spletnih storitev TOCP, ki smo jih predstavili v poglavju 3.4.2.

Na nivoju podatkovne baze je za namen hranjenja podatkov uporabljen Oracle 11g RAC, ki je rešitev v gruči za namen doseganja visoke razpoložljivosti sistema. Za delo z relacijsko podatkovno bazo je uporabljena prilagojena implementacija *LINQ to SQL*¹¹ za Oracle, saj uradna izdaja podjetja Microsoft podpira zgolj delo z Microsoft SQL Server.

4.2.2. Podpora poslovnim procesom

V prvotni implementaciji OMS se je kot sistem za upravljanje s poslovnimi procesi uporabljal *Oracle BPM Suite 10g*. Tehnični razlog za izbiro Oracle BPMS so bila dobra analitična orodja in

¹¹ *Language-Integrated Query (LINQ)* je programsko ogrodje, ki omogoča implementacijo poizvedb nad objekti, v implementaciji *LINQ to SQL* pa generira poizvedbe SQL in opravlja mapiranje iz podatkov relacijske baze v objekte programskega jezika C#, torej opravlja nalogo ORM.

dobri licenčni pogoji. Poleg tega je naročnik že uporabljal nekatere njihove sisteme in imel za to usposobljene kadre. [15]

Oracle BPMS 10g, ki je bil uporabljen kot platforma za poslovne procese, nudi tudi možnost izdelave uporabniških vmesnikov, a se ti ne obnesejo najbolje pri zahtevnejših opravilih, zato se pri razvoju rešitve nismo odločili izkoristiti vseh funkcionalnosti, ki jih je ponujal Oracle BPMS. Kasneje se je to izkazalo za zelo dobro odločitev.

Podjetje se je namreč s koncem leta 2012 odločilo zamenjati Oracle BPMS za IBM BPM. Razlog za to je bila racionalizacija z vidika stroškov za informacijsko podporo, saj je naše telekomunikacijsko podjetje del večje, mednarodne skupine ponudnikov telekomunikacijskih storitev, in medtem ko je skupina rasla, je prevladala pripadnost sistemom za upravljanje poslovnih procesov na platformi IBM.

Moduli OMS trenutno nudijo podporo poslovnim procesom, ki so navedeni v spodnji tabeli.

Tabela 4-1: Poslovni procesi, ki jih podpira OMS

Poslovni proces	Modul OMS
naročilo nove storitve in reaktivacije	naročila
spremembe na storitvah, strankah in računih	naročila
proces izključitev storitev	naročila
pritožbe	pritožbe
arhiviranje podpisanih dokumentov, ki so na POS	arhiviranje
vračanje opreme	naročila
zadržanje strank	zadržanje

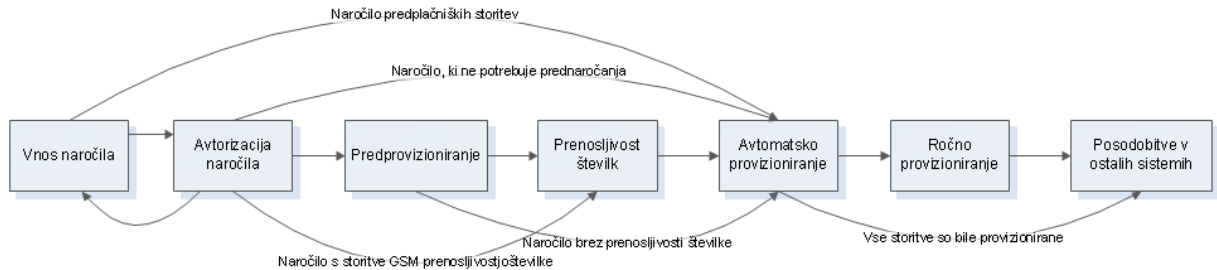
Kot vidimo, je poslovnih procesov, ki jih podpira OMS, veliko. Za namen te diplomske naloge pa se bomo osredotočili na naročilo nove storitve. V trenutni verziji OMS poslovni proces obsega vsega skupaj:

- 5 možnih ročnih korakov, katerih prejemniki so uporabniki na prodajnem mestu (angl. *POS user*),
- 5 možnih ročnih korakov, katerih prejemniki so uporabniki v zalednem poslovanju (angl. *Back office user*),
- 1 ročni korak avtorizacije za prejemnika iz skupine BCF¹² ali LCD¹³,
- 7 možnih avtomatskih korakov za izvedbo integracij.

Ker za namen tega diplomskega dela ni potrebna podrobna predstavitev vseh korakov, si poglejmo poenostavljen poslovni proces za vnos novega naročila.

¹² Oddelek za zaračunavanje in preprečevanje zlorab (angl. *Billing Collections and Fraud*)

¹³ Oddelek za opomine in izterjavo (angl. *Late Collections Department*)



Slika 4-2: Poenostavljen proces novega naročila

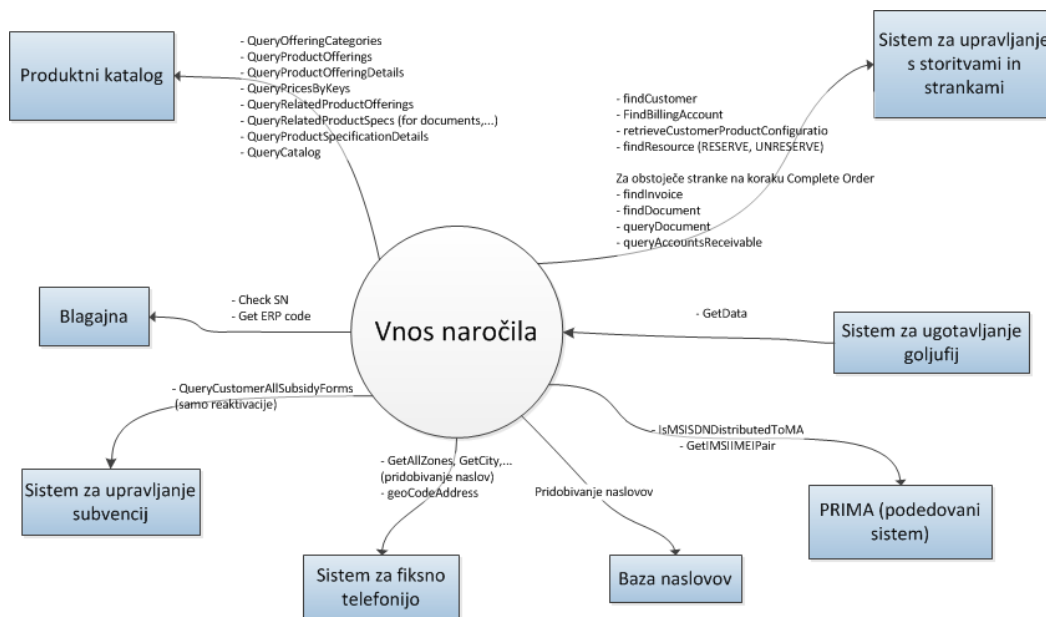
Na podlagi opisanih informacijskih prenov iz poglavja 4.2 je slika verjetno že precej razumljiva, zato z opisom posameznih delov ne bom preveč izčrpen.

1. **Vnos naročila** – dvofazni vnos podatkov, najprej vnos stranke, nato še vnos naročila.
2. **Avtorizacija naročila** – uporabnik iz skupine BCF ali LCD preveri stranko in bodisi odobri bodisi zavrne naročilo.
3. **Predprovizioniranje** – v primeru, da gre za tip naročila, za katerega je treba izpolniti nekatere predpogoje, se o tem obvesti zunanji sistem za pred-provizioniranje. Ta pod-proces je povsem avtomatiziran.
4. **Prenosljivost številke** – avtomatski pod-proces, kadar je potrebno prenesti številko od drugega operaterja.
5. **Avtomatsko provizioniranje** – avtomatsko omogočanje storitev, ki je podprto za predplačniške storitve, ter nekatere spremembe. Tudi posodobitve v drugih zunanjih sistemih, če je to potrebno.
6. **Ročno provizioniranje** – v tem delu uporabnik poskrbi za aktivacijo storitev v sistemu za upravljanje s storitvami.
7. **Posodobitev ostalih sistemov** – posodabljanje podatkov v sistemu za subvencije, pošiljanje dokumentov v sistem za hrambo dokumentov ipd., ko je že jasno, da so storitve bile realizirane.

4.3. Zaledni sistemi in način povezovanja z OMS

OMS se v času pisanja povezuje s trinajstimi sistemi oz. moduli, ki skrbijo vsak za svoje področje in tvorijo celoto sistemov OSS/BSS operaterja.

Spodnji diagram prikazuje sisteme, s katerimi se OMS povezuje v fazi vnosa naročila v procesu novega naročila. Kot je razvidno iz diagrama, so v fazi vnosa vse integracije, razen sistema za ugotavljanje goljufij, tipa na zahtevek¹⁴ s strani OMS. Tudi integracija s sistemom za ugotavljanje goljufij je pravzaprav tipa na zahtevek, a je v tem primeru sistem tisti, ki izda zahtevo. Ta namreč periodično prebira vnaprej pripravljene podatke iz sistema OMS.



Slika 4-3: Prikaz povezovanja OMS z drugimi sistemi v prvem in tretjem koraku poslovnega procesa vnosa novega naročila

Ker je sistemov preveč za podrobno obravnavo, si bomo pogledali le najpomembnejše z vidika integracij z OMS.

4.3.1. Produktni katalog

Produktni katalog je rešitev, ki na strukturiran in konsistenten način ponuja upravljanje informacij o produktih, ponudbah, njihovih cenah, parametrih in relacijah med njimi. Po priporočilih eTOM naj bi se v njem odražal rezultat poslovnega procesa *upravljanja življenjskega cikla proizvoda*¹⁵.

Rešitev, ki jo uporablja naše telekomunikacijsko podjetje, je razvita s strani zunanega izvajalca in je prilagojena telekomunikacijski domeni, torej njegovi vmesniki sledijo priporočilom SID. Nadaljnji opis se nanaša na to konkretno rešitev, kot jo preko SOAP vmesnikov vidi OMS.

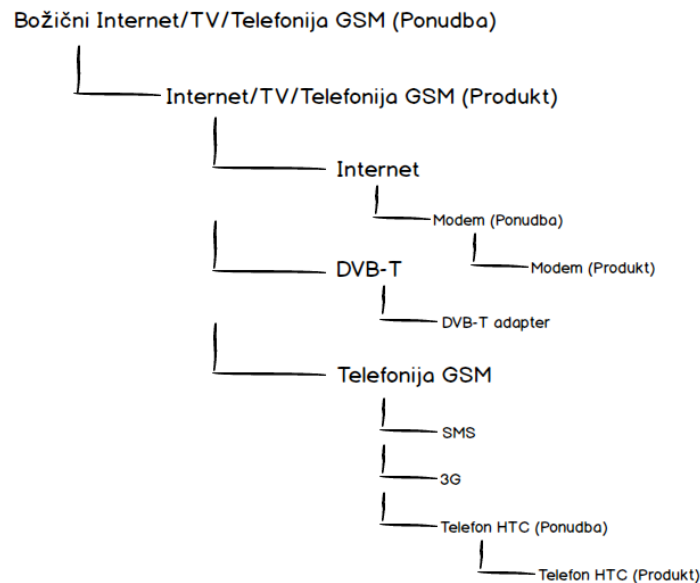
Produktni katalog s pridom izkorišča modelirni pristop **kompozicije** (3.5.3) tako, da omogoča sestavo produktov, ki lahko vsebujejo seznam produktov, vsak nadaljnji produkt pa lahko spet vsebuje sezname produktov itd. Tako dobimo možnost sestave hierarhije, tipične za ponudbo v telekomunikacijski domeni.

¹⁴ angl. pull request

¹⁵ *Product Lifecycle Management* proces najdemo v eTOM dekompozicija prvega nivoja

Naj nadaljujem, kjer smo ostali v poglavju 3.5.5 – na primeru specifikacije produkta *Internet/TV/Telefonija GSM*. Ko je ta skonfigurirana, lahko v produktnem katalogu iz njega generiramo **ponudbo** (*ProductOffering*).

Najprej pojasnimo, kaj pomeni ponudba, saj je prevod v slovenski jezik lahko zavajajoč. Ponudba v tem kontekstu namreč predstavlja **konkreten primerek in ne nabora celotne ponudbe**, torej gre zgolj za posamezno instanco iz palete ponudbe. Povedano poenostavljeno, ponudba ni nič drugega kot produkt s cenami in možnostjo redefiniranja posameznih parametrov produkta (karakteristik, dodatnih storitev ...).



Slika 4-4: Primer ponudbe (product offering) "Božični Internet/TV/Telefonija GSM"

To je izredno priročno. Zamislimo si, da želi ponudnik storitev imeti prodajno akcijo obstoječega »*XXL Internet/TV/Telefonija GSM*« paketa v božičnem času. Da uvede novo ponudbo »*Božični Internet/TV/Telefonija GSM*«, mu vzame le nekaj minut, in sicer, da ustvari novo ponudbo iz obstoječe kompozicije produktov *Internet/TV/Telefonija GSM* ter jim nastavi cene paketa. Pri tem npr. naročniško ceno in kazen ob predčasni prekinitvi katere izmed storitev vnese drugačno kot pri paketu »*XXL Internet/TV/Telefonija GSM*«, ostale cene (npr. ceno terminalne opreme za DVB-T, ki je v kompoziciji) pa prekopira iz obstoječega »*XXL Internet/TV/ Telefonija GSM*«. Novo ponudbo vključi še v eno izmed kategorij in ta je na voljo za prodajo v OMS.

Iz zgornjega je razvidno, da morajo v podjetju za vsako storitev ali opremo narediti en produkt, iz katerega nato tvorijo večje število kompozicij in praktično neomejeno količino ponudbe na tržišču v zelo kratkem času.

Na sliki je viden tudi koncept **dodatne ponudbe**. V principu lahko ločimo:

- Samostojno ponudbo (npr. Božični internet/TV/Telefonija GSM)
- Dodatno ponudbo (*Telefon HTC*)

Dodatna ponudba je v produktnem katalogu realizirana kot relacija na neko ponudbo (ali produkt). To pomeni, da je dodatna ponudba vedno ponudba za neko drugo ponudbo, v našem primeru na sliki 4-4 je to za ponudbo *Božični internet/TV/Telefonija GSM* in produkt *Telefonija GSM*.

Z relacijami se da podpreti veliko različnih konceptov, npr. dodatno ponudbo samo za nove storitve ali samo za spremembe, določitev tega, kateri prehodi med paketi so možni ipd.

Oglejmo si še, kako so videti deli XML strukture tako pridobljenega rezultata za samostojno ponudbo. Odgovor, ki ga pridobimo, je tipa *ProductOfferingDetails* in vsebuje osnovne podatke ponudbe: ime, identifikatorje (na podlagi katerih pridobimo cene, dodatne storitve), datume veljavnosti ponudbe, kategorije, kamor spada ta ponudba ipd. Nato pa pridobljeni podatki vsebujejo hierarhijo *ProductSpecificationDetails* objektov, ki opiše strukturo ponudbe iz prej omenjenega primera, seveda brez dodatne ponudbe, saj je ta pridobljena kot ločena poizvedba SOAP.

ProductOfferingDetailsType					
xm: xsi	http://www.w3.org/2001/XMLSchema-instance				
xm: xsd	http://www.w3.org/2001/XMLSchema				
OfferingId	2236628				
OfferingCode	493-43-10-30				
OfferingName	Božični internet/TV/Telefonija GSM				
TaxRateCode	ODS				
TaxRateName	STANDARD TAX RATE (18%)				
ProductSpecificationDetails					
Id	2251091				
Code	MM-43-10-30-P				
Name	Internet/TV/Telefonija GSM (produkt)				
Quantity	1				
ProductSpecificationDetailsList					
ProductSpecificationDetails (3)					
Id	Code	Name	Quantity	Characteristics	ProductSpecificationDetailsList
1	1000003	MM-200-10-CFS	Internet	Characteristics	
2	1999012	MM-2-43-CFS	Telefonija GSM	Characteristics	
3	110	MM-400-30-CFS	DVB-T	Characteristics	
ProductSpecificationDetailsList					
ProductSpecificationDetails					
Id	1999019				
Code	MM-3-EQ				
Name	Sagem DVB-T adapter				
Quantity	1				
Characteristics					
Characteristic (4)					
DisplayName	Final				
1 Available immediately	true				
2 Is equipment	true				
3 CPE ownership					
4 Serial number					
EffectiveStart...	2012-09-01T00:00:00+02:00				
EffectiveEndD...	3000-01-01T00:00:00+01:00				
Version	1				

Slika 4-5: Rezultat produktnega kataloga, pridobljena za ponudbo »Božični Internet/TV/Telefonija GSM«

Poglejmo še, kako se OMS povezuje s sistemom produktnega kataloga preko spletnih storitev SOAP. V primeru vnosa naročila za paketno ponudbo treh storitev mora OMS napraviti naslednje poizvedbe.

Pred izbiro paketne ponudbe:

- narediti poizvedbo za vse kategorije storitev,
- pridobiti seznam storitev za izbrano kategorijo storitev.

Ob sami izbiri paketa:

- pridobiti podrobne informacije o izbranem paketu,
- za vsak kompleksen parameter¹⁶ pridobiti vrednost iz šifranta podatkov za prikaz,
- pridobiti cene za ta paket. Cene storitev, ki so v kompoziciji celotnega paketa, pridobi z enim klicem, saj je cena lastnost celotne ponudbe (angl. *offering*),
- pridobiti seznam relacij¹⁷ za izbran paket.

¹⁶ Kompleksen parameter storitve je v opisanem produktnem katalogu tipa »ENTITY_ID«, kar pomeni, da vmesnik vrne le identifikator in zato mora za pridobitev opisne vrednosti (ki ima nek pomen za uporabnika) povezujoč se sistem spet narediti novo poizvedbo v produktni katalog za poizvedbo po šifrantu.

Ob kliku na gumb za izbiro dodatnih storitev (v primeru, da obstaja možnost dodajanja dodatnih storitev):

- pridobiti detaljne informacije o dodatnih storitvah,
- pridobiti cene za vse dodatne storitve.

Ob izbiri dodatne storitve:

- za vsak kompleksen parameter pridobiti vrednost iz šifranta podatkov za prikaz.

Parametri ne obsegajo zgolj tistih, ki jih uporabnik vidi na vnosnih formah, ampak tudi veliko metapodatkov, npr. tehnologijo, na kateri je realizirana storitev, kodo storitve v sistemu za upravljanje storitev, zato je teh parametrov lahko veliko za vsako izmed storitev (za storitve na nivoju CFS okrog 30). Večina jih je sicer enostavnega tipa, a vendar je na koncu ob izbiri paketa z več storitvami potrebno narediti nekaj deset klicev do produktnega kataloga.

Taka rešitev zaradi počasnega delovanja ne bi bila sprejemljiva, zato je na strani OMS implementirano avtomatsko opravilo, ki vsak dan v nočnem času izvede vse kombinacije klicev do produktnega kataloga in jih shrani v pomnilnik kot tudi na datotečni sistem strežnika z enotnim ključem *ImeMetode_VhodniParametriMetode*.

Ob prvem dostopu do nekega ključa nato OMS najprej preveri, ali zapis že obstaja v pomnilniku. V primeru, da ne obstaja, preveri, ali zapis že obstaja na datotečnem sistemu in če ne, potem pokliče spletno storitev produktnega kataloga.

Ker so spremembe v produktnem katalogu pri taki implementaciji za OMS vidne šele naslednji dan, ponuja OMS tudi možnost brisanja pred-pomnjenih vrednosti.

4.3.2. Sistem za upravljanje s storitvami in strankami

Sistem je bil uveden hkrati s sistemom za upravljanje storitev in je sprva skrbel le za storitve, temelječe na tehnologiji GSM. Leto po prehodu v produkcijsko okolje je prevzel še naloge sistema za upravljanje s fiksnimi storitvami.

Je centralni sistem operaterja, razvit znotraj skupine telekomunikacijskih operaterjev. Deluje v domenah »Stranka« in »Storitev«, opravlja pa tudi nekatere dele domene »Vir«, konkretnije upravljanje in rezervacije logičnih virov (telefonskih števil in uporabniških imen, upravljanje fizičnih virov (SIM), ter izpostavlja vmesnike za preverjanje pravilnosti fizičnih virov, ki so sicer shranjeni v modulu za upravljanje inventarja.

Sistem dosledno sledi priporočilom Framework in, kot že omenjeno, izpostavlja vmesnike, imenovane TOCP, ki implementirajo shemo SID in predstavljajo poseben nivo v arhitekturi sistema. Ta shema je uporabljena znotraj ponudnikove SOA infrastrukture in na voljo tudi zunanjim sistemom, kot je naš sistem za upravljanje s storitvami.

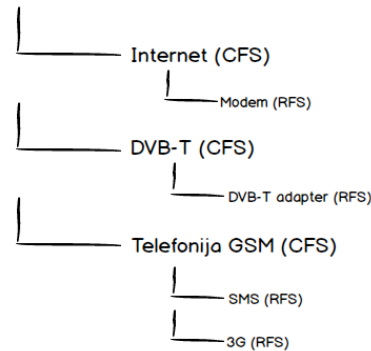
Ker gre za centralni sistem, ki podpira ključne funkcionalnosti, specifične za telekomunikacijsko domeno, je umestitev sistema v neko klasično kategorijo informacijske podpore podjetja precej težka naloga. Vsekakor gre za sistem CRM s temeljitim poudarkom na upravljanju storitev. Sistem namreč izpostavlja velik del funkcionalnosti, ki so prej osredotočene na storitve kot na stranko, kot npr. proženje provizioiniranja v za to namenjenih sistemih ali upravljanje z nekaterimi viri (npr. množični uvoz kartic SIM).

Poglejmo si še vmesnike sistema za delo s storitvami. Kot smo videli pri predstavitvi ABE za domeno storitev, sta ključna koncepta CustomerFacingService (CFS) in ResourceFacingService

¹⁷ Relacije opišejo druge ponudbe v povezavi na obstoječo in v primeru novega naročila iščemo seznam dodatnih storitev ob novem naročilu.

(RFS) in na spodnji sliki vidimo, kakšna je struktura podatkov, ki jo vrnejo vmesniki, ko OMS proizvede po storitvah stranke. Kot lahko opazimo, ima ta v primerjavi s produktim katalogom manj nivojev, v razliko pa se bomo temeljiteje poglobili v poglavju 5.2, v katerem bomo razložili, kako obe strukturi upari OMS. Na tem mestu povejmo zgolj to, da je sam odgovor TOCP neuporaben za prikaz na uporabniškem vmesniku, saj ne vsebuje nobenih uporabniku prijaznih sporočil. Gre za suhoparen odgovor z identifikatorji in parametri, ki jih je potrebno upariti s produktim katalogom, da dobijo smisel za uporabnika.

Božični Internet/TV/Telefonija GSM (ProductOffering)



Slika 4-6: TOCP struktura paketa »Božični internet/TV/Telefonija GSM«

Pokažimo še, da koncept CFS in RFS obstaja skozi vse vmesnike za delo s storitvami, kar je bistvena prednost sheme SID. Naslednja slika prikazuje zahtevek vmesnikom TOCP za zamenjavo opreme, ki ga bo ta poslal naprej sistemu za provizioniranje. V spodnjem primeru:

- prvo vozlišče *AgreementItem* pove, da se na storitvi CFS s podanim identifikatorjem **končuje veljavnost vira** (RFS) s podanim identifikatorjem, saj ima ta nastavljen atribut *validFor.endDateTime*,
- drugo vozlišče *AgreementItem* **doda** na CFS iz prejšnje točke nov RFS (zaradi odsotnosti polja *Id* za RFS) s karakteristiko (*CharacteristicValue*), ki označuje serijsko številko opreme.

Na tem mestu velja omeniti še atribut *validFor*. Ta je tipa *TimePeriod*, ki ima atributa *endDateTime* ter *startDateTime*. Gre za enega osnovnih objektov v SID, ki označuje veljavnost življenja entitet. Opazili smo ga lahko že v prejšnjih poglavjih, recimo na entiteti *ProductSpecification*.

customerOrder																			
▲ AgreementItemOrderItem (2)																			
⚙ Action	⚙ AgreementItem																		
1 rfs.update.eq	▲ AgreementItem <table border="1"> <thead> <tr> <th colspan="2">Product</th> </tr> </thead> <tbody> <tr> <td colspan="2"> ▲ CustomerFacingService </td> </tr> <tr> <td>⚙ Id</td> <td>2402384</td> </tr> <tr> <td colspan="2"> ▲ ResourceFacingService </td> </tr> <tr> <td>⚙ Id</td> <td>2697112</td> </tr> <tr> <td colspan="2"> ▲ validFor </td> </tr> <tr> <td>⚙ endDateTime</td> <td>2012-12-02T12:44:36.000+01:00</td> </tr> </tbody> </table>	Product		▲ CustomerFacingService		⚙ Id	2402384	▲ ResourceFacingService		⚙ Id	2697112	▲ validFor		⚙ endDateTime	2012-12-02T12:44:36.000+01:00				
Product																			
▲ CustomerFacingService																			
⚙ Id	2402384																		
▲ ResourceFacingService																			
⚙ Id	2697112																		
▲ validFor																			
⚙ endDateTime	2012-12-02T12:44:36.000+01:00																		
2 rfs.update.eq	▲ AgreementItem <table border="1"> <thead> <tr> <th colspan="2">Product</th> </tr> </thead> <tbody> <tr> <td colspan="2"> ▲ CustomerFacingService </td> </tr> <tr> <td>⚙ Id</td> <td>2402384</td> </tr> <tr> <td colspan="2"> ▲ ResourceFacingService </td> </tr> <tr> <td colspan="2"> ▼ Specification </td> </tr> <tr> <td colspan="2"> ▲ CharacteristicValue </td> </tr> <tr> <td colspan="2"> ▲ Characteristic </td> </tr> <tr> <td>⚙ Name</td> <td>CPE SERIAL NO</td> </tr> <tr> <td>⚙ Value</td> <td>5001288821</td> </tr> </tbody> </table>	Product		▲ CustomerFacingService		⚙ Id	2402384	▲ ResourceFacingService		▼ Specification		▲ CharacteristicValue		▲ Characteristic		⚙ Name	CPE SERIAL NO	⚙ Value	5001288821
Product																			
▲ CustomerFacingService																			
⚙ Id	2402384																		
▲ ResourceFacingService																			
▼ Specification																			
▲ CharacteristicValue																			
▲ Characteristic																			
⚙ Name	CPE SERIAL NO																		
⚙ Value	5001288821																		

Slika 4-7: Primer objekta, temelječega na strukturi SID, za naročilo zamenjavo opreme, kot ga OMS pošlje sistemu za upravljenje storitev

4.3.3. Sistem za zagotavljanje pred-provizioniranja

Sistem je bil na kratko predstavljen že v točki 3 poglavja 4.1., povzemimo pa še, katera področja ureja.

Ker naš ponudnik telekomunikacijskih storitev ne pokriva celotne države z lastnimi linijami za fiksne storitve, mora te najemati od nacionalnega operaterja. Njihov najem ni vedno možen, za prejem odgovorov in realizacijo najema pa lahko preteče tudi nekaj tednov. Tako sistem poskrbi za avtomatizacijo procesa in s tem prejemanje statusov tehničnih zmogljivosti, ki jih nato posreduje OMS. Ker lahko pride do napak pri komunikaciji med nacionalnim operaterjem in našim operaterjem, sistem ponuja tudi uporabniški vmesnik za urejanje statusa storitve. Omenimo še, da ta sistem pomeni velik korak k avtomatizaciji poslovnih procesov, saj so podobna opravila pred njegovo uvedbo opravljali ročno.

Sistem opravlja še eno nalogo. V odročnih krajih, kjer je uvedba fiksne linije velik strošek, podjetje namreč ponuja brezžični internet. Za postopek pred-provizioniranja to pomeni, da mora ponudnik storitev zagotoviti signal in spet obvestiti OMS, ko so storitve mogoče, da ta nadaljuje z izvedbo poslovnega procesa.

4.3.4. Sistem za upravljanje subvencij

Sistem za upravljanje subvencij hrani podatke o subvencijah za naročniška razmerja, kupljenih telefonih in najetih opremah. Ključ za prikazovanje, vnos, posodobljanje, prenos in brisanje subvencij je identifikator storitve iz sistema za upravljanje s storitvami.

Vmesnik sistema za povezovanje z ostalimi sistemi ne sledi shemi SID. Razlog za to je v dejstvu, da je sistem že precej star, saj prihaja še iz časov pred združitvijo obeh ponudnikov storitev. To pomeni, da mora vsak razvijalec OMS poznati in razumeti še eno dodatno shemo.

Za povrh je poslovna logika za delo s podatki za subvencije precej kompleksna, zato je pri integracijah s tem sistemom največ težav pri implementaciji novih funkcionalnosti in sprememb ali odpravljanju napak.

4.4. Primer novega naročila v OMS z opisom odvisnosti

Primer vnosa novega naročila z vidika integracij bomo opravili na že znani ponudbi *Božični Internet/TV/Telefonija GSM*.

1. **Vnos stranke** in njenih računov
 - V primeru vnosa naročila za obstoječo stranko OMS poišče stranko in njene uporabniške račune po vnesenem kriteriju preko vmesnikov TOCP.
 - V primeru, da obstoječe stranke ne najde, dovoli vnos nove, a naredi validacijo za obstoj stranke po njeni matični številki.
 - Če je stranka nova, preko spletne storitve preveri veljavnost vnesenega naslova v centralni bazi naslovov.
2. V koraku **avtorizacije** OMS poleg zgodovine naročil in detajlnih podatkov iz sistema za upravljanje storitev omogoča vpogled v podatke iz finančnih modulov (zadnji računi, odprte terjatve ...) in v primeru, da je naročilo avtorizirano in je stranka nova, vpiše stranko v CRM modul preko vmesnikov TOCP.
3. Pri **vnosu naročila** OMS
 - v produktnem katalogu pridobi podatke o paketu, kot je to opisano v 4.3.1;
 - v modulu za upravljanje z viri:
 - preveri, ali je vnesena telefonska številka prosta oz. na zahtevo uporabnika napolni spustni meni s telefonskimi številkami. Te številke OMS rezervira,

- preveri, ali je vneseno uporabniško ime prosto,
 - pridobi identifikator *SmartCardId* na podlagi serijske številke terminalne opreme za DVB-T;
 - v sistemu blagajne preveri mobilni telefon z vneseno IMEI številko;
 - ob izdaji opreme za internet in televizijo obvesti sistem za inventar. V primeru, da oprema s tako serijsko številko ne obstaja, OMS ne dovoli njene izdaje (in nadaljevanja naročila skozi proces naročanja).
4. Ob **koncu vnosa** naročila OMS odloži podatke na datotečni sistem v sistemu za ugotavljanje goljufij.
5. V pretežno avtomatiziranem podprocesu za **pred-provizioinranje** OMS:
- obvesti sistem za upravljanje najetih linij o naročilu. V primeru, da je naročilo realizirano na lokaciji in tehnologiji, ki ni v lasti našega operaterja, se bo ta pozanimal o možnostih za najem pri nacionalnem operaterju;
 - v primeru, da je zahtevana prenosljivost številke za telefonijo, se zahtevek pošlje v sistem za prenosljivost številok.

Na podlagi obeh odgovorov se proces bodisi zaključi bodisi gre v popraviljanje, ali pa se uspešno nadaljuje. V naslednjih korakih bomo predpostavili, da do zapletov ni prišlo in bomo po najenostavnejšem scenariju nadaljevali izvedbo naročila.

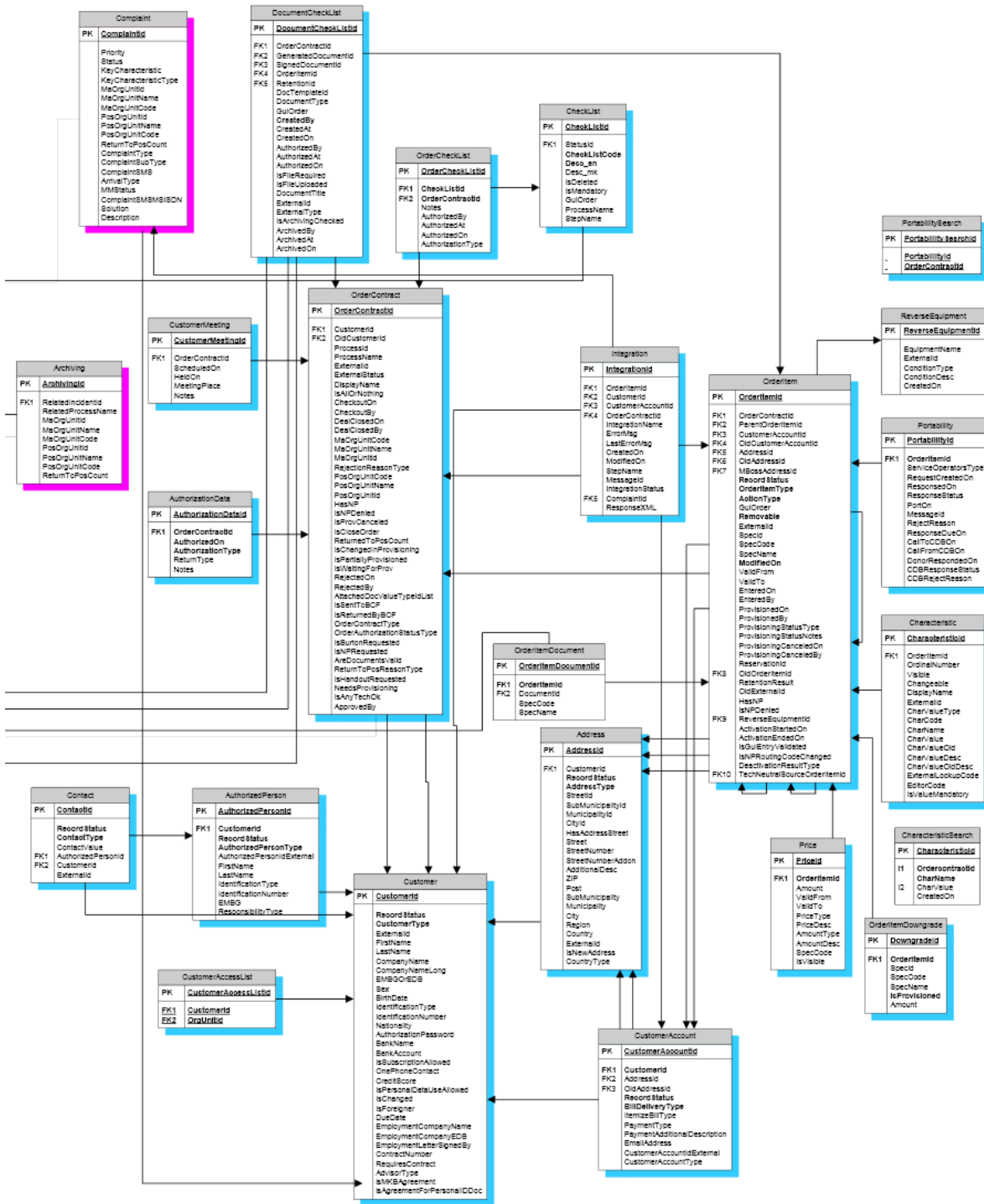
6. Sledi avtomatski korak, ki pripravi **pogoje za provizioniranje**; v našem scenariju so možne le spodnje integracije:
- če je bil vnesen naslov, ki še ne obstaja v bazi naslovov, se ga vpiše preko spletne storitve,
 - če je stranka obstoječa v CRM in so bili kateri podatki posodobljeni, se opravi posodobitev stranke;
 - v primeru, da je bil strankin račun za zaračunavanje nov, ga vpiše v CRM.
7. V **ročnem koraku provizioniranja** se preveri, ali je uporabnik res vnesel vse attribute v sistem za upravljanje s storitvami (ta aktivira storitve na storitvenem nivoju). Aktivacija novih storitev je v času pisanja avtomatska le za predplačniške sisteme, prav tako pa je avtomatiziran proces nekaterih sprememb (npr. telefonske številke, kartice SIM, opreme ...).
8. Ob **zaključku naročila** se zgodi naslednje:
- v sistem za upravljanje s subvencijami se zapišejo podatki o subvencionirani opremi (v scenarijih sprememb se prenesejo subvencije na podlagi poslovnih pravil) in zneski o kaznovanjih ob predčasni prekinitvi katere izmed storitev,
 - odrezvirajo se vse pridobljene telefonske številke iz naročila, ki niso bile uporabljene. V primeru preklica naročila bi bilo potrebno odrezvirati tudi uporabniško ime za storitev internet,
 - v sistem za hranjenje dokumentov se pošljejo vsi dokumenti, ki jih je podpisal uporabnik.

5. Podatkovna plast OMS

Naročnik, za katerega je bil razvit sistem OMS, je imel potrebo po popolnoma delujočem sistemu v izredno kratkem času. Poleg tega je bilo znano, da bo informacijska arhitektura telekomunikacijskega podjetja šla še skozi nekaj faz prenove in nadgradenj. Pri načrtovanju modela OMS je bilo tako smiselno razmišljati o dveh vidikih, in sicer:

1. učinkoviti realizaciji podatkovnega modela, tako da bo čim lažje zagotoviti funkcionalnosti v prvotni in tudi v vseh nadaljnjih različicah sistema, in sicer na tak način, da bo čim enostavneje podprti integracije in poslovna pravila;
2. ker so vmesniki najtesneje povezanih sistemov (upravljanje storitev, produktni katalog) sledili priporočilom ogrodja SID, je bilo smiselno čim bolj upoštevati ta priporočila z namenom zagotoviti čim enostavnejšo izmenjavo podatkov ter doseči konsistentnosti pri terminologiji. To poceni implementacijo sprememb, saj se s tem olajša delo analitikov in razvijalcev.

Rezultat zgornjih dveh vodil je podatkovna shema (slika 5-1) za podporo osnovnim poslovnim procesom naročanja (novo naročilo, spremembe, izključitve in vračanje opreme), na shemi pa sta vidni tudi dve krovni entiteti podpornih procesov pritožb (*Complaint*) in arhiviranja (*Archiving*), ki sta lahko povezani z naročilom.



Slika 5-1: Del podatkovne sheme sistema za OMS s tabelami za podporo osnovnih poslovnih procesov naročanja in dve krovni entiteti podpornih procesov (Archiving, Complaint)

Krovna entiteta za podporo omenjenim procesom je *OrderContract*, najbolj zanimiva pa je njena pod-tabela (angl. *child table*) *OrderItem*, ki ima rekurzivno relacijo in tako omogoča generiranje drevesne strukture naročila.

5.1. OrderContract

Korenska entiteta za podatke o naročilu, ki shranjuje podatke za ključne poslovne procese sistema za upravljanje z naročili in proces vračanja opreme (modul naročila). Vsebuje podatke o tem, kdaj so se zgodili kateri ključni dogodki pri procesu naročanja in kdo jih je naredil (uporabnik in prodajno mesto). Za namen enostavnejšega poročanja vsebuje nekatere podatke iz drugih tabel, torej model ni povsem normaliziran, vendar je takih izjem malo.

5.2. OrderItem

Je tabela, ki vsebuje hierarhijo naročila, ki predstavlja strukturo ponudb, produktov, storitev, oprem in dodatnih storitev.

OMS v svojem podatkovnem modelu ne loči entitet, kot so Ponudba, Produkt, Storitev ali Oprema, saj je mogoče hierarhijo teh entitet zaradi podobnih atributov predstaviti bolj generično, kar omogoča enostavnejšo razširljivost. Podoben pristop smo videli že pri produktnem katalogu. Glede na to, da OMS pobira naročila za ponudbo iz produktnega kataloga, se seveda zdi tak pristop smiseln. Hierarhično strukturo omogoča rekurzivna relacija ena – mnogo, ki se na fizičnem nivoju podatkovnega modeliranja odraža kot tuj ključ na polje *ParentOrderItemId*. Vnos v tabeli ima lahko natanko enega starša, kar omogoča drevesno strukturo.

Nekateri pomembnejši stolpci tabele *OrderItem*, ki so uporabljeni na vsakem nivoju in zato opravičujejo generično tabelo, so navedeni v Tabela 5-1.

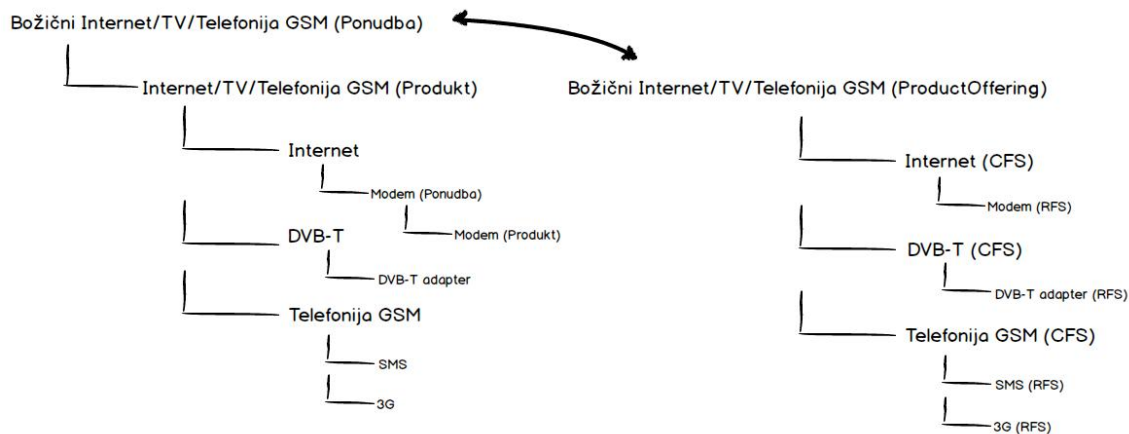
Tabela 5-1: Nekateri stolpci tabele *OrderItem*, ki so skupni vsem nivojem v hierarhiji naročila

Stolpec	Uporaba
RecordStatus	Tip vnosa. Za novo naročilo je vedno »New«. [New, Existing, Deleted, Undefined, MappingError]
Removable	Ali je možno <i>OrderItem</i> odstraniti iz strukture. Načeloma je možno odstraniti dodatno ponudbo.
ExternalId	Identifikator storitve sistema za upravljanje s storitvami. Pri procesu novega naročila OMS pridobi ta identifikator šele, ko so storitve vključene.
SpecId/SpecCode/SpecName	Identifikator/Šifra/Ime za prikaz iz produktnega kataloga.
ModifiedOn/EnteredOn	Časovni žigi.
EnteredBy	
ProvisionedOn/ProvisionedBy/ ProvisioningStatusType	Časovni žig/Uporabnik/Status realizacije v zalednem sistemi.
OrderItemType	Tip vnosa, ki označuje tip storitve v sistemu za upravljanje storitev. [Offering, Other, Product, CustomerFacingService, ResourceFacingService, Reactivation]

Tabela *OrderItem* hrani predvsem podatke, ki jih pridobi preko produktnega kataloga iz tipov *ProductOfferingDetails* in *ProductSpecificationDetails*, kot smo to predstavili v poglavju o

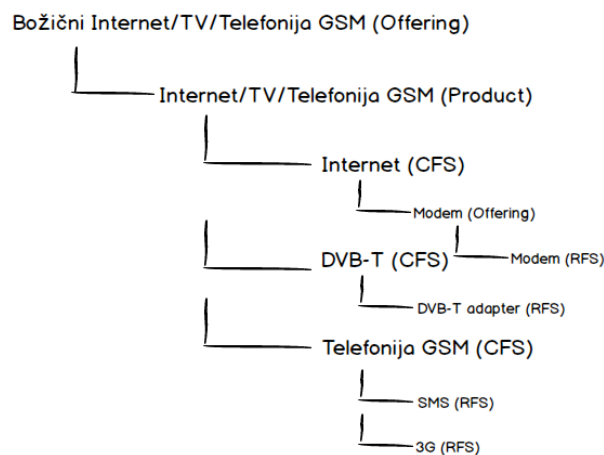
produktnem katalogu. Pri procesu sprememb ali izključitev mora OMS te podatke upariti še s podatki sistema za upravljanje s storitvami: *ProductOffering*, *CustomerFacingService* in *ResourceFacingService*. To mapiranje med podatki je razvidno iz slike 5-2. Kot lahko vidimo, sistem za upravljanje s storitvami hrani manj informacij, kot jih definira struktura produktnega kataloga, in sicer:

- ne ve, iz katere specifikacije produkta izhaja ponudba (*ProductOffering*). Zanj ta podatek nima nobene teže. Na nivoju paketa je zanj pomembno le, za katero konkretno ponudbo gre, saj ta določa cene storitev v paketu,
- ne ve, kateri RFS so del dodatne ponudbe in kateri so del kompozicije specifikacije produkta, a ga tudi to ne sme zanimati, saj je to delo produktnega kataloga.



Slika 5-2: Mapiranje med podatki produktnega kataloga (levo) in sistema za upravljanje s storitvami (desno)

In kako hrani podatke o strukturi OMS? Preprost odgovor na to vprašanje bi bil: s tako strukturo, kot jo predpisuje produktni katalog, a storitve označuje tako kot sistem za upravljanje s storitvami, seveda tam, kjer te informacije obstajajo. Tip vnosa v hierarhiji drevesa naročila je označen v stolpcu *OrderItemType*, kot je to opisano v Tabela 5-1.



Slika 5-3: Struktura naročila v OMS s pripadajočimi tipi (*OrderItemType*)

Prednosti hierarhične strukture *OrderItem* pred namenskimi entitetami:

1. **Enostavna razširljivost podatkovnega modela**
OMS zna delati s poljubno globino drevesa in zavedati se moramo, da niso vsi scenariji tako enostavni kot vnos novega naročila, ki smo ga navedli kot primer.
2. **Podpora generičnemu urejevalniku naročila**
Relativno enostavna implementacija generičnega urejevalnika, ki rekurzivno izriše uporabniški vmesnik za celotno drevo naročila (*OrderItemEditor*).
3. **Enostaven API in manjše število vrstic kode za poslovno logiko**
Poizvedovanje po isti podatkovni strukturi v jeziku C# je lažje kot po različnih, API, razvit za poslovno logiko OMS, pa omogoča pisanje kratke kode za delo z naročilom. Če recimo želimo dobiti iz naročila (*OrderContract*) vse na novo dodane ponudbe, storitve (CFS) in dodatne ponudbe (RFS) iz naročila, napišemo spodnjo kodo v eni vrstici.

```
public static IEnumerable<OrderItem> GetNewItemS(OrderContract oc)
{
    return oc.WhereRecursive(oi => oi.RecordStatus == (int)RecordStatus.New);
}
```

Primer 5-1: Primer kode za pridobivanje novih vnosov *OrderItem*. Metoda *WhereRecursive* je del API-ja za poslovno logiko OMS, ki rekurzivno najde vnose s podanim pogojem

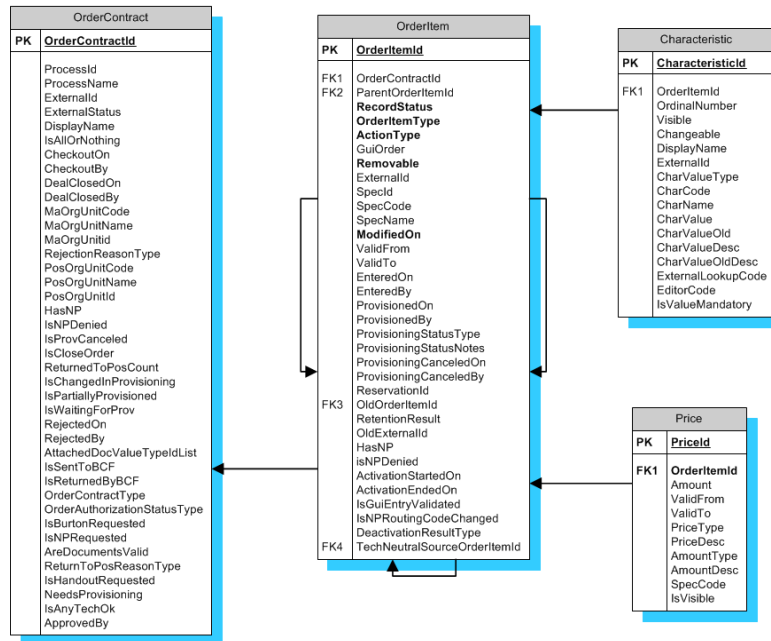
5.3. Characteristic

Karakteristike se nanašajo na vnose v tabeli *OrderItem* z relacijo ena – mnogo. Služijo bodisi izpolnjevanju podatkov v procesu naročanja bodisi zgolj okarakterizirajo storitev in tako vplivajo na poslovni proces. Kot že omenjeno v teoretičnih poglavjih, omogočajo enostavno dodajanje in spreminjanje atributov, ne da bi bila za to potrebna sprememba sheme. Poslovna logika OMS jih razločuje po konstantnem imenu (*CharName*), prav tako pa poteka uparjanje karakteristik med produktnim katalogom in sistemom za upravljanje storitev.

Karakteristike predpiše produktni katalog, ki pove ali je karakteristika vidna, spremenljiva, kakšno je ime za prikaz in njihov vrstni red ipd.

Za urejanje karakteristik na uporabniškem vmesniku skrbi generična kontrola *CharacteristicEditor*, ki jo na uporabniški vmesnik izriše kontrola *OrderItemEditor*. Za karakteristike, za katerimi stoji veliko poslovnih pravil in validacij, je implementirana prilagojena kontrola *CharacteristicEditor*, v osnovni implementaciji pa se ta urejevalnik izriše kot vnosno polje (*CharValueType=STRING*), kot potrditveno polje (*CharValueType=BOOLEAN*) ali kot spustni meni (*CharValueType=ENTITY_ID*).

Če pogledamo podatkovni model, vidimo, da bi lahko bila poizvedba za iskanje naročila po neki karakteristiki (npr. »USERNAME«) zelo potratna, saj bi bilo potrebno opraviti iskanje po tabeli karakteristik, potem pa skozi drevo *OrderItem* priplezati do tabele *OrderContract*, kjer bi našli številko naročila (seveda še ob upoštevanju ostalih pogojev iz ostalih tabel). To OMS rešuje s tabelo *CharacteristicSearch*, ki hrani ime in vrednost pomembnih karakteristik vključno z identifikatorjema *OrderItemId* in *OrderContractId*, h katerima pripada. Ta tabela se osvežuje s pomočjo baznih prožilcev ob vseh operacijah, ki posodablja tabelo.



Slika 5-4: Shema najpomembnejših tabel za vnos naročila

5.4. Price

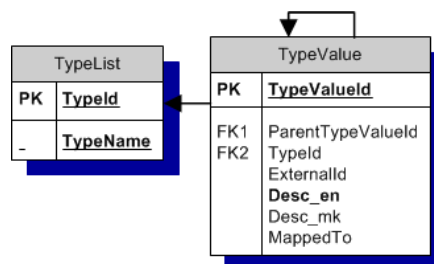
Cene so lastnost ponudbe (*OrderItemType=Offering*), bodisi samostojne bodisi dodatne, in so pridobljene iz produktnega kataloga.

5.5. Customer

Glede entitete za stranko v OMS velja omeniti, da je za vsako naročilo v OMS vedno znova kreirana stranka, ne glede na to, ali je ta konkretna stranka že opravila naročilo prej. Prednost tega je dejstvo, da so podatki o stranki hranjeni vedno zgolj na enem sistemu – CRM, na katerega se zanašajo vsi ostali sistemi. Tudi to je ideja deljenega informacijskega modela.

5.6. Globalen šifrant

Z implementacijo globalnega šifranta *TypeList-TypeValue* se izognemo velikemu številu šifrantov in nepotrebnim dodatnim relacijam v podatkovnem modelu. Nepotrebnim relacijam zato, ker vzdrževanje namenskih šifrantov terja več časa in ker bi med vsakim dostopom do teh šifrantov (npr. do polja za opis) bilo potrebno delati stike, ki so potratni z vidika zmogljivosti. Globalen šifrant se naloži v pomnilnik ob startu spletne aplikacije na strežniku, tako da je dostop do njega izredno hiter.



Slika 5-5: Globalen šifrant v OMS

Poleg omenjenega pa prinaša dodatne prednosti. Omogoča namreč, da je v OMS zgolj ena implementacija za sinhronizacijo šifrantov z zunanjimi sistemi, podpira pa tudi hierarhične šifrante. Tabela *TypeValue* omogoča večjezično podporo.

6. Zaključek

V diplomskem delu sem predstavil ključne koncepte za uspešno implementacijo sistema za upravljanje z naročili, ki poleg vnosnega dela skrbi tudi za proces realizacije naročila, osredotočil pa sem se predvsem na deljen informacijski model. V telekomunikacijskem podjetju, ki sem ga predstavil, je nivo prizadevanja za sledenje priporočil modela SID za namen integracij visok, kar je nedvomno pohvalno.

Na projektu, ki je tema tega diplomskega dela, sam delam že skoraj tri leta. Na podlagi pridobljenih izkušenj sem spoznal, da je v analizo, specifikacijo in implementacijo integracij, pri katerih shema vmesnikov temelji na priporočilih SID, potrebno vložiti manj truda kot v implementacijo tistih, ki vzpostavljajo povsem svojo shemo. To sem predstavil tudi na primeru sistema za subvencije. Bistven doprinos, ki ga vidim pri sledenju priporočilom ogrodja SID, je enotna terminologija, saj je razumevanje ostalih sistemov tako bistveno lažje. Primerjava z esperanto jezikom iz te diplomske naloge se mi zdi odlična.

Na žalost pa sledenje priporočilom ni popolno. Veliko razvijalcev (tako OMS kot tudi ostalih sistemov) shemo pozna zgolj bežno, saj je ta izredno obsežna, njeno razumevanje pa terja veliko časa, kar sem ugotovil tudi med pisanjem te diplomske naloge. Tako se včasih ob tesno zastavljenih rokih za izvedbo projektov ne razmišlja v smeri preučevanja razumljive in priporočilom sledljive implementacije, ampak take, ki je realizirana v najkrajšem možnem času. Zato bi si v sodelovanju s telekomunikacijskim podjetjem želeli, da bi ti imeli na svoji strani strokovnjaka, ki bi to področje obvladoval in bi skrbel za pomoč pri definiciji vmesnikov ter izmenjavo podatkov med vsemi (tudi zunanjimi) sistemi. Se pa zavedam, da je to zaradi obsežnosti celotne informacijske domene precejšen izziv.

Na podlagi razvoja informacijske podpore v zadnjih letih se je naš ponudnik telekomunikacijskih storitev izkazal za zelo fleksibilno podjetje, ki investira v informacijsko podporo, skrbi, kako so realizirani poslovni procesi naročanja in je pripravljen na spremembe na tržišču. Ravno v obdobju pisanja diplomske naloge se na tržišču našega podjetja soočajo z ukinitvijo najemanja linij za storitev PSTN s strani nacionalnega operaterja. V kratkem bo namreč realizacija storitve na najeti liniji možna zgolj na tehnologiji VoIP. Naš operater je bil na tako spremembo dobro pripravljen in uvedba prodaje novih storitev VoIP je bila zato izvedena v izredno hitrem času, celo pred rokom, ki ga je predpisal nacionalni operater. Tak rezultat kaže na to, da je informacijska podpora dobro organizirana, naš ponudnik storitev pa dobro pripravljen na nadaljnje izzive.

Literatura

- [1] Y. Kim, D.-H. Lee, "Convergence Technologies and the Layered Policy Model: Implication for Regulating Future Communications," v *Conference Papers - International Communication Association*, 2006. Dostopno na: <http://articles.portal-tol.com/english-language-en/Convergence%20%28telecommunications%29>
- [2] (2012) TM Forum - About the TM Forum. Dostopno na: <http://www.tmforum.org/AbouttheTMForum/730/home.html>.
- [3] K. Misra, *OSS for Telecom Networks: An Introduction to Network Management*, Springer, 2004.
- [4] Z. A. Al Mani, *Introduction to eTOM - eTOM as a model to increase Telecom Co. efficiency and effectiveness*, 2009. Dostopno na: <http://www.scribd.com/doc/80230575/Introduction-of-eTOM>
- [5] (2012) Progress Software *DataXtend Browser, version: 8.5.1*. Dostopno na: <http://www.progress.com/en/dataxtend-landing/downloads/download-sid-browser.html>
- [6] *TM Forum*, "Shared Information/Data (SID) Model - Common Business Entity Definitions - Business Interaction", *GB922 Addendum 1B1*, maj 2009.
- [7] *TM Forum*, "Shared Information/Data (SID) Model, Addendum 0 - SID Primer", *GB922 Addendum-0*, avgust 2004.
- [8] Y. T. Lee, "Information Modeling: From Design to Implementation", *National Institute of Standards and Technology*. Dostopno na: <http://www.mel.nist.gov/msidlibrary/doc/tina99im.pdf>
- [9] A. Pras, J. Schoenwaelder, "On the Difference between Information Models and Data Models", Network Working Group, 2003.
- [10] *TM Forum*, "Shared Information/Data (SID) Model - Business View Concepts, Principles and Domains", *GB922 TM Forum Approved Version 7.7*, maj 2009.
- [11] (2004) W3C, "Web Services Glossary - W3C Working Group Note 11 February 2004". Dostopno na: <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.
- [12] M. B. Jurič, K. Pant, "*Business Process Execution Language for Web Services*", Packt Publishing Ltd., 2006.
- [13] (2007) W3C, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)". Dostopno na: <http://www.w3.org/TR/soap12-part1/>.
- [14] (2001) W3C, "Web Services Description Language (WSDL) 1.1,". Dostopno na: <http://www.w3.org/TR/wsdl>.
- [15] T. Sečnik, "*Načrtovanje in izvedba podpore poslovnim procesom z orodji BPM*", diplomsko delo Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2012.
- [16] A. Smith, "*Bogastvo narodov*", Ljubljana: Studia Humanitatis, 2010.

- [17] K. Rashid, "*Business Process Management, A Practical Guide*", Meghan-Kiffer Press, 2004.
- [18] U. Nabernik, "*Optimizacija avtomatiziranih poslovnih procesov z uporabo poslovne inteligence*", magistrsko delo Ekonomska fakulteta, Univerza v Ljubljani, 2010.
- [19] B. Vukšić, A. Kovačič, "*Management poslovnih procesov: Prenova in informatizacija poslovanja*", GV Založba, 2005.
- [20] A. Daneu, "*Vpliv managementa poslovnih procesov na poslovanje organizacije*", magistrsko delo, Ekonomska fakulteta, Univerza v Ljubljani, 2006.
- [21] J. W. John P. Reilly, "*Application Integration Using the SID*", TM Forum, 2008.
- [22] TM Forum, "Shared Information/Data (SID) Model - System View Concepts and Principles", *GB922 TM Forum Approved Version 7.7*, maj 2009.
- [23] (2008) A. McFadyen, "Telecoms enterprise information modelling". Dostopno na: <http://telecomseim.blogspot.com/2008/08/eai-and-sid.html>.
- [24] J. P. Reilly, "*Getting Started with the SID - A SID Modeler's Guide*", TM Forum, 2007.
- [25] TM Forum, Product Business Entity Definitions, *GB922 TM Forum Approved Version 7.9*, maj 2009.
- [26] (2012) S. Mewada, "From Bespoke to Standard, Embracing OSS/BSS Standards,". Dostopno na: http://www.pipelinepub.com/1011/OSS_BSS/NetCracker-on-Standards-1.php

Slike

Slika 2-1: Model nivoja 0 ogrodja eTOM z entitetami, ki se raztezajo horizontalno skozi procese eTOM.....	7
Slika 2-2: Razmerje med informacijskim in podatkovnim modelom.....	9
Slika 2-3: Povezava med eTOM procesnimi vertikalami (levo) in domenami poslovnega pogleda SID (desno).....	10
Slika 2-4: Struktura sporočila SOAP.....	12
Slika 3-1: NGOSS pogledi	17
Slika 3-2: domene SID in agregirane poslovne entitete prvega nivoja za vsako domeno, povzete po TM Forumu.....	19
Slika 3-3: Primer podaljška na sistemskem nivoju.....	21
Slika 3-4: Prikaz maksimalnega števila povezav (12) med štirimi sistemi	22
Slika 3-5: Modelirni vzorec »specifikacija entitete/entiteta«	24
Slika 3-6: Primer uporabe modelirnega vzorca specifikacija entitete/entiteta v domeni vira na poslovni entiteti PhysicalResource iz sheme SID	24
Slika 3-7: Modelirni pristop kompozitno/atomarno	25
Slika 3-8: Entiteta specifikacija karakteristik/entiteta karakteristik	26
Slika 3-9: Prikaz pristopov načrtovanja na ABE specifikacija produkta (ProductSpecification) ..	27
Slika 3-10: Hierarhična struktura, ki jo omogoča specifikacija produkta	28
Slika 4-1: Plasti rešitve OMS.....	31
Slika 4-2: Poenostavljen proces novega naročila	33
Slika 4-3: Prikaz povezovanja OMS z drugimi sistemi v prvem in tretjem koraku poslovnega procesa vnosa novega naročila	34
Slika 4-4: Primer ponudbe »Božični Internet/TV/Telefonija GSM«.....	35
Slika 4-5: Rezultat produktnega kataloga, pridobljena za ponudbo »Božični Internet/TV/Telefonija GSM«.....	36
Slika 4-6: TOCP struktura paketa »Božični internet/TV/Telefonija GSM«.....	38
Slika 4-7: Primer objekta, temelječega na strukturi SID, za naročilo zamenjavo opreme, kot ga OMS pošlje sistemu za upravljenje storitev	38
Slika 5-1: Del podatkovne sheme sistema za OMS s tabelami za podporo osnovnih poslovnih procesov naročanja in dve krovni entiteti podpornih procesov (Archiving, Complaint).....	42
Slika 5-2: Mapiranje med podatki produktnega kataloga in sistema za upravljanje s storitvami	44
Slika 5-3: Struktura naročila v OMS s pripadajočimi tipi (<i>OrderItemType</i>).....	44
Slika 5-4: Shema najpomembnejših tabel za vnos naročila.....	46
Slika 5-5: Globalen šifrant v OMS.....	46

Tabele

Tabela 2-1: Razlike med BPR in BPM [19]	15
Tabela 3-1: Število potrebnih translacij za komunikacijo med več sistemi	23
Tabela 4-1: Poslovni procesi, ki jih podpira OMS	32
Tabela 5-1: Nekateri stolpci tabele OrderItem, ki so skupni vsem nivojem v hierarhiji naročila ..	43