



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
У НОВОМ САДУ



Вељко Илкић

**Једно решење приказа
мултимедијалног садржаја на
пријемнику дигиталног ТВ
сигнала заснованог на Андроид
платформи**

МАСТЕР РАД

Нови Сад, 2013

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА МАСТЕР РАД	Датум:

(Податке уноси предметни наставник - ментор)

СТУДИЈСКИ ПРОГРАМ:	Рачунарство и аутоматика
РУКОВОДИЛАЦ СТУДИЈСКОГ ПРОГРАМА:	Никола Јорговановић

Студент:	Вељко Илкић	Број индекса:	E2 9/2012
Област:	Програмска подршка у телевизији и обради слике 2		
Ментор:	Никола Теслић		
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА МАСТЕР РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА: - проблем – тема рада; - начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;			

НАСЛОВ МАСТЕР РАДА:

Једно решење приказа мултимедијалног садржаја на пријемнику дигиталног ТВ сигнала заснованог на Андроид платформи

ТЕКСТ ЗАДАТКА:

Реализовати програмску подршку за приказ и руковање мултимедијалним садржајем на дигиталном пријемнику телевизијског сигнала заснованог на Андроид платформи. Подршка треба да обједини све доступне мултимедијалне садржаје у јединствени скуп (телевизијске канале, радио канале, Андроид апликације, послужеоце из кућне мреже и њихов садржај, итд.), и да омогући једноставно руковање и репродукцију коришћењем јединствене корисничке спреге. Доступан мултимедијални садржај мора бити третиран на исти начин са стране корисничке апликације и крајњег корисника.
--

Руководилац студијског програма:	Ментор рада:

Примерак за: ☐ - Студента; ☐ - Ментора
--



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Дипломски – мастер рад		
Аутор, АУ:	Вељко Илкић		
Ментор, МН:	Проф. др Никола Теслић		
Наслов рада, НР:	Једно решење приказа мултимедијалног садржаја на пријемнику дигиталног ТВ сигнала заснованог на Андроид платформи		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УП:	Војводина		
Година, ГО:	2013.		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО:	7/88/0/0/16/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Телевизија, програмска подршка телевизијског пријемника, Јава, Андроид, мултимедијални садржај		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	Ovaj rad prikazuje jedno rešenje prikupljanja, prikaza, rukovanja i reprodukovanja različitog multimedijalnog sadržaja na Android prijemniku digitalnog televizijskog signala zasnovanog na Android platformi.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	Проф. др Јелена Ковачевић	
	Члан:	Проф. др Иштван Пап	Потпис ментора
	Члан, ментор:	Проф. др Никола Теслић	



KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	Monographic publication		
Type of record, TR :	Textual printed material		
Contents code, CC :	Master Thesis		
Author, AU :	Veljko Ilić		
Mentor, MN :	PhD. Nikola Teslić		
Title, TI :	One solution of realization and representation various multimedia content on Android based digital STB		
Language of text, LT :	Serbian		
Language of abstract, LA :	Serbian		
Country of publication, CP :	Republic of Serbia		
Locality of publication, LP :	Vojvodina		
Publication year, PY :	2013.		
Publisher, PB :	Author's reprint		
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD :	7/88/0/0/16/0/0		
Scientific field, SF :	Electrical Engineering		
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW :	Television, middleware, Java, Android, Multimedia		
UC			
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N :			
Abstract, AB :	This paper represents one solution of collecting, representing, manipulating and reproduction of different multimedia content on Android based digital STB.		
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	PhD Jelena Kovačević	
	Member:	PhD Ištvan Pap	Menthor's sign
	Member, Mentor:	PhD Nikola Teslić	

Zahvalnost

Zahvalnost na podršci dugujem mentoru, prof. dr Nikoli Tesliću, redovnom profesoru Fakulteta tehničkih nauka Univerziteta u Novom Sadu.

Podjednaku zahvalnost na korisnim informacijama, sugestijama i savetima dugujem Tomislavu Maruni.

Posebnu zahvalnost dugujem svojoj porodici i prijateljima na podršci.

U Novom Sadu, maja 2013. godine

Veljko Ilkić

SADRŽAJ

1. Uvod	1
2. Teorijske osnove	5
2.1 Razvoj digitalne televizije	5
2.2 Android platforma i DTV programska podrška	8
3. Koncept rešenja	19
3.1 Opis ciljne platform	19
3.2 Ključne komponente sistema	20
4. Programsko rešenje	26
4.1 Korisnička aplikacija	26
4.2 Aplikativni servis	34
4.2.1 Android aplikacije	37
4.2.2 Android namenske grafičke korisničke kontrole	40
4.2.3 DLNA i lokalni sistem podataka	43
4.2.4 DVB sadržaj	48

4.2.5	IP sadržaj	53
4.2.6	PVR sadržaj	55
4.2.7	Zaštita od neželjenog pristupa sadržaju	59
5.	Ispitivanje i verifikacija	62
5.1	Prikupljanje i grafički prikaz multimedijalnog sadržaja ...	63
5.2	Reprodukcija multimedijalnog sadržaja	65
5.3	Zaštita od neželjenog pristupa sadržaju	69
5.3.1	Zaključavanje željenog sadržaja	70
5.3.2	Prikaz zaključanog sadržaja unutar grafičke sprege	71
5.3.3	Reprodukcija zaključanog sadržaja	72
6.	Zaključak	73
7.	Literatura	75

SPISAK SLIKA

Slika 1 Programska podrška Android platforme sa DTV proširenjima	9
Slika 2 Životni ciklus Android aktivnosti	11
Slika 3 Životni ciklus Android servisa	14
Slika 4 Izgled ciljne platforme	20
Slika 5 Prikaz organizacije modula programskog rešenja	21
Slika 6 Organizacija modula unutar korisničke aplikacije	31
Slika 7 Organizacija modula u okviru aplikativnog servisa	37
Slika 8 Organizacija PVR modula u sklopu programske podrške	56
Slika 9 Grafička sprega sa prikazanim Android i DVB sadržajem	64
Slika 10 Grafička sprega sa prikazanim sadržajem sa sistema podataka i DLNA mreže	64
Slika 11 Izgled korisničke aplikacije pri reprodukciji televizijskog sadržaja	66

Slika 12 Izgled korisničke aplikacije pri reprodukciji snimljenog PVR sadržaja	67
Slika 13 Izgled grafičke sprege za prikaz galerije slika	68
Slika 14 Izgled korisničke aplikacije sa aktivnim Android namenskim grafičkim kontrolama	69
Slika 15 Korisnička sprega za zaključavanje multimedijalnog sadržaja.....	70
Slika 16 Korisnička sprega sa zaključanim multimedijalnim sadržajem	71
Slika 17 Korisnička sprega za iniciranje zaključanog sadržaja	72

SKRAĆENICE

STB *Set-Top-Box*, Prijemnik digitalnog televizijskog signala

DTV *Digital Television*, Digitalna televizija

DVB *Digital Video Broadcasting*, Standard digitalne televizije

MHEG *Multimedia and Hypermedia Experts Group*, Skup standarda za prikaz multimedijalnog sadržaja

VOD *Video On Demand*, Servis za dostavljanje video sadržaja

AIDL *Android Interface Definition Language*, Jezik sa opis programske sprege

JNI *Java Native Interface*, Sprega Java programskog jezika i C koda

MAL	<i>Middleware Abstraction Layer</i> , Apstrakcija programske podrške televizijskog prijemnika
CHAL	<i>Comedia Hardware Adaptation Layer</i> , Adapcioni modul fizičke arhitekture
DLNA	<i>Digital Living Network Alliance</i> , Nefitabilna organizacija, koja omogućuje razmenu multimedijalnih podataka
UPnP	<i>Universal Plug And Play Protocol</i> , Mrežni protokol
HTTP	<i>Hypertext Transfer Protocol</i> , Aplikativni protokol za komunikaciju
IP	<i>Internet Protocol</i> , Skup pravila za razmenu podataka preko mreže
TCP	<i>Transmission Control Protocol</i> , Komunikacioni protokol
UDP	<i>User Datagram Protocol</i> , Komunikacioni protokol
HTML	<i>HyperText Markup Language</i> , Jezik za razvoj web stranica
XML	<i>Extensible Markup Language</i> , Jezik koji definiše pravila za dekodovanje dokumenata određenog formata
JNI	<i>Java Native Interface</i> , Programska sprega u okviru Android operativnog sistema
DVM	<i>Dalvik Virtual Machine</i> , Virtualna mašina unutar Android operativnog sistema
IPC	<i>Inter - Process Communication</i> , Mehanizam za komunikaciju između procesa u Android operativnom sistemu
DMS	<i>Digital Media Server</i> , Usluživač u DLNA mreži

DMP DLNA mreži	<i>Digital Media Player</i> , Modul za reprodukciju u
DMC DLNA mreži	<i>Digital Media Controller</i> , Modul za kontrolu u
DMP_r DLNA mreže	<i>Digital Media Printer</i> , Mrežni štampač u okviru
SIP	<i>Session Initiation Protocol</i> , Mrežni protokol
VoIP	<i>Voice over IP</i> , Komunikacioni protokol
FTP	<i>File Transfer Protocol</i> , Komunikacioni protokol
APK	<i>Android package</i> , Ekstenzija Android aplikacije

1. Uvod

Ovaj rad prikazuje jedno rešenje grafičkog prikaza, reprodukciju, način rukovanja i kontrolu nad različitim multimedijalnim sadržajem na prijemniku digitalnog televizijskog signala zasnovanog na Android platformi.

Ovo rešenje objedinjuje različite multimedijalne sadržaje sa stanovništva implementacije i tipa podataka u jedinstveni skup, gde se televizijski kanali, Android aplikacije i namenske grafičke korisničke kontrole (eng. *Widget applications*), DLNA (eng. *Digital Living Network Alliance*) poslužioc, USB (eng. *Universal Serial Bus*) uređaji, PVR (eng. *Personal Video Recorder*) datoteke, slike, video i audio zapisi tretiraju na isti način sa stanovništva krajnjeg korisnika korišćenjem jedinstvene grafičke sprege.

Dato rešenje je realizovano na RK-2010 platformi, proizvođača RT-RK [1].

Prilikom razvoja programske podrške za Android platformu, prirodno okruženje za razvoj predstavlja Java programski jezik. Ovaj jezik obično zahteva fizičku arhitekturu sa više resursa, što često nije slučaj sa sistemima u realnom vremenu. Da bi iskoristili sve dostupne resurse, koriste se programski jezici poput C/C++, ali ovim rešenjima nedostaje olakšana prenosivost koda i pojednostavljeno otklanjanje grešaka [2]. U poslednje vreme, platforme za rad u realnom vremenu dobijaju sve više procesorske snage sa više procesorskih jezgara i većim radnim taktom. Iz tog razloga moguće je koristiti programske jezike poput Java programskog jezika radi zadovoljenja većine potreba.

Pojedini poslovi se i dalje moraju obavljati u programskom okruženju C/C++, a sve u cilju približavanja granici mogućnosti same fizičke arhitekture [3].

Ovo programsko rešenje razvijeno je u sklopu Android4TV projekta [4].

Android4TV definiše poseban pristup integraciji digitalne televizije u okruženje Android platforme. Android4TV proširenja ne utiču na aplikativno okruženje operativnog sistema Android platforme, već nudi standardizovani model implementacije aplikacija namenjenih televizijskim prijemnicima.

Funkcionalnost Android4TV sistema čine:

- Funkcionalnosti digitalne televizije (pretraga kanala, liste kanala, elektronski programski vodič, teletekst informacije, itd.)
- Istovremeni pristup standardnim Android aplikacijama i televizijskom sadržaju

- Prikupljanje podataka putem aktivne internet konekcije
- Podršku za interaktivne sadržaje (HbbTV, MHEG, VOD servisi)
- Prepozavanje asinhronih događaja iz drugih Android aplikacija
- Preuzimanje drugih Android aplikacija sa Android marketa

Android4TV koncept omogućuje razvoj vizuelno bogatih aplikacija u Java programskom jeziku (3D grafika, namenske grafičke kontrole i aplikacije, internet pretraživači, itd.).

Dato rešenje koristi prednosti drugih Android aplikacija koje dolaze sa Android platformom, kao što su: e-mail klijenti, različite socijalne mreže, video igre, itd.

Rezultati rada su provereni uz pomoć različitih skupova ispitnih slučajeva, gde je utvrđeno identično, ispravno ponašanje, čime je potvrđena ispravnost datog rešenja.

Ovaj rad sačinjen je od sedam poglavlja.

Prvo poglavlje sadrži osnovne podatke o radu i opis rada.

Drugo poglavlje sadrži opis Android platforme i opis programske podrške za digitalnu televiziju i kratak istorijat razvoja digitalne televizije.

Treće poglavlje daje opis ciljne platforme, pregled slojeva programske podrške, koji detaljnije opisuje njenu organizaciju i način realizacije, i funkcionalnost programskih modula za rukovanje dostupnim multimedijalnim sadržajem.

Četvrto poglavlje sadrži detaljan opis programskog rešenja, odnosno realizacije programskih modula za rukovanje multimedijalnim sadržajem u okviru programske podrške televizijskog prijemnika kao i viših slojeva programske podrške, potrebnih za kompletnu funkcionalnost na prijemniku digitalnog televizijskog signala na Android platformi.

Peto poglavlje je sačinjeno od opisa verifikacije funkcionalnosti i provera koje su sprovedene nakon realizacije modula za rukovanje multimedijalnim sadržajem. U ovom poglavlju su opisani i dobijeni rezultati.

Šesto poglavlje sadrži kratak pregled onoga što je urađeno u okviru ovog rada, kao i to kakvi su dalji pravci razvoja istog.

U sedmom poglavlju je dat spisak korišćene literature tokom izrade ovog rada.

2. Teorijske osnove

U ovom poglavlju dat je kratak istorijat razvoja digitalne televizije i opis Android i DTV (eng. *Digital Television*) programske podrške, kao i opis svih važnih modula, koji čine programsku podršku ovog programskog rešenja.

2.1 Razvoj digitalne televizije

Reč televizija prvi put se koristi 1900 godine. Početak razvoja televizije potiče od 1923. godine kada su u SAD i Engleskoj ostvareni prvi prenosi crno-belih silueta (mehanička televizija). Elektronska televizija počinje sa eksperimentalnim radom 1935. godine (analogna televizija, osnovni sistem koji se i danas koristi). Emitovanje počinje u Nemačkoj, Engleskoj, SAD, ali je tokom ratnih godina prekinuto. Od 1945. godine televizija ponovo oživljava u savršenijem obliku zahvaljujući televizijskoj tehnici koja je razvijana u ratne svrhe. Značajnu prekretnicu predstavlja 1954. godina kada u SAD počinje emitovanje

prvog programa televizije u boji. Standardi na kojima se još uvek zasniva današnja analogna televizija postavljeni su pre skoro pola veka. Od tada se skoro ništa nije promenilo. Televizijski (u daljem tekstu TV) prijemnici su napravljeni od analognih komponenti (otpornici, tranzistori, itd).

Zbog toga postoje brojni novi standardi koji se trude da prevaziđu ograničenja današnje televizije. Osnovna karakteristika digitalne televizije predstavlja prenos zvuka i slike sa dodatnim informacijama u digitalnom formatu.

Digitalni prenos obezbeđuje bolji kvalitet slike i zvuka koji više ne mogu biti u prenosu ometani interferencijom sa drugim signalima bez obzira na rastojanje na koje se slika i zvuk prenose. Slika i zvuk koju digitalni signal nosi su isti kao i na izvoru emitovanja, sve dok signal ne postane toliko slab da prijem više nije moguć.

Digitalni signal se može prenositi zemaljskim, kablovskim i satelitskim putem.

Pojava i uspostavljanje standarda koji se koriste u digitalnoj televiziji vezuje se za poslednju dekadu prošlog veka. Najvažniji standardi digitalne televizije (u daljem tekstu DTV) :

- DVB (eng. *Digital Video Broadcasting*) [5] je grupa standarda proizašla iz međunarodne inicijative.

Ovaj standard se dominantno koristi u Evropi ali i u većem delu sveta.

DVB standardi za prenos digitalne televizije:

1. DVB-S / S2 (eng. *Satellite*) – definiše DTV satelitski prenos. Za transportno kodovanje DVB-S

- koristi MPEG-2, a radi u frekvencijskom opsegu od 11/12 GHz.
2. DVB-C (eng. *Cable*) – definiše DTV prenos putem digitalne kablovske mreže. Ovaj standard za prenos slike i zvuka koristi MPEG-2 ili MPEG-4 uz QAM modulaciju sa kodiranjem kanala.
 3. DVB-T / T2 (eng. *Terrestrial*) – definiše zemaljski DTV prenos putem UHF/VHF. Ovaj sistem prenosi komprimovan digitalni zvuk, video i druge podatke u MPEG prenosnom toku, koristeći COFDM modulaciju. Umesto prenosa podataka na jednoj radio frekvenciji nosioca, OFDM deli digitalni tok podataka na valiki broj manjih digitalnih tokova, koji svaki digitalno moduliše skup blisko odvojenih susednih frekvencija za prenos. U slučaju DVB-T, postoje dve mogućnosti za broj nosilaca poznatih kao 2K-mod ili 8K-mod. To su zapravo 1705 ili 6817 nosilaca koji su približno 4 kHz, odnosno 1 kHz, odvojeni.
 4. DVB-H (eng. *Handheld*)– DTV za prenosne uređaje kao što su mobilni telefoni.
- ATSC (eng. *Advanced Television System Committee*) standardi se primenjuju u digitalnom TV prenosu zemaljskim vezama u SAD, Kanadi, Meksiku a njihova primena se razmatra i u drugim zemljama.

- OCAP (eng. *Open Cable Protocol*) je standard koji se koristi u SAD u kablovskoj digitalnoj televiziji.

Digitalna televizija obezbeđuje mnogo bolji kvalitet slike i zvuka i njihovu različitu prezentaciju, mogućnost izbora formata slike (4:3 ili 16:9) kao i zvuka (mono, stereo ili "surround").

Pojava digitalnih prijemnika omogućila je izvršavanje računarskih aplikacija na televizijskim platformama, pisanih u različitim jezicima kao što su Java, HTML (eng. *HyperText Markup Language*), itd.

2.2 Android platforma i DTV programska podrška

U ovom poglavlju dat je opis operativnog sistema Android platforme i osnovnih modula koji ulaze u sklop programske podrške ovog rešenja.

Android predstavlja programsku podršku namenjenu prvenstveno za mobilne uređaje, koji u sebe uključuje operativni sistem, sprežni sloj za povezivanje Java i C programskog jezika i pojedine aplikacije koje su dostupne korisnicima. Sistem je zasnovan na Linux jezgru.

Android predstavlja trenutno najrasprostanjeniji sistem za mobilne uređaje, ali su sve češće dostupna određena prilagođenja operativnog sistema Android platforme za druge uređaje, koji rade u realnom vremenu poput: tablet računara, televizijskih prijemnika i drugih.

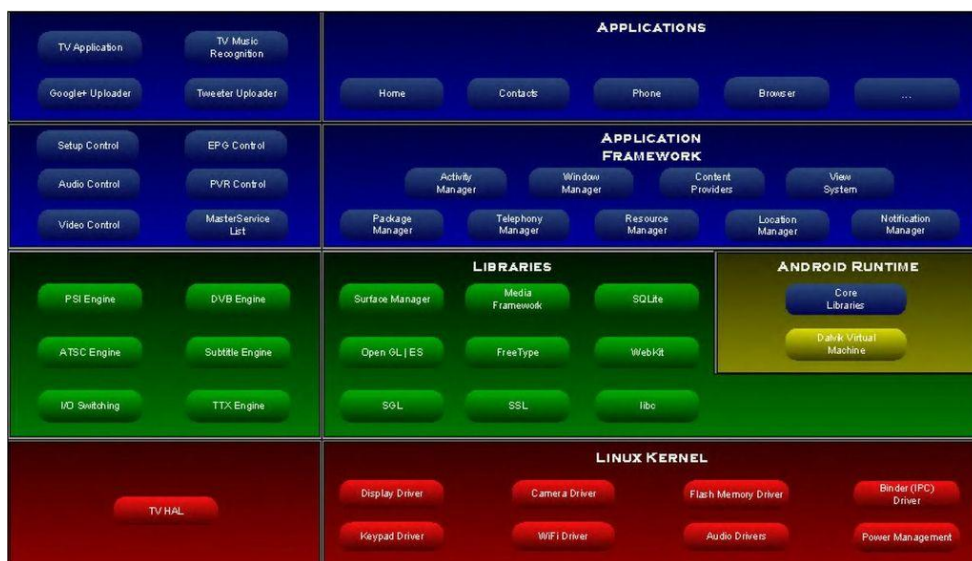
Na slici 1 dat je prikaz značajnijih komponenti operativnog sistema Android platforme, sa proširenjima koje unosi programska podrška digitalnog televizijskog prijemnika.

Na postojeće slojeve operativnog sistema Android platforme dodat je deo programske podrške za digitalne televizijske prijemnike. Na ovaj

način postojeća rešenja programske podrške televizijskog prijemnika moguće je iskoristiti u okviru Android platforme.

Ovo rešenje sastoji se iz više programskih slojeva, gde svaki nivo programske podrške unosi dodatni nivo apstrakcije i enkapsulacije podataka.

Detaljni opis osnovnih modula programske podrške dat je u daljem tekstu.



Slika 1 Programska podrška Android platforme sa DTV proširenjima

Vezu ka krajnjem korisniku digitalnog televizijskog prijemnika smeštena je u aplikativnom sloju. Android aplikacija je objektno orijetnisana i napisana je u Java programskom jeziku. Ona predstavlja veoma značajan sloj programskog steka, jer omogućava razvijanje grafičke sprege, što omogućuje vizuelizaciju funkcionalnosti celokupnog sistema, pa se njena funkcionalnosti može iskoristiti i za verifikaciju ispravnosti čitavog sistema.

Android aplikacija sakriva implementaciju ostalih slojeva i krajnji korisnik nema uvid u način njihovog funkcionisanja.

Operativni sistem Android platforme predstavlja višekorisnički Linux sistem, gde se svaka aplikacija posmatra kao nezavistan korisnik, jer se svaki proces unutar sistema izvršava u nezavisnoj virtuelnoj mašini (eng. *DVM – Dalvik Virtual Machine*).

Sistem svakoj aplikaciji dodeljuje jedinstvenu oznaku, čime štiti podatke drugih aplikacija od neželjenog pristupa.

Ovim se definiše veoma bezbedno programsko okruženje, gde aplikacija nema mogućnosti da pristupa delovima operativnog sistema, za koje nema dozvolu.

Najvažniji gradivni blokovi svake Android aplikaciju su aktivnost (eng. *Activity*) i servis (eng. *Service*).

Svaki gradivni blok ima posebnu namenu i funkcionalnost, a zajedno određuju opšte ponašanje celokupne aplikacije.

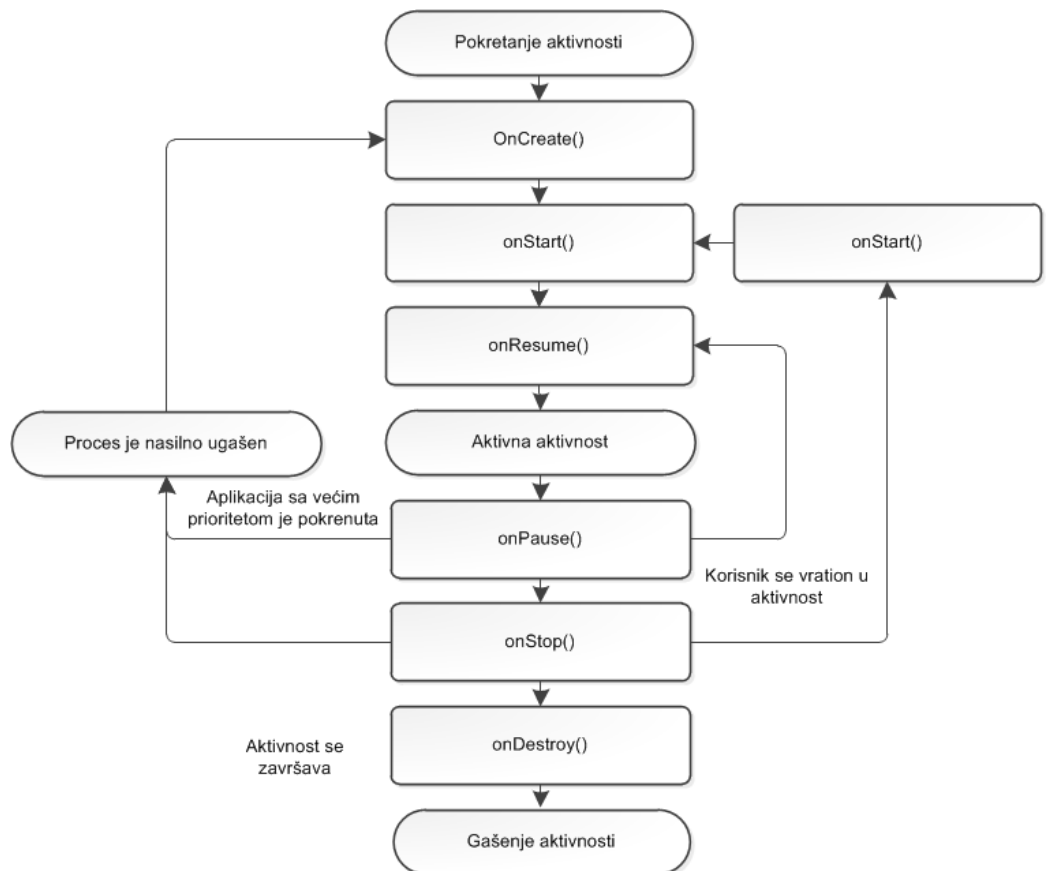
Aktivnost je blok u okviru aplikacije koji omogućuje interakciju sa krajnjim korisnikom, gde korisnik zadaje odgovarajuće akcije putem podržanih periferija (miš, tastatura, dodir, itd.). Svaka aktivnost sadrži prozor koji se koristi za iscertavanje korisničke sprege. Prozor obično zauzima ceo ekran, ali postoji mogućnost da on bude i manji ili da bude postavljen u ravni iznad drugih prozora (eng. *Z-ordering*).

Android aplikacija se obično sastoji iz više međusobno povezanih aktivnosti, što nije slučaj u ovom programskom rešenju, gde se TV korisnička aplikacija sastoji samo iz jedne aktivnosti, o čemu će detaljnije biti reči u narednim poglavljima.

Aktivnost se definiše nasleđivanjem klase *Activity*, koja je definisana u okviru Android razvojnog okruženja i implementiranjem

određenih povratnih metoda koje definišu životni ciklus svake aktivnosti [6].

Životni ciklus Android aktivnosti prikazan je na slici 2.



Slika 2 Životni ciklus Android aktivnosti

`onCreate()` - Poziva se kada se aktivnost prvi put pokreće, tj. stvara. U okviru ove metode se obično vrše inicijalizacije svih važnih polja i grafičkih komponenti.

`onRestart()` - Poziva se pre ponovnog pokretanja, nakon prethodnog zaustavljanja aktivnosti. Uvek je praćena pozivom `onStart()` metode.

`onStart()` - Poziva se pre nego što aktivnost postane vidljiva krajnjem korisniku.

`onResume()` - Poziva se pre nego što aktivnost započne interakciju sa korisnikom. Uvek je praćena `onPause()` metodom.

`onPause()` - Poziva se pre završavanja ili pauziranja aktivnosti. Obično se koristi za skladištenje važnih podataka i određenih stanja. Mogu je slediti pozivi metoda `onResume()` ako će se aktivnost ponovo pokrenuti, ili `onStop()` ako je aktivnost u potpunosti završena.

`onStop()` - Poziva se kada aktivnosti nije vidljiva za korisnika.

`onDestroy()` - Poziva se pre zaustavljanja i uništavanja aktivnosti. Ujedno predstavlja poslednji poziv koji aktivnost dobije u toku životnog ciklusa.

Android servis je komponenta koja se, za razliku od Android aktivnosti, izvršava u pozadini i služi za izvršavanje složenih i dugih operacija ili za interakciju sa drugim procesima (eng. *IPC – Inter Process Communication*).

Servis ne poseduje grafičku korisničku spregu prema krajnjem korisniku, što značajno utiče na povećanje njegovih performansi.

Neka druga komponenta, kao što je aktivnost, može pokrenuti servis i vršiti interakciju sa njim nakon međusobnog povezivanja.

Servis u suštini može posedovati dva stanja, tj. dva režima rada:

Pokrenut servis - Servis je pokrenut kada ga neka aplikativna komponenta pokrene pozivom `startService()` metode. Kada je jednom pokrenut, servis se izvršava u pozadini, iako komponenta koja ga je

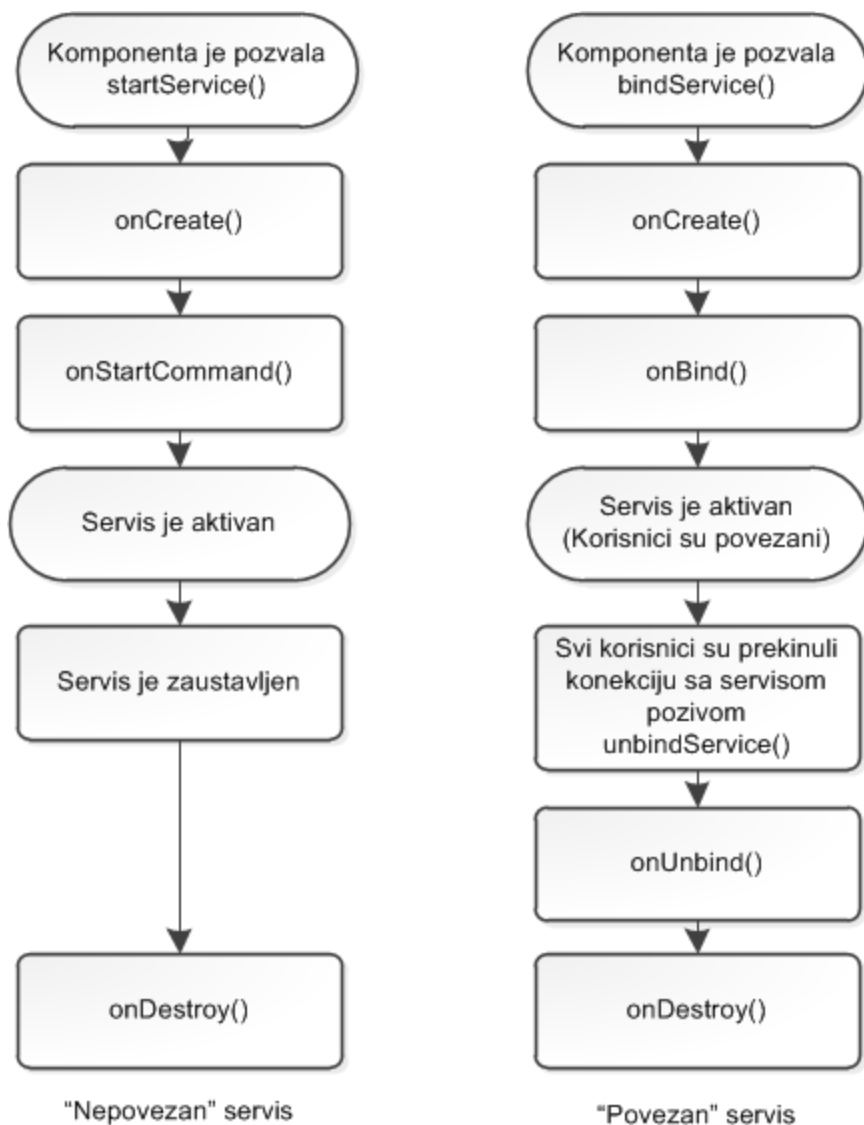
pokrenula prestane da bude aktivna. Obično pokrenut servis izvršava jednu operaciju i nema povratnu vrednost.

Povezan servis– Nakon poziva `bindService()` metode sa strane neke aplikativne komponente, omogućuje se komunikacija sa pokrenutim Android servisom. Na ovaj način realizuje se posluživač-korisnik programska sprega, koja omogućuje različitim komponentama da komuniciraju sa samim servisom. Na ovaj način omogućeno je slanje zahteva, prijem rezultata, pa čak i komunikacija sa drugim procesima u okviru operativnog sistema. Ovakav servis se izvršava sve dok postoji veza sa aplikativnom komponentom. Više različitih komponenti može biti povezano na isti servis, tada servis prestaje sa izvršavanjem, kada sve komponente prekinu interakciju sa njim pozivom funkcije `unbindService()`.

Servis se definiše nasleđivanjem ugrađene Android klase *Service* i implementiranjem metoda za definisanje ponašanja servisa tokom životnog ciklusa. Takođe neophodno je obezbediti rukovanje mehanizmom za komunikaciju sa drugim aplikativnim blokovima.

Na slici 3 prikazan je dijagram koji opisuje životni ciklus Android servis komponente za oba režima rada.

`onStartCommand()` - Sistem poziva ovu metodu kada neka aplikativna komponenta, npr. aktivnost, pozovme `startService()` metodu. Izvršavanjem ove metode servis se pokreće i može se beskonačno dugo izvršavati u pozadini, naravno servis može biti zaustavljen pozivom metode `stopSelf()` ili `stopService()`. Servis može biti nasilno zaustavljen od strane operativnog sistema u slučaju nedostatka memorije



Slika 3 Životni ciklus Android servisa

`onBind()` - Proziva se kada neka komponenta ili proces želi da se poveže sa servisom pozivom metode `bindService()`. U okviru ove metode, ukoliko se želi omogućiti povezivanje sa drugim komponentama

neophodno je definisati korisničku spregu i povratnu vrednost koji mora biti instanca IBinder klase.

IBinder je ugrađena korisnička sprega Android operativnog sistema, koja omogućuje komunikaciju visokih performansi, kao i deljenje podataka između različitih procesa.

onCreate() - Sistem poziva ovu metodu pri prvom startovanju servis komponente. U okviru ove metode se obično vrši inicijalizacija modula, bitnih za ispravno funkcionisanje samog servisa.

onDestroy - Poziva se za zaustavljanje servisa u trenutku kada više nije potreban. Neophodno je u okviru ove metode izvršiti denicijalizaciju i čišćenje nepotrebnih resursa, kako bi se sprečilo nepotrebno trošenje memorije (eng. *memory leak*).

Da bi korisničke aplikacije povezale sa servisom potrebno je definisati spregu kojom se odvija komunikacija između klijenta i servisa. Definisanje programske sprege se obavlja korišćenjem odgovarajućeg namenskog jezika (eng. *AIDL - Android Interface Definition Language*). Kao što je već rečeno, u Android operativnom sistemu jedan proces ne može jednostavno da pristupi memorijskom prostoru drugog procesa. AIDL sprega poseduje mogućnost raščlanjivanja složenih objekata i tipova podataka na primitivne tipove, koje operativni sistem može da deli između različitih procesa. Svaki složeni objekat koji se prenosi kroz IBinder spregu mora podržati protokol raščlanjivanja na primitivne tipove, i ponovnog sastavljanja u složen objekat (eng. *Parcelable protocol* - u daljem tekstu korišćen je izraz parselabilan protokol).

Važno je napomenuti da se za potrebe programske podrške ovog rešenja koristi servis komponenta, koja dozvoljava komunikaciju sa

drugim procesima. Na taj način se uspostavlja sprega između različitih slojeva aplikativnog dela programske podrške.

Aplikacija ili servis napisan u Java programskom jeziku prevodi se u bajt-kod. Ovakav kod izvršava se u Java virtuelnoj mašini. Za razliku od Java programskog jezika, drugi programski jezici prevode se u mašinski kod koji se direktno izvršava u procesoru. Ukoliko postoji potreba da aplikacija napisana u Java programskom jeziku, komunicira sa nekim modulom napisanom u C programskom jeziku, neophodno je koristiti JNI spregu (eng. *Java Native Interface*), kako bi se obezbedilo ispravno pozivanje C koda iz Java programa i prosleđivanje podataka u oba smera.

JNI sprega definiše način na koji komuniciraju moduli napisani u Java i C kodu. Sprega je nezavisna od platforme, omogućava korišćenje deljenih biblioteka i u mnogome doprinosi efikasnosti i performansama celokupnog sistema. JNI omogućava lak pristup razvojnom okruženju i razvojnoj podršci, koju omogućava operativni sistem Android platforme. Zbog toga JNI predstavlja prirodan način za povezivanje sistemskih i aplikativnih komponenti u operativnom sistemu Android platforme.

Operativni sistem Android platforme je prvi sloj programske podrške iznad fizičke arhitekture. Najvažnija uloga operativnog sistema u DTV prijemniku je da obezbedi okruženje za podršku obrade više zadataka (eng. *Multitasking*). Ograničeni resursi DTV prijemnika zahtevaju da memorija koju koristi operativni sistem bude minimalna, da

bi sa ostatkom memorije raspolagala aplikacija i da bi rešenje bilo što ekonomičnije.

Iznad operativnog sistema, iz perspektive programskog steka, nalazi se programska podrška DTV prijemnika, koja obezbeđuje usluge višeg nivoa koje su sastavni deo DTV funkcionalnosti i omogućava izvršavanje naprednih DTV aplikacija.

Programska podrška DTV prijemnika obezbeđuje odgovarajuću programsku spregu koja apstrahuje funkcionalnost DTV uređaja, fizičke arhitekture, kao i funkcija operativnog sistema, čime je omogućeno da proizvođači aplikativne DTV programske podrške ne moraju da poznaju specifičnosti fizičke arhitekture DTV prijemnika. Ovim je omogućeno da se aplikacije, koje koriste istu programsku podršku DTV prijemnika, mogu izvršavati na različitim fizičkim arhitekturama.

Niži nivo ili sloj apstrakcije fizičke arhitekture (u daljem tekstu CHAL) je sloj koji se nalazi između operativnog sistema i programske podrške DTV prijemnika. Raspolaze skupom funkcija koje apstrakuju funkcionalnosti vezane za korišćeni operativni sistem i obaveštava više programske slojeve o nastalim događajima.

Da bi se izvršilo prilagođenje aplikacije i programske podrške DTV prijemnika nekoj drugoj platformi dovoljno je izvršiti promenu samo ovog sloja.

CHAL (eng. *Comedia hardware adaptation layer*) sloj se sastoji od tri modula:

- TBOX (eng. *Tool Box*)
- TKEL (eng. *Thin Kernel Encapsulation Layer*)

- TDAL (eng. *Thin Driver Adaptation Layer*)

Cilj TBOX modula je da prikazuje kompletnu spregu (razmenu poruka, merenja, ulaznih i povratnih vrednosti funkcija itd.) između programske podrške u DTV prijemniku i adaptacionog sloja. Olakšava otklanjanje grešaka.

TKEL modul enkapsulira funkcije koje su specifične za korišćeni operativni sistem. (sinhronizacija, kritične sekcije, semafori, komunikacija, pravljenje programskih niti itd.).

Svaki funkcionalni nivo neophodan programskoj podršci TV prijemnika je obuhvaćen sa TDAL modulom, npr. podešavanje frekvencije, podešavanje filtera, kontrolisanje jačine zvuka, zaustavljanje videa, upravljanje formatom slike i zvuka itd.

Neki od TDAL modula koji su neophodni za osnovnu funkcionalnost ovog programskog rešenja su:

- TDAL_GFX - Ovaj modul se koristi za iscrtavanje grafičkih elemenata.
- TDAL_DISP - Njegova uloga je da obezbedi spregu između programske podrške TV prijemnika i video prikaza.
- TDAL_AV - Omogućava upravljanje reprodukcijom slike i zvuka (pokretanje, zaustavljanje).
- TDAL_TS - Predstavlja spregu za upravljanje prenosnim tokovima podataka (usmeravanje toka podataka od demodulatora do demultipleksera, prikazivanje i snimanje u istom trenutku itd.)

3. Koncept rešenja

Integracija digitalne televizije u Android platformu zahteva postojanje više slojeva programske podrške. Svaki od slojeva predstavlja prilagođen servis za druge slojeve. Potrebni elementi sistema i ciljna platforma su opisani u nastavku.

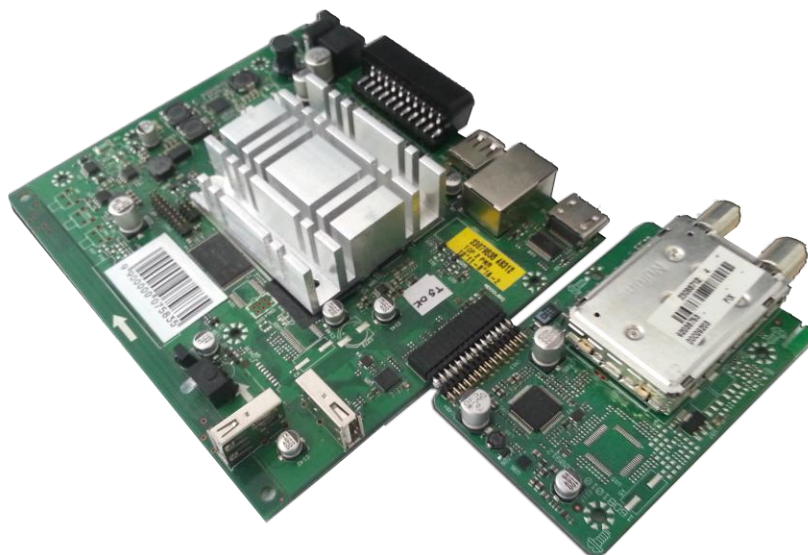
3.1 Opis ciljne platform

RK-2010 platforma raspolaže sledećim ključnim komponentama: DVB-T/T2 birač kanala, procesor ARM familije Marvell BG2 88DE3100, 1GB DDR3 operativne memorije, 8GB NAND memorije, 1 HDMI izlaz, 1 SCART izlaz, 1 S/PDIF izlaz, 1 Ethernet port i 3 USB priključaka. Upravo zbog ovih karatkeristika ova platforma je izabrana kao ciljna platforma jednog prijemnika digitalnog TV signala zasnovanog na Android platformi.

Platforma poseduje aplikativnu programsku spregu (eng. *Application Programming Interface*) koja obezbeđuje funkcije višeg

nivoa za rukovanje sastavnim delovima fizičke arhitekture. Pomenuta sprega je iskorišćena i značajno je olakšala pronalaženje rešenja zadanog problema.

Izgled ciljne platforme dat je na slici 4.

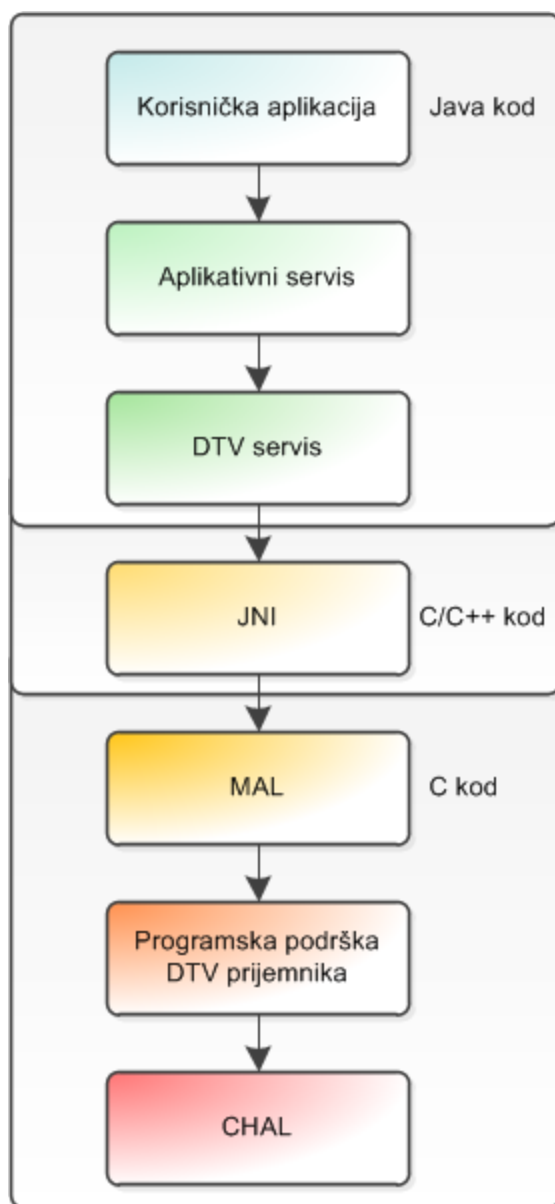


Slika 4 Izgled ciljne platforme

3.2 Ključne komponente sistema

Cilj ovog rešenja jeste da krajnjem korisniku TV aplikacije na prijemniku televizijskog signala zasnovanog na Android platformi, omogući jednostavan pristup velikom skupu različitog multimedijalnog sadržaja, rukovanje i njegovu reprodukciju.

Akcent se stavlja na objedinjavanje svih tipova podataka koje nudi digitalna televizija i uređaji iz kućne mreže, zajedno sa širokim mogućnostima Android platforme.



Slika 5 Prikaz organizacije modula programskog rešenja

Jedan od glavnih zahteva je omogućiti preglednu grafičku spregu, koja omogućuje izlistavanje dostupnog multimedijalnog sadržaja i njegovu reprodukciju, dobre performance i vreme odziva.

Prilikom realizacije potrebno je iskoristiti ugrađenu korisničku spregu i programsku podršku, koju nudi Android platforma, i imajući u vidu ciljnu platformu za koju se ovakav sistem razvija prilagoditi i izmeniti određene funkcionalnosti.

Digitalni televizijski prijemnik (eng. *STB – Set Top Box*), je uređaj koji obično sadrži birač kanala i povezuje se na televizijski uređaj. Vrší obradu nad signalom i prikazuje ga na televizijskom ekranu. Koristi se u zemaljskim, kablovskim i satelitskim sistemima, kako bi pretvorio signal iz prenosnog oblika u oblik pogodan za prikaz na ekranu i određenom obradom poboljšao kvaliteta signala.

Ovo programsko rešenje zasniva se na višeslojnoj programskoj podršci, gde svaki sloj u hijerarhiji unosi dodatni nivo apstrakcije podataka. Ovakvom organizacijom omogućuje se lak prelaz na neku drugu platformu, operativni sistem ili realizaciju nove grafičke sprege ka krajnjem korisniku bez potrebe menjanja ostalih modula u sistemu.

Na slici 5 prikazana je organizacija modula u programskoj podršci ovog programskog rešenja.

Svaki od osnovnih modula sastoji se iz nekoliko podmodula, čineći tako složenu programsku strukturu. Podmoduli približnije opisuju funkcionalnost slojeva programske podrške.

Aplikacija – Predstavlja Android aplikaciju pisanu u Java programskom jeziku. Realizuje grafičku spregu, što predstavlja vezu ka krajnjem korisniku. Korisniku pruža jednostavnu spregu ka podržanim

multimedijalnim sadržajima i omogućuje njegovu kontrolu i reprodukciju. Obrađuje akcije i komande koje zadaje krajnji korisnik i na osnovu njih inicira komunikaciju sa različitim modulima programske podrške. U direktnoj je vezi sa aplikativnim servisom.

Aplikativni servis – Predstavlja Android servis napisan u Java programskom jeziku. Zadužen je za pripremu i prikupljanje podataka o dostupnom multimedijalnom sadržaju, koje kasnije koristi Android korisnička aplikacija. Usko je povezan sa radom aplikacije i aktivan je dok god je aktivna i aplikacija.

U zavisnosti od tipa dostupnog sadržaja inicira komunikaciju sa modulima nižeg nivoa programskog steka, to može biti DTV servis, ili neke od izvornih biblioteka za DLNA podršku i sl.

DTV servis – Android servis napisan u Java programskom jeziku, sa dodatkom da poseduje mogućnost komuniciranja sa JNI slojem. Zadužen je za pripremu i dobavljanje dostupnog televizijskog sadržaja iz DTV programske podrške, kojeg kasnije koristi aplikativni servis i korisnička aplikacija.

Postojanje DTV servis modula omogućuje da se aplikativni moduli odvoje od programske podrške televizijskog prijemnika.

Na ovaj način DTV programska podrška se može iskoristiti za razvoj neke druge TV aplikacije ili aplikativnog sloja.

JNI sloj – Omogućuje vezu između Java servisa i aplikativnog sloja sa slojevima nižeg nivoa, tj. televizijske programske podrške (eng. *middleware*).

MAL sloj (eng. *Middleware Abstraction Layer*) – Omogućuje dvosmernu komunikaciju viših aplikativnih slojeva programskog steka sa televizijskom programskom podrškom, koja predstavlja najvažniji modul u radu digitalnog televizijskog prijemnika.

MAL sloj predstavlja apstrakciju modula programske podrške televizijskog prijemnika.

Ovakvom organizacijom se postiže da ukoliko je potrebno iskoristiti aplikativni sloj na nekoj drugoj platformi koja koristi neku drugu DTV programsku podršku, potrebno je izmeniti samo MAL sloj.

Televizijska programska podrška – Predstavlja modul koji sadrži DTV funkcionalnost (pretraga televizijskog sadržaja, liste televizijskih servisa, reprodukcija, itd.).

Televizijsku programsku podršku za digitalnu televiziju je moguće izvršavati na različitim platformama jer poseduje prilagodni sloj, odnosno sloj apstrakcije fizičke arhitekture, CHAL.

Programska podrška za digitalnu televiziju nudi punu DTV funkcionalnost, ali od izabrane platforme zavisi da li je moguće u potpunosti iskoristiti njene mogućnosti.

CHAL sloj – Predstavlja programsku spregu ka modulima fizičke arhitekture. Omogućuje komunikaciju televizijske programske podrške sa modulima fizičke arhitekture STB uređaja (birač kanala, demultiplekseri, dekoderi, itd.).

Zbog višeslojne programske podrške neophodno je ostvariti odgovarajuće komunikacione sprege, kako bi se omogućila ispravna komunikacija između mnogobrojnih modula programske podrške.

Isto tako potrebno je obezbediti ispravan rad svakog od modula i njegovu funkcionalnost ispitati odgovarajućim nizom ispitnih slučajeva.

Za realizaciju, ranije navedenih zahteva korišćene su različite tehnike prilagođavanja ciljnoj platform, koje će kasnije biti detaljno opisane.

4. Programsko rešenje

Ovo rešenje zasniva se na višeslojnoj programskog podršci, gde svaki sloj u hijerarhiji unosi dodatni nivo apstrakcije podataka. Ovakvom organizacijom omogućuje se lak prelaz na neku drugu platformu, operativni sistem ili realizaciju nove grafičke sprege ka krajnjem korisniku bez potrebe menjanja ostalih modula u sistemu.

4.1 Korisnička aplikacija

Korisnička aplikacija smeštena je na vrhu programske hijerarhije.

Predstavlja instancu najviše apstrakcije programske podrške. Osnovni zahtev koji se postavlja pred ovaj sloj programske podrške jeste da omogući jednostavan pregled dostupnog multimedijalnog sadržaja i rukovanje njime. Pored toga aplikacija treba da očuva dobre performanse i kompatibilnost sa mnogobrojnim različitim verzijama operativnog sistema Android platforme.

Kako bi se u što većoj meri omogućile dobre performanse i smanjilo vreme odziva korisničke aplikacije, aplikacija treba da je, što je više moguće, nezavisna od ostatka programske podrške.

Na osnovu zadatih komandi od strane krajnjeg korisnika, korisnička aplikacija inicira komunikaciju sa nižim modulima programske podrške, radi pribavljanja i pripreme podataka, koje treba prikazati na ekranu.

Osnovna uloga korisničke aplikacije jeste iscertavanje i prikaz dostupnog sadržaja na ekranu i iniciranje njegove reprodukcije ili nekih drugih dostupnih akcija, pozivanjem odgovarajućih metoda iz nižih slojeva programske podrške.

Skup dostupnih akcija je specifičan za određeni tip multimedijalnog sadržaja, ali se sve oslanjaju na iste module aplikativnog servisa, što predstavlja prednost višeslojne organizacije programske podrške.

Krajnjem korisniku se daje mogućnost razvrstavanja i izdvajanja multimedijalnog sadržaja po tipu i formatu.

Aplikacija razlikuje sledeće tipove multimedijalnog sadržaja i omogućuje njihovo izdvajanje iz skupa svih dostupnih sadržaja (eng. *content filtering*):

- DVB T sadržaj
- DVB C sadržaj
- DVB S sadržaj
- Data sadržaj
- Radio sadržaj
- IP sadržaj

- Android aplikacije
- Android namenske grafičke kontrole
- DLNA poslužioi
- USB uređaji
- Video, audio datoteke i slike
- APK datoteke (eng. *Android Package*)
- PVR sadržaj i druge

Dostupan multimedijalni sadržaj raspoređuje se u tri nezavisne liste:

- Poslednje korišćen sadržaj (eng. *Recently Accessed List*)
- Omiljeni sadržaj (eng. *Favorite List*)
- Sav dosupan sadržaj iz određene kategorije (eng. *All List*)

Bitno je napomenuti da se prilikom izdvajanja željenog tipa sadržaja iz skupa svih dostupnih sadržaja (u daljem tekstu filtriranje), filtriraju sve tri prikazane liste podataka.

Korisnička aplikacija nudi dodatne mogućnosti, korišćenjem jednostavne grafičke sprege, koje olakšavaju rukovanje listama dostupnog sadržajima, a to su: mogućnost ručnog dodavanja sadržaja u listu omiljenih sadržaja ili brisanje elemenata iz lista omiljenih i prethodno pristupanih sadržaja.

Takođe veoma važna opcija koju nudi ovo programsko rešenje jeste sprečavanje neželjenog pristupa nekom od dostupnog sadržaja, postavljanjem šifre za autentikaciju korisnika. Na taj način korisnik

može, ukoliko postoji potreba za tim, da zabrani pristup nekom sadržaju, drugim korisnicima.

Korisnička Android aplikacija realizovana je u Java programskom jeziku, korišćenjem osnovnih Android grafičkih komponenti, koje su prilagođene zahtevima moderne televizijske aplikacije, koja se zasniva na tzv. „10-foot“ grafičkoj korisničkoj sprezi.

Način realizacije korisničke sprege opisan je u daljem tekstu.

Grafička sprega koja se predstavlja krajnjem korisniku TV aplikacije realizovana je unutar jedinstvenog dijaloga (prozora). Osnovna Android dijalog komponenta je izmenjena tako da prekriva čitav ekran. Kada je Dialog komponenta aktivna, ostatak korisničke aplikacije gubi fokus, i korisnička sprega za prikaz dostupnog multimedijalnog sadržaja je spremna za prikupljanje akcija od strane korisnika.

Izgled i raspored grafičkih komponenti unutar Android dijaloga organizovan je tako da se zadovolje zahtevi „10-foot“ korisničke sprege, što znači da su grafički elementi osmišljeni i realizovani tako da budu vidljivi i čitljivi sa veće udaljenosti. Takođe potrebno je izbeći moguće probleme koje izaziva „overscan“ efekat na nekim ekranima i TV uređajima.

Unutar dijaloga postavljena su tri objekta klase GridView, koji prikazuju na ekranu elemente lista prethodno korišćenog sadržaja, omiljenih sadržaja i svih dostupnih sadržaja.

Za svaki GridView element povezan je odgovarajuća BaseAdapter klasa, koja određuje koji element je potrebno iscrtati na ekranu.

Osnovni moduli koji su zaduženi za rukovanje multimedijalnim sadržajem u okviru korisničke aplikacije su:

- Content List Handler
- All List Handler
- Recently Handler
- Favorite Handler
- Grid View Helper
- Grid View Scroller

Content List Handler modul predstavlja najvažniji modul za rukovanje multimedijalnim sadržajem unutar korisničke aplikacije, jer implementira odgovarajuću logiku koja kontroliše rad ostalih navedenih modula.

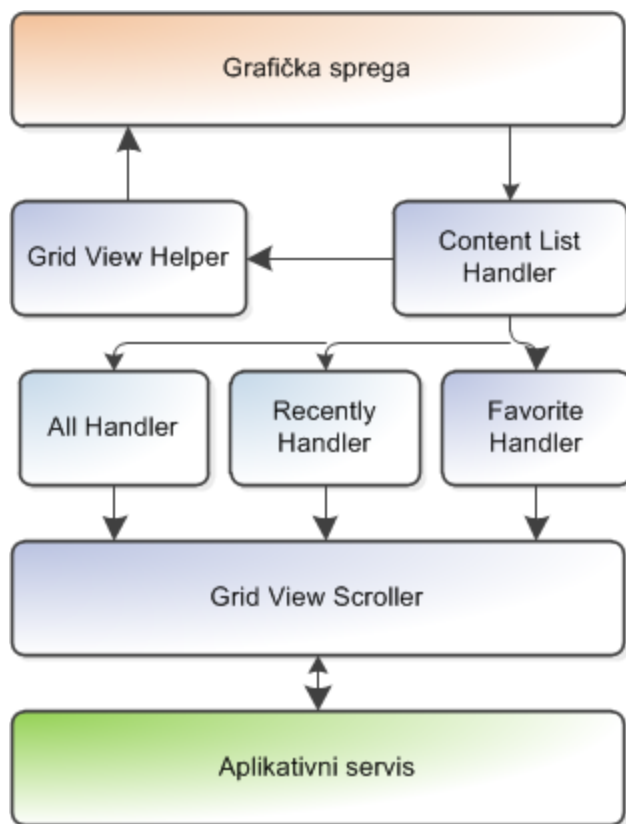
Content List Handler modul takođe održava komunikaciju sa nižim slojevima programske podrške.

Osim što kontroliše rad ostalih aplikativnih modula, Content List Handler inicira filtriranje sadržaja, zaključavanje sadržaja, reprodukciju, rukovanje dostupnim listama i inicira njihovo iscrtavanje na ekranu.

Slika 6 prikazuje organizaciju modula unutar korisničke aplikacije.

All List Handler, Recently Handler i Favorite Handler moduli rukuju sadržajem koji je raspoređen u listi svih sadržaja, listi prethodno korišćenog i listi omiljenog sadržaja, respektivno, tj. kontrolišu rad BaseAdapter objekata povezanih sa odgovarajućim GridView elementima, prikazanim na ekranu u okviru grafičke korisničke sprege.

Grid View Helper modul zadužen je za pripremu sadržaja za iscrtavanje na ekran.



Slika 6 Organizacija modula unutar korisničke aplikacije

Grafički elementi koji se iscrtavaju na ekran pomoću GridView komponente, realizovani su kombinacijom različitih ugrađenih Android grafičkih komponenti, a njihov izgled definisan je pomoću XML (eng. *Extensible Markup Language*) programskog jezika.

Nakon pripreme podataka u Grid View Helper modulu, grafički elementi se popunjavaju odgovarajućim sadržajem.

Grid View Scroller modul, zadužen je za dobavljanje podataka sa strane aplikativnog servisa i rukovanje navigacijom unutar grafičke sprege.

Pošto korisnička aplikacija ne treba da vrši složene računске operacije, zbog smanjenja vreme odziva i poboljšanja performansi

sistema, prikupljanje podataka za prikaz vrši aplikativni servis napisan u Java programskom jeziku, o kojem će kasnije biti više reči.

Kako je jedan od zahteva da aplikacija na jednak način tretira sve tipove multimedijalnih podataka, potrebno je uvesti dodatnu apstrakciju koja sve tipove podataka i njihove osobine objedinjuje u jednu klasu. Za te potrebe uvodi se klasa Content, koja predstavlja instancu najveće apstrakcije.

U cilju što veće optimizacije korisničke aplikacije i što boljeg iskorišćenja procesorskog vremena, liste koje prikazuju multimedijalni sadržaj uvek imaju konstantan broj elemenata (šest, dvanaest ili osamnaest), iako npr. broj dostupnog sadržaja može biti mnogo veći od toga. Na taj način se u memoriji čuvaju samo trenutno vidljivi objekti na ekranu, smanjuje se zauzetost resursa čime se dobija na ekonomičnosti programskog rešenja.

Grid View Scroller modul vodi računa o tome koji će se elementi iscrtavati na ekran, u trenucima kada korisnik zadaje komande za navigaciju uz pomoć daljinskog upravljača, tako da korisnik ne gubi osećaj kontinualnog pomeranja kroz sadržaj prikazan na ekranu.

Komunikacija korisničke aplikacije i aplikativnog servisa započinje prilikom pokretanja korisničke aplikacije, gde se aplikativni servis pokreće pozivom:

```
Intent i = new Intent("com.applicationService.PROXY_SERVICE");  
MainActivity.getApplicationContext().startService(i);
```

Nakon pokretanja aplikativnog servisa neophodno je uspostaviti vezu sa njim, kako bi se omogućila komunikacija i razmena podataka:

```
Intent remoteServiceIntent = new  
Intent("com.applicationService.PROXY_SERVICE");  
BinderServiceConnection connection = new BinderServiceConnection();  
boolean isBound = activity.bindService(remoteServiceIntent,  
connection, Context.BIND_AUTO_CREATE);
```

Prilikom povezivanja konstanta `Context.Bind_AUTO_CREATE`, određuje da će pokrenuti aplikativni servis biti aktivan za vreme trajanja veze sa korisničkom aplikacijom.

Nakon uspešnog povezivanja sa servisom, korisnička sprega koju implementira aplikativni servis, dostupna je za korišćenje Java korisničkoj aplikaciji i aplikativni moduli mogu korisniti medote implementirane unutar aplikativnog servisa.

Tok podataka unutar korisničke aplikacije i odvijanje komunikacije između aplikativnih modula, od inicijalizacije do prikaza multimedijalnog sadržaja na ekranu opisan je u daljem tekstu.

Prilikom prvog iniciranja prikaza korisničke sprege za prikaz dostupnog multimedijalnog sadržaja, inicijalizuje se `Content List Handler` modul, tom prilikom inicijalizuju se i neophodne grafičke komponente, dostupni filteri i moduli za rukovanje listama.

Zatim se iz aplikativnog servisa preuzima informacija o broju dostupnih multimedijalnih sadržaja koje se prosleđuju `All`, `Favorite` i `Recently Handler` modulima.

Moduli za rukovanje listama na osnovu informacija o broju dostupnih sadržaja, putem Grid View Scroller modula učitavaju elemente iz aplikativnog servisa na osnovu jedinstvenog identifikatora.

Nakon učitavanja elemenata u odgovarajuće liste, Grid View Scroller modul osvežava BaseAdapter objekte povezane sa grafičkim komponentama. Zatim uz pomoć Grid View Helper modula, koji u zavisnosti od tipa sadržaja preuzima informacije od interesa i popunjava grafičke elemente, nakon čega ih iscrtava na ekran.

Postupak inicijalizacije se obavlja samo jedan put u toku rada korisničke aplikacije, dok se prilikom navigacije kroz liste učitavaju samo novi elementi, koji do tad nisu bili vidljivi na ekranu.

Na ovaj način postiže se značajna ušteda u potrošnji procesorskog vremena, i održava se vreme odziva aplikacije na zahtevanom nivou.

4.2 Aplikativni servis

Aplikativni servis modul realizovan je kao Android servis u Java programskom jeziku. Njegova osnovna uloga u okviru ovog programskog rešenja jeste da doprinese poboljšanju performansi i smanjenju vremena odziva korisničke aplikacije.

Takođe aplikativni servis omogućava, da ukoliko postoji potreba, razvoj neke druge korisničke aplikacije bez potrebe za izmenama drugih modula u okviru programske podrške.

Pored višestrukih funkcionalnosti koje su realizovane unutar aplikativnog servisa, realizovani su i moduli koji su zaduženi za prikupljanje, pripremu i obradu različitog multimedijalnog sadržaja i njegovu propagaciju ka korisničkoj aplikaciji.

Definisanjem AIDL programske sprege, implementirane metode za rukovanje multimedijalnim sadržajem dostupne su u korisničkoj aplikaciji, na osnovu kojih se i zasniva njihova komunikacija.

Glavni modul koji rukuje dostupnim multimedijalnim sadržajem i podmodulima unutar aplikativnog servisa koji vrše njihovu obradu podataka, naziva se Content List Control modul.

Dostupni multimedijalni sadržaj razvrstava se u tri nezavisne liste, kako bi se zadovoljile potrebe korisničke aplikacije.

Sve tri liste se konstantno ažuriraju, kako bi korisnik u svakom trenutku imao uvid u sav dostupan multimedijalni sadržaj.

Važno je napomenuti da se sadržaj liste omiljenih sadržaja i liste poslednje korišćenih sadržaja uz posredstvo namenskih modula u okviru aplikativnog servisa, čuvaju u lokalnoj bazi podataka, kako bi se nakon gašenja korisničke aplikacije zadržalo ispravno stanje i sadržaj pomenutih lista.

Filtriranje multimedijalog sadržaja omogućava Content Manager modul, koji vodi računa o tipu trenutno aktivnog filtera i njegovog sadržaja.

Svaki od podržanih tipova poseduje odgovarajući filter modul, koji je obrađuje podatke vezane samo za taj tip sadržaja.

Svaki od modula koji opisuje zaseban tip filtera, tj. skup multimedijalnih sadržaja istog tipa, implementira programsku spregu IContentFilter, koja definiše metode kojima je moguće pristupiti informacijama o željenom multimedijalnom sadržaju i način kojim se rukuje njime. Neke od najznačajnijih metoda koje se koriste u okviru ovog programskog rešenja su:

Metoda koja omogućava korisničkoj aplikaciji da pristupi informacijama o nekom određenom multimedijalnom sadržaju.

```
Content getContent(int index);
```

Metoda koja daje informaciju o broju dostupnih multimedijalnih sadržaja u okviru određenog tipa filtera.

```
int getContentListSize();
```

Metoda koja daje informaciju o broju omiljenih multimedijalnih sadržaja u okviru određenog tipa filtera.

```
int getFavoriteListSize();
```

Metoda koja daje informaciju o broju prethodno korišćenih multimedijalnih sadržaja u okviru određenog tipa filtera.

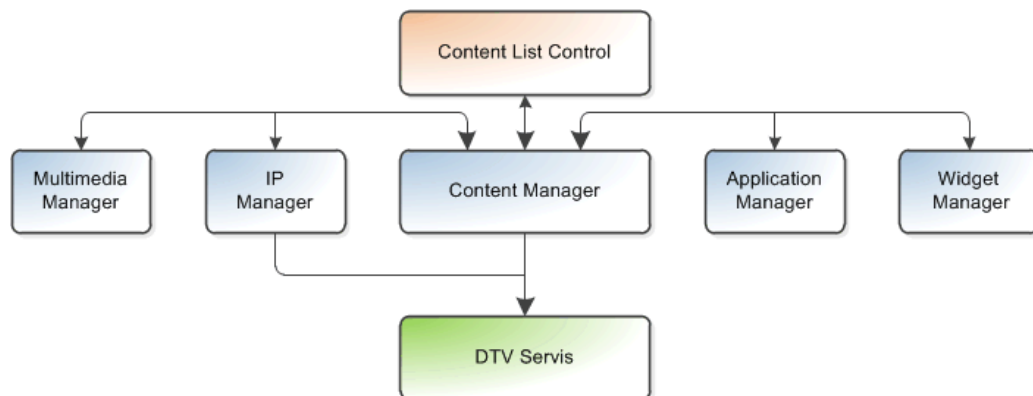
```
int getRecentlyListSize();
```

Metoda koja inicira reprodukciju željenog multimedijalnog sadržaja i ujedno ažurira sadržaj liste prethodno korištenih sadržaja.

```
void goContent(Content content);
```

U zavisnosti od tipa filtera i tipa multimedijalnog sadržaja sam način implementacije ovih metoda se razlikuje i to će kasnije biti detaljnije objašnjeno.

Na sledećoj slici prikazana je organizacija modula unutar aplikativnog servisa, zaduženih za rukovanje multimedijalnim sadržajem.



Slika 7 Organizacija modula u okviru aplikativnog servisa

Kada korisnička aplikacija zatraži informacije o nekom multimedijalnom sadržaju, Content List Control modul, putem Content Manager modula preuzima informacije o trenutno aktivnom filteru, proziva metode aktivnog filtera i prosleđuje povratne vrednosti ka korisničkoj aplikaciji.

Rukovanje određenim tipovima podataka opisan je u daljem tekstu.

4.2.1 Android aplikacije

Kako se ovo programsko rešenje zasniva na Android platformi, jedan od osnovnih zahteva koji se nameće je da se korisniku omogući pristup i korišćenje svih dostupnih instaliranih aplikacija koje se nalaze na uređaju.

Na ovaj način se skupu standardnih televizijskih opcija i mogućnosti dodaje veoma važno proširenje, koje u mnogome doprinosi

boljem korisničkom doživljaju prilikom korišćenja digitalnog televizijskog prijemnika.

Za rukovanje instaliranim Android aplikacijama unutar aplikativnog servisa zadužen je Application Manager modul. Prilikom njegove inicijalizacije, pozivanjem sistemskih funkcija operativnog sistema Android platforme, dobavljaju se informacije o dostupnim aplikacijama.

```
List<ApplicationInfo> apps = ApplicationService.getInstance().  
getPackageManager().getInstalledApplications(0);
```

ApplicationInfo objekat sadrži osnovne informacije o Android aplikaciji, koje su obično definisane u Manifest.xml datoteci.

Kako bi se dobavljene informacije prilagodile definisanoj korisničkoj sprezi i načinu rada korisničke aplikacije, potrebno je uvesti dodatni nivo apstrakcije.

Iz ovog razloga definišu se klase AppItem i ApplicationContent.

AppItem klasa sadrži polja koja čuvaju informacije o imenu aplikacije, nazivu paketa i tipu instalirane aplikacije.

ApplicationContent klasa nasleđuje klasu Content, koja apstahuje informacije o različitim multimedijalnim sadržajima, i implementira Parcelable programsku spregu, kako bi se njen objekat mogao deliti između različitih procesa pomoću IBinder programske sprege.

ApplicationContent klasa sadrži informacije o imenu i nazivu paketa aplikacije, koji se preuzima iz objekata klase AppItem, redni broj aplikacije i ikonicu.

Kada korisnička aplikacija želi da prikaže dostupne Android aplikacije, proziva metodu `getContent(int index)` implementiranu u Content List Control modulu aplikativnog servisa. Content List Control modul proverava tip aktivnog filtera u Content Manager modulu, i zatim se informacije o aplikacijama preuzimaju iz Application Manager modula.

Ukoliko korisnik želi da pokrene neku od dostupnih aplikacija, na definisanu akciju daljinskog upravljača, korisnička aplikacija proziva `goContent(Content content)` metodu aplikativnog servisa, definisanu u ContentFilterApps modulu koji implementira `IContentFilter` korisničku spregu.

Željena aplikacija se pokreće pozivom ugrađenih Android metoda i ugrađene klase `Intent`.

```
Intent launchIntent = ApplicationService.getInstance().  
getPackageManager().getLaunchIntentForPackage  
(content.getPackageName());  
ApplicationService.getInstance().startActivity(launchIntent);
```

Za vreme rada neke druge Android aplikacije, korisnička TV aplikacija nije vidljiva na ekranu, ali se ona i dalje izvršava u pozadini.

Uvođenjem ovog proširenja, prijemnik TV signala pruža mnogo veći ugođaj u odnosu na standardne prijemnike, gde korisnik može da proširi svoj doživljaj korišćenjem različitih video igrica, socijalnih mreža ili aplikacija edukativnog sadržaja.

4.2.2 Android namenske grafičke korisničke kontrole

Android namenske grafičke korisničke kontrole su uprošćene Android aplikacije, koje mogu biti ugrađene u druge Android aplikacije. Njihova glavna osobina jeste da se njihov sadržaj može periodično osvežavati bez neophodne interakcijom sa korisnikom.

Namenske grafičke korisničke kontrole se prikazuju pomoću ugrađenog App Widget Provider modula. Komponenta operativnog sistema Android platforme koja prikazuje grafički sadržaj grafičke kontrole aplikacije i omogućuje interakciju sa korisnikom naziva se App Widget Host.

Rukovanje namenskim grafičkim kontrolama unutar aplikativnog servisa je slično kao i rukovanje običnim Android aplikacijama.

Za prikupljanje informacija o dostupnim namenskim grafičkim kontrolama zadužen je Widget Manager modul. Informacije se dobavljaju pozivanjem ugrađenih Android sistemskih metoda.

```
List<AppWidgetProviderInfo> widgetList = AppWidgetManager.  
getInstance(context).getInstalledProviders();
```

AppWidgetProviderInfo sadrži informacije o izgledu, definisanom periodu osvežavanja i druge važne informacije koje opisuju widget aplikaciju.

Kako pribavljene informacije o dostupnim grafičkim kontrolama treba prilagoditi korisničkoj sprezi i definisanom načinu komunikacije, potrebno je uvesti apstrakciju ovih podataka.

Kako bi se to omogućilo uvode se klase WidgetItem i WidgetContent.

WidgetItem klasa sadrži informacije o imenu, nazivu paketa i nazivu glavne klase Android namenske grafičke kontrole.

WidgetContent klasa nasleđuje klasu Content i ujedno implementira Parcelable protokol.

WidgetContent klasa proširuje sadržaj WidgetItem klase novim poljima, od kojih su najvažnija: redni broj i ikonica koja će biti prikaza u grafičkoj sprezi korisničke aplikacije.

Pošto namenske grafičke kontrole mogu biti ugrađene i prikazane u nekoj drugoj aplikaciji, cilj je omogućiti korisniku da ih prikaže u okviru korisničke TV aplikacije.

Da bi se to omogućilo potrebno je formirati dodatnu grafičku ravan u TV aplikaciji, koja će se nalaziti iznad video ravni, kako bi namenske grafičke kontrole bile vidljive na ekranu u svakom trenutku.

Specifičnost vezana za Android namenske grafičke kontrole jeste da se njihovo pokretanje mora vršiti iz istog Android procesa u kojem su i prikazane, što znači da aplikativni servis ne poseduje mogućnost pokretanja grafičkih kontrola.

Kako bi se prevazišao taj problem, u okviru korisničke aplikacije realizovan je Widget Manager modul koji je zadužen za rukovanje radom namenskih grafičkih kontrola, dok je aplikativni servis zadužen za njihovo prikupljanje kao i za sve ostale dostupne multimedijalne sadržaje.

Kako bi se poboljšao korisnički doživljaj realizovan je Drag Controller modul koji omogućuje pomeranje grafičkih kontrola po ekranu pomoću „drag and drop“ komandi. Još jedan važan modul koji rukuje namenskim grafičkim kontrolama je i Widget Arranging Handler modul, koji realizuje logiku koja ih raspoređuje po ekranu.

Na ovaj način se sprečava preklapanje sadržaja različitih grafičkih kontrola na ekranu.

Kako bi se omogućio rad Drag Controller modula, prikaz grafičke kontrole se mora prilagoditi korisničkoj sprezi Android platforme.

Element koji se prikazuje na ekranu u zasebnoj grafičkoj ravni sastoji se iz više grafičkih slojeva.

Prvi sloj predstavlja sam sadržaj grafičke kontrole tj. App Widget Host komponentu. Kako ovaj element ne poseduje mogućnost prepoznavanja „drag and drop“ komandi, iznad ovog sloja postavlja se dodatni LinearLayout element koji osluškuje i prihvata korisničke komande.

Na taj način ukoliko je reč o „drag and drop“ komandi, kojom korisnik želi da pomeri grafičku kontrolu na neku drugu poziciju na ekranu, ona se pomoću LinearLayout elementa prosleđuje Drag Controller modulu na potrebnu obradu.

Zbog nemogućnosti operativnog sistema Android platforme da dinamički pomera grafičke elemente po ekranu, kao što je moguće u nekim drugim operativnim sistemima, Drag Controller je realizovan tako da omogućí ovu funkcionalnost.

Dodatna grafička ravan namenjena za prikaz namenskih grafičkih kontrola sadrži prilagođenu grafičku komponentu DragLayer.

DragLayer komponenta omogućuje prepoznavanje X i Y pozicije na ekranu „drag and drop“ komande i pomeranje Bitmap objekta po svojoj površini.

Kada Drag Controller propozna zadatu „drag and drop“ komadu, uz pomoć sistemskim metoda preuzima sadržaj grafičke kontrole i od

njega stvara privremeni Bitmap objekat na trenutnoj poziciji grafičke kontrole, koji će se pomerati po ekranu.

Android namenska grafička kontrola se privremeno sklanja sa ekrana, i u trenutku prestanka „drag and drop“ komande, grafička kontrola se postavlja na zadatu poziciju, a Bitmap objekat se briše sa ekrana i iz memorije.

Na taj način se stvara iluzija i efekat da se grafičke kontrole stvarno pomeraju po ekranu, što je veoma bitno sa stanovišta korisničkog doživljaja.

Ukoliko korisnik želi da ostvari interakciju sa grafičkim kontrolama, komanda se prosleđuje nižem grafičkom sloju, tj. App Widget Host komponenti, koja dalje obrađuje komandu u zavisnosti od realizacije i funkcionalnosti same grafičke kontrole.

Widget Arranging Handler modul čuva informacije o aktivnim grafičkim kontrolama na ekranu, njihove trenutne pozicije, veličine, itd.

Ove informacije se prilikom izlaska iz korisničke aplikacije smeštaju u binarnu datoteku, kako bi se zadržalo prethodno stanje grafičkih kontrola, prilikom ponovnog ulaska u korisničku aplikaciju.

Slika 14 prikazuje izgled korisničke TV aplikacije sa aktivnim Android namenskim grafičkim kontrolama na ekranu.

4.2.3 DLNA i lokalni sistem podataka

U ovom programskom rešenju korisniku je omogućeno rukovanje različitim video, audio zapisima i slikama, koji mogu biti dostupni na nekom od DLNA poslužioca u okviru kućne mreže, USB uređaju ili lokalnom sistemu podataka.

DLNA je neprofitabilna organizacija koja je definisala smernice za razmenu digitalnih multimedijalnih sadržaja između korisničkih uređaja kao što su računari, mobilni telefoni, kamere i ostalih multimedijalni uređaji.

DLNA koristi UPnP (eng. *Univesal Plug and Play*) protokol za pretragu, kontrolu i upravljanje multimedijalnim sadržajem. Uređaje koje DLNA podržava i mehanizam pristupa dostupnom sadržaju preko mreže definiše UPnP protokol [7].

UPnP je skup mrežnih protokola koji omogućavaju umreženim uređajima kao što su računari, štampači, bežične pristupne tačke, mobilni uređaji da otkriju prisustvo drugih uređaja u mreži i da uspostave mrežne usluge za deljenje podataka i komunikaciju.

UPnP tehnologija je nezavisna od operativnog sistema i programskog jezika i izgrađena je na internet baziranim tehnologijama kao što su IP (eng. *Internet Protocol*), TCP (eng. *Transmition Control Protocol*), UDP (eng. *User Datagram Protocol*), HTTP (eng. *Hypertext Transfer Protocol*) i XML.

DLNA uređaji su podeljeni u tri klase: kućne mrežne uređaje, ručne mobilne uređaje i kućne infrastrukturne uređaje.

Kućni mrežni uređaji se dele na :

- Digital Media Server (DMS) čuva sadržaj i čini ga dostupnim u mreži digital media player-u i digital media renderer-u
- Digital Media Player (DMP) nalazi sadržaj na DMS-u i omogućuje reprodukciju i prikazivanje tog sadržaja

- Digital Media Renderer (DMR) reprodukuje sadržaj primljen od strane digital media controller-a
- Digital Media Controller (DMC) pronalazi sadržaj na DMS-u i reprodukuje ga na DMR-u
- Digital Media Printer (DMP) omogućuje štampanje sadržaja koga mogu koristiti DMP i DMC

Programska podrška u okviru ovog rešenja implementira funkcionalnost DMS, DMP i DMR uređaja. To znači da se prijemnik digitalnog televizijskog signala u okviru kućne mreže može koristiti kao jedan od tri ponuđena tipa uređaja [8].

Na nivou aplikativnog servisa, prikupljanjem i rukovanjem multimedijalnim sadržajem sa DLNA posluživača ili nekog dostupnog sistema podataka rukuje Multimedia Manager modul.

Zbog složenosti operacija koje se izvršavaju prilikom komunikacije sa DLNA uređajima i sistemom podataka, programska podrška koja rukuje komunikacijom realizuje se u C programskom jeziku. Na ovaj način postiže se kompatibilnost sa različitim DLNA uređajima i smanjuje se vreme odziva [9].

Kako je realizacija modula za komunikaciju smeštena u izvornim (eng. *native*) bibliotekama, aplikativni servis im mora pristupati putem JNI sloja.

Prilikom inicijalizacije Multimedia Manager modula, inicijalizuju se izvorne biblioteke za rukovanje sistemom podataka i biblioteke za komunikaciju sa DLNA uređajima, i podešava se režim rada (DMS, DMP ili DMR).

U okviru ovog rada najviše vremena će biti posvećeno DMP režimu, iz razloga što je moguće iskoristiti prethodno definisane programske sprege i module.

U slučaju DMS i DMR režima rada, programska podrška i korisnička aplikacija koriste druge programske module.

Nakon inicijalizacije prikupljaju se informacije o dostupnim medijumima na mreži i priključenim USB uređajima.

Kako bi se doprinelo većoj preglednosti, zbog velikog broja različitog sadržaja, multimedijalni sadržaj koji je dostupan na DLNA poslužiocima i sistemima podataka objedinjuje se uz pomoć ContentFilterMultimedia modula, koji kao i u prethodnim slučajevima implementira IContentFilter programsku spregu.

Zbog specifičnosti i formata podataka uvodi se dodatni nivo apstrakcije u vidu MultimediaContent klase koja objedinjuje sve dostupne informacije o raspoloživom multimedijalnom sadržaju (naziv, format, putanja, veličina, opis) i prilagođava ih definisanim programskim spregama i načinu komunikacije.

Objekat MultimediaContent klase može predstavljati DLNA posluživač, USB uređaj, direktorijum, video ili audio datoteku, sliku ili PVR datoteku).

U zavisnosti od sadržaja objekta MultimediaContent klase, Multimedia Manager modul rukuje na različit način zadatim komandama.

U slučaju direktorijuma i DLNA poslužioca, vrši se zadavanje komande za izlistavanje njihovog sadržaja, a u slučaju video, audio, APK ili PVR datoteka i slika inicira se njihova reprodukcija na odgovarajući način.

Multimedia Manager modul omogućuje navigaciju kroz stablo direktorijuma, implementirajući metode za otvaranje DLNA poslužioca i direktorijuma, zadavanjem URL (eng. *Uniform Resource Locator*) adrese, i vraćanja na prethodni nivo stabla.

Ove funkcije putem JNI sprege komuniciraju sa izvornim bibliotekama, koje ažuriraju svoja stanja i omogućavaju pregled sadržaja na trenutnom nivou stabla direktorijuma.

Za reprodukciju video i audio sadržaja zaduženi su ugrađeni Android MediaPlayer i VideoView moduli. Inicijalizaciju ugrađenih modula vrši Media Controller modul u okviru korisničke aplikacije, koji im kao parametar prosleđuje putanju do željenog sadržaja, i inicira njegovo dekodovanje.

```
MediaController.stopLiveStream();  
MultimediaController.getControlProvider().setContent(content);  
MultimediaController.getControlProvider().play();
```

U slučaju APK datoteke, putem sistemskih metoda, inicira se njihova instalacija.

```
Intent intent = new Intent(Intent.ACTION_VIEW);  
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
String path = mContent.getAbsolutePath();  
  
intent.setDataAndType(Uri.fromFile(new File(path)),  
"application/vnd.android.package-archive");  
ApplicationService.getInstance().startActivity(intent);
```

Nakon uspešno završene instalacije, Android aplikacija će se pojaviti kao dostupna aplikacija u okviru korisničke sprege.

Za reprodukciju PVR datoteka zadužena programska podrška televizijskog prijemnika, o čemu će kasnije biti više reči.

Za reprodukciju slika, zadužen je Image Viewer modul, realizovan u okviru Android korisničke aplikacije, zbog jednostavnije realizacije grafičke sprege ka krajnjem korisniku.

Image Viewer modul realizuje metode zadužene za dekodovanje i algoritme za skaliranje slike uz očuvanje originalnog odnosa između širine i visine.

Na taj način se prilikom prikazivanja slike na ekranu pravi kompromis između njene originalne veličine i kvaliteta, kako bi se zadovoljila pravila “10-foot” grafičke korisničke sprege.

Takođe u Image Viewer modulu su realizovane metode za pronalazak sledeće i prethodne slike u direktorijumu, koje pozivaju metode aplikativnog servisa, kako bi se omogućio jednostavniji pregled velikog broja slika.

Korisnička sprega u okviru korisničke aplikacije Image Viewer modula prikazana je na slici 12.

4.2.4 DVB sadržaj

DVB standard je grupa internacionalno prihvaćenih standarda za digitalnu televiziju, za standardizaciju sistema za prenos digitalizovanog audio i video sadržaja.

DVB sadržaji koji mogu biti prisutni u okviru ovog programskog rešenja su:

- DVB T

- DVB C
- DVB S
- Data
- Radio

Prilikom pretraživanja DVB sadržaja podaci o televizijskim servisima (ime, tip, frekvencija, indeks, itd.) skladište se u lokalnu bazu podataka u obliku tekstualne datoteke.

Obradom DVB sadržaja rukuje programska podrška televizijskog prijemnika.

DTV programska podrška komunicira sa nižim slojevima programske podrške (birači kanala, dekoderi, itd.) putem CHAL modula.

Kako bi se aplikativni deo programskog rešenja (korisnička aplikacija i aplikativni servis) logički odvojio od programske podrške DTV prijemnika uvodi se DTV servis, koji predstavlja spregu televizijske programske podrške sa aplikativnim delom programske podrške.

Na ovaj način neko drugo programsko rešenje može realizovati zasebni aplikativni sloj bez potrebe za menjanjem programske sprege ka programskoj podršci televizijskog prijemnika.

Na nivou aplikativnog servisa DVB sadržajem rukuju modeli filtera definisanih u `ContentFilterDVB_T`, `ContentFilterDVB_C`, `ContentFilterDVB_S`, `ContentFilterData`, `ContentFilterRadio` filter klasama, koje nasleđuju `IContent` programsku spregu i implementiraju metode koje rukuju njima.

Kao apstrakcija prikupljenih podataka uvodi se klasa `ServiceContent`, koja sadrži sve značajne informacije vezane za televizijski servis (ime, indeks, frekvencija, tip, video i audio PID).

DTV servis se pokreće prilikom uključivanja uređaja.

Nakon povezivanja aplikativnog i DTV servisa, sprega ka programskog podršci televizijskog prijemnika je dostupna aplikativnom servisu.

Kako je programska podrška televizijskog prijemnika realizovana u C programskom jeziku DTV servis joj mora pristupati putem JNI sloja.

Moduli koji rukuju prikupljanjem DVB sadržaja u DTV servisu nazivaju se `Service List Control` i `Service List Control Native`. `Content List Control Native` modul sadrži programsku spregu ka JNI sloju programske podrške.

U okviru `Service List Control` modula vrši se dobavljanje prikupljenih informacija o DTV sadržaju iz DTV programke podrške i prosleđuje se putem `IBinder` sprege do aplikativnog servisa, koji dalje obrađuje podatke i prosleđuje ih korisničkoj aplikaciji.

U daljem tekstu opisuje se način na koji aplikativni servis dobavlja informacije o DVB sadržaju iz programske podrške televizijskog prijemnika.

Prilikom inicijalizacije DVB filtera u aplikativnom servisu, neophodno je prikupiti informaciju o broju dostupnih DVB sadržaja preko DTV servisa.

```
int numberOfDvbServices = ApplicationService.getInstance().  
getDtvProxy().getServiceListControl().getServiceList(filterType).  
getServicesNumber(dvbType);
```

DTV servis preko JNI sloja pristupa modulima MAL sloja, a potom se dalje pristupa modulima programske podrške televizijskog prijemnika.

Programska podrška televizijskog prijemnika rukuje informacijama o televizijskom sadržaju u `cm_mlist` modulu.

Modul `cm_mlist` dobavlja informacije o televizijskim servisima iz tekstualne datoteke, čiji je sadržaj popunjen prilikom pretrage DTV sadržaja.

Dobavljene informacije se potom povratnom spregom prosleđuju nazad u više slojeve programskog steka.

Nakon pribavljene informacije o broju dostupnih DVB sadržaja, aplikativni servis mora da zatraži od nižih slojeva detaljnije informacije o njima i prilagodi ih definisanim programskim spregama za komunikaciju.

`getContent(int index)` metoda koja prosleđuje multimedijalni sadržaj od aplikativnog servisa do korisničke aplikacije u slučaju DVB sadržaja kao povratnu vrednost vraća objekat `ServiceContent` klase.

Polja `ServiceContent` klase vezane za televizijski sadržaj se u aplikativnom servisu popunjavaju dostupnim podacima iz DTV programske podrške putem DTV servisa.

Svaki `ServiceContent` objekat sadrži i putanju do slike, tj. logo-a televizijskog servisa, koji se prikazuje u grafičkoj sprezi korisničke aplikacije.

Modul koji rukuje slikama i logo-ima DVB sadržaja naziva se `Image Manager` modul.

`Image Manager` modul pristupa direktorijumu sa skupom slika, koji se nalazi na lokalnom sistemu podataka na samom uređaju, na

predefinisanoj putanji, i na osnovu imena DVB sadržaja pronalazi odgovarajuću sliku koju treba prikazati na ekranu.

Kada korisnik u korisničkoj aplikaciji zada komandu za početak reprodukcije nekog DVB sadržaja, korisnička aplikacija proziva `goContent(Content content)` metodu definisanu u DVB filter modulima u aplikativnom servisu. Ova metoda prosleđuje indeks DVB sadržaja i identifikator rute do DTV servisa.

```
ApplicationService.getInstance().getDtvProxy()  
.getServiceLists().getList (ServiceType.MASTER_LIST) .  
setCurrentServiceByIndex (liveRouteId, indexInMasterList);
```

Dalje se informacije o željenom DVB sadržaju putem JNI sloja i MAL modula prosleđuju do programske podrške televizijskog prijemnika, odakle se rukuje reprodukcijom DVB sadržaja.

Na osnovu tipa dostupnog birača kanala (DVB T, DVB C, DVB S), izabranog dekodera video podataka, formira se identifikator rute.

Identifikator rute određuje koji će resursi fizičke arhitekture (npr. dekodер, demultiplekser, itd.) biti korišteni za reprodukciju DVB sadržaja.

Na osnovu ova dva parametra programska podrška televizijskog prijemnika može da započne reprodukciju željenog sadržaja.

Prikaz video komponente iz DTV sadržaja prikazuje se u zasebnoj grafičkoj ravni u korisničkoj aplikaciji uz pomoć `VideoView` grafičke komponente.

Izgled korisničke aplikacije prilikom reprodukcije video sadržaja dat je na slici 11.

4.2.5 IP sadržaj

Internet protokol televizija (IPTV) je sistem kroz koji se pružaju usluge za internet televiziju putem komutacije mrežne infrastrukture, arhitekture i umrežavanja internet protokola. Na primer širokopojsnim mrežama se pristupa putem globalne mreže, umesto da se isporučuju kroz tradicionalne formate. IPTV je jedno od novijih dostignuća iz oblasti primene telekomunikacija u televizijske svrhe.

Pre nego što se video signal propusti kroz sistem, potrebno je izvršiti njegovu kompresiju. Najčešće primenjivan postupak za kompresiju u ove svrhe je H.264.

U okviru ovog programskog rešenja IP sadržaj se razlikuje u nekim detaljima u odnosu na DVB televizijski sadržaj.

Postoji više načina prikupljanja informacija o IP sadržaju u zavisnosti od vrste i izvora podataka. Ove informacije su definisane od strane dobavljača IP sadržaja, koji je zadužen za definisanje sadržaja toka podataka, kao i ograničenja koje je neophodno ispuniti zarad njihovog dobavljanja.

Prikupljanje informacija se vrši parsiranjem podataka iz konfiguracione datoteke.

Konfiguraciona datoteka obično predstavlja tabelu raspoloživog sadržaja, gde su od najvećeg značaja polja koja definišu ime IP kanala i URL adresu.

U zavisnosti od snabdevača, URL adresa može predstavljati IP adresu sa definisanim komunikacionim portom, ili putanju do datoteke .m3u8 formata.

Datoteka m3u8 formata, je tekstualna datoteka, koja podržava UTF-8 tekstualni format, i koja definiše putanje do jednog ili više multimedijalnih sadržaja.

Parsiranjem podataka može rukovati programska podrška televizijskog signala ili aplikativni servis. Ukoliko parsiranjem podataka rukuje programska podrška televizijskog prijemnika, informacije se čuvaju na isti način kao i informacije o DVB sadržaju.

Te informacije se kasnije putem DTV servisa, na zahtev korisničke aplikacije, dobavljaju do aplikativnog sloja.

Ukoliko parsiranjem podataka rukuje aplikativni servis, te informacije se naknadno dodaju u listu dostupnog televizijskog sadržaja.

Na ovaj način je omogućena podrška za različite dobavljače IP sadržaja.

Nakon prikupljanja informacija o svim dostupnim IP sadržajima, formira se jedinstvena lista unutar aplikativnog servisa, koja okuplja sve IP kanale na jednom mestu.

Kako bi se objedinile sve moguće informacije vezane za IP sadržaj, uvodi se klasa IpContent. Kao i ostale uvedene apstrakcije podataka IpContent implementira parselabilni protokol, kako bi se mogao proslediti putem IBinder sprege do korisničke aplikacije.

Reprodukcijom IP sadržaja moguće je rukovati na više načina.

Reprodukcijom IP sadržaja obično rukuje programska podrška televizijskog prijemnika, ukoliko format datoteke nije poznat ili nije podržan od strane DTV podrške, moguće je izvršiti reprodukciju sadržaja putem ugrađenog Android MediaPlayer modula, koji poseduje mogućnost reprodukcije toka podataka putem HTTP protokola. Tada se

pomoću FFMpeg biblioteke ispituje tip sadržaja i instancira neki od modula nižeg nivoa za reprodukciju Android platforme.

Ukoliko je prikupljanje informacija obavljeno unutar aplikativnog servisa, programska podrška televizijskog prijemnika, nema uvid u njegovo postojanje.

U tom slučaju aplikativni servis ne može koristiti prethodno pomenute metode za iniciranje reprodukcije unutar DTV programske podrške (npr.za reprodukciju DVB sadržaja).

Iz tog razloga se u okviru DTV servisa definiše metoda `setServiceUrl(String url, int routeId)`, koja omogućava da se ovaj problem prevaziđe.

`setServiceUrl` metoda omogućuju prozivanje metoda nižih slojeva programske podrške koje iniciraju reprodukciju IP sadržaja na definisanoj URL adresi.

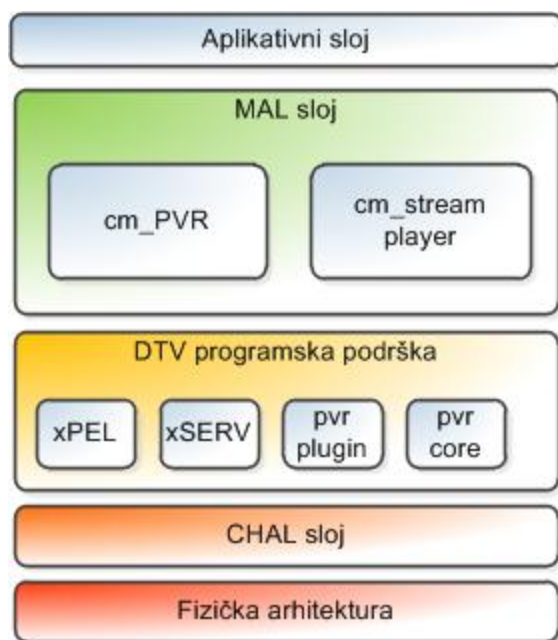
Ukoliko su informacije o IP sadržaju prikupljene parsiranjem konfiguracione datoteke od strane programske podrške televizijskog prijemnika, poziv metoda nižih nivoa je isti kao i za DVB format, jer programska sprega ne pravi razliku između tipa sadržaja, ukoliko su informacije o njemu već uskladištene u listama dostupnog sadržaja.

4.2.6 PVR sadržaj

Sa pojavom digitalne televizije, mogućnosti manipulacije snimanja i reprodukcije televizijskog sadržaja su porasle. Može se reći da je sa pojavom digitalnih video rekordera (eng. *Digital Video Recorder*) krajem prošlog veka započela revolucija u načinu manipulacije televizijskim sadržajima koja je u mnogome promenila koncept same televizije. Profilisanjem digitalnih video rekordera u lične video rekordere (eng.

Personal Video Recorder) postalo je moguće zakazivanje snimanja emisija, serija, filmova na osnovu elektronskog programskog vodiča (eng. *Electronic Program Guide*).

Reprodukcija snimljenog sadržaja mora da bude verodostojan prikaz televizijskog servisa u smislu kvaliteta njegovih komponenti – zvuka, slike, teleteksta i prevoda i u smislu sinhronizacije njihovog predstavljanja. Mogućnosti koje aplikacija mora da pruži u smislu upravljanja reprodukcijom jesu pauziranje, premotavanje unapred i unazad, kao i traženje određenih sekvenci na osnovu vremenskih odrednica.



Slika 8 Organizacija PVR modula u sklopu programske podrške

U okviru ovog programskog rešenja prikupljanjem dostupnog PVR sadržaja i njegovom reprodukcijom rukuju Pvr Control modul u DTV servisu i niži slojevi programske podrške.

Realizacija PVR programske podrške prikazana je na slici 8.

Uloga cm_Pvr modula jeste omogućavanje fleksibilnog i jednostavnog rukovanja nižim modulima PVR sistema, sa strane aplikativnog sloja.

Upravljanje snimanjem TV servisa, odloženim gledanjem i listama snimljenih servisa vrši xPEL modul. Zajedno sa xSERV modulom predstavlja osnovnu logiku PVR sistema.

Takođe, omogućava programabilno snimanje servisa pomoću ugrađenih vremenskih brojača (u formatu vreme, datum, godina) ili na osnovu EPG informacija.

Realizaciju nižeg nivoa programske podrške TV prijemnika čine moduli pvr_plugin i pvr_core. Ovi moduli predstavljaju spregu sa adaptacionim slojem programske podrške TV prijemnika (CHAL sloj).

Takođe omogućavaju instalaciju, komunikaciju i sve vrste standardnih operacija nad medijumima koji se koriste za skladištenje podataka.

Dobavljanjem PVR sadržaja na nivou aplikativnog sloja rukuje Multimedia Manager modul u okviru aplikativnog servisa.

Multimedia Manager pribavlja informacije o broju dostupnih PVR sadržaja pozivanjem metoda DTV servisa.

```
int numberOfPvrContents = ApplicationService.getInstance().  
getDtvProxy().getPvrControl().updatePlaylist();
```

Kao apstrakcija dostupnih informacija koristi se MultimediaContent klasa i ContentFilterPvrRecorded modul.

Nakon dobavljanja informacije o broju dostupnih PVR sadržaja Multimedia Manager zatražuje detaljnije informacije o njima od programske podrške televizijskog prijemnika i formira objekte klase MultimediaContent.

Nakon uspešnog snimanja PVR sadržaja i njegovog skladištenja na neki sistem podataka, korisnička aplikacija može zatražiti od programske podrške televizijskog prijemnika iniciranje njegove reprodukcije.

Korisnička aplikacija prosleđuje informaciju o rednom broju PVR sadržaja, koji predstavlja njegovu jedinstvenu oznaku, ka aplikativnom servisu, odakle se putem DTV servisa nastavlja komunikacija sa nižim slojevima programske podrške.

```
ApplicationService.getInstance().getDtvProxy().getPvrControl().  
startPlayback(int routeId, int pvrContentIndex);
```

Kako se izvor sadržaja koji treba dekodovati razlikuje od izvora podataka DVB sadržaja, koji se dobavlja preko birača kanala, identifikator rute mora odrediti drugi skup modula fizičke arhitekture koji će se koristiti prilikom reprodukcije PVR sadržaja.

DTV servis dalje proziva metode cm_Pvr modula, koji komunicira sa samom programskom podrškom televizijskog prijemnika i obezbeđuje prikaz snimljenog sadržaja na ekranu.

Za prikaz usnimljenog PVR sadržaja korisnička aplikacija koristi identičnu grafičku spregu kao i za reprodukciju nekog video sadržaja sa DLNA usluživača. To znači da krajnji korisnik ima kontrolu nad

reprodukcijom PVR sadržaja, tj. reprodukcija može biti zaustavljena ili privremeno pauzirana. Isto tako moguće je premotavati usnimljeni sadržaj unapred ili unazad.

Korisnička aplikacija obezbeđuje odgovarajuću grafičku spregu, koja omogućuje krajnjem korisniku da pristupa pomenutim kontrolama reprodukcije.

Izgled grafičke sprege pri reprodukovanju usnimljenog PVR sadržaja prikaz je na slici 12.

4.2.7 Zaštita od neželjenog pristupa sadržaju

U okviru ovog programskog rešenja realizovana je mogućnost zaštite od neželjenog pristupa multimedijalnim sadržajima.

Na taj način krajnjem korisniku se omogućuje da zaštiti željeni multimedijalni sadržaj od neželjenog pristupa drugih korisnika.

Korisnička aplikacija omogućuje krajnjem korisniku da zaključa ili odključa željeni sadržaj putem jednostavne grafičke sprege.

Prilikom zaključavanja sadržaja, korisniku se nudi mogućnost da definiše željenu šifru, koja se potom skladišti u DTV programskoj podršci.

Za iniciranje reprodukcije zaključanog sadržaja, neophodno je da korisnik unese ispravnu šifru.

Kako ovo programsko rešenje sadrži i multimedijalne sadržaje, kojima ne rukuje DTV podrška, ova funkcionalnost je realizovana na više nivoa programskog steka.

U zavisnosti od tipa multimedijalnog sadržaja zaštita neželjenom pristupu se obezbeđuje uz pomoć DTV programske podrške ili aplikativnog servisa.

Na nivou DTV programske podrške informacije o zaključanim DVB sadržajima se čuvaju u slist modulu, koji je u sprezi sa već pomenutim cm_mlist modulom.

Ovi moduli pristupaju lokalnoj bazi podataka, gde prikupljaju osnovne informacije o dostupnom DVB sadržaju.

Sadržaj slist modula se periodično skladišti u lokalnu bazu podataka, kako bi se nakon iskućivanja uređaja zadržalo ispravno stanje.

Iniciranje obrade informacija o zaključanim DVB sadržajima vrši se pomoću Parental Control modula u DTV servisu.

Na nivou aplikativnog servisa rukovanjem o zaključanim informacijama zadužen je Content List Control modul.

Treba napomenuti da se na aplikativnom nivou rukuje zaključavanjem Android aplikacija, Android namenskih grafičkih kontrola i IP sadržaja, jer DTV programska podrška, nema uvid u njihovo postojanje i ne ume da rukuje njima.

U okviru Content List Control modula realizuje se lista zaključanih sadržaja, koja se periodično osvežava i čuva u lokalnoj bazi podataka, na aplikativnom nivou.

Na taj način se obezbeđuje ispravno stanje aplikacije nakon gašenja uređaja.

Svaki zaključani sadržaj se dodaje u listu sa jedinstvenom oznakom, kako bi se prilikom zahteva za otključavanjem, sadržaj jednostavno mogao ukloniti iz liste zaključanih sadržaja.

Zbog različitosti realizacije modula za kontrolu neželjenog pristupa, korisnička aplikacija na osnovu tipa sadržaja, određuje koji će moduli biti iskorišteni.

Prilikom iscertavanja grafičkih elemenata, korisnička aplikacija za svaki vidljivi sadržaj u grafičkoj sprezi traži informacije o njegovom statusu (zaključan ili otključan) od nižih slojeva programskog steka.

Na ovaj način korisniku se u svakom trenutku prikazuje ažurno stanje svih dostupnih multimedijalnih sadržaja.

Činjenica je da televizijski prijemnik obično koristi više osoba u okviru jednog domaćinstva, baš iz razloga što ovakav digitalni prijemnik TV signala podržava veliki broj različitih multimedijalnih sadržaja, moguća je pojava i multimedijalnih sadržaja koji nisu namenjeni svim uzrastima.

Zbog toga realizacija ove funkcionalnosti predstavlja veoma značajnu opciju televizijskog prijemnika, jer korisnicima putem jednostavne korisničke sprege pruža kontrolu nad „nedozvoljenim“ sadržajem.

Slika 15 prikazuje izgled korisničke sprege, koja omogućuje zaključavanje željenog sadržaja.

Na slici 16 prikazana je grafička sprega korisničke aplikacije, koja sadrži zaključane multimedijalne sadržaje.

Slika 17 prikazuje korisničku spregu koja inicira reprodukciju zaključanog sadržaja.

5. Ispitivanje i verifikacija

U okviru ovog rada obavljena su ispitivanja realizovanih modula kroz tri vrste ispitivanja.

Kao prva vrsta ispitivanja, definisana je i napravljena zasebna konzolna aplikacija. U okviru date aplikacije, napravljen je niz ispitnih slučajeva koji potvrđuju ispravno funkcionisanje nezavisnih modula.

Druga vrsta ispitivanja obuhvata korišćenje nezavisnih JUnit ispitnih slučajeva, kojima su verifikovani svi moduli koji su predmet ovog rada.

Treća vrsta ispitivanja predstavlja zasebna Android aplikacija pisana u cilju uspostavljanja okruženja koje odgovara funkcionalnosti digitalnog TV prijemnika u realnom vremenu. Svi realizovani moduli su uključeni u prethodno pomenutnu Android aplikaciju i utvrđeno je da izolovani ispitni slučajevi ne pružaju validne rezultate, tj. metode koje vraćaju ispravnu vrednost kada se izolovano pozivaju, vraćaju neispravne

rezultate kada se uključe u televizijsku aplikaciju, tako da je kod u iteracijama ispravljan da bi se došlo do potpuno ispravnog rešenja.

U nastavku su prikazani ispitni slučajevi u okviru zasebne TV aplikacije, kojom se potvrđuju funkcionalnosti digitalnog TV prijemnika u realnom sistemu:

- Prikupljanje i grafički prikaz dostupnog multimedijalnog sadržaja
- Reprodukcija multimedijalnog sadržaja
- Kontrola zaštite neželjenom pristupu

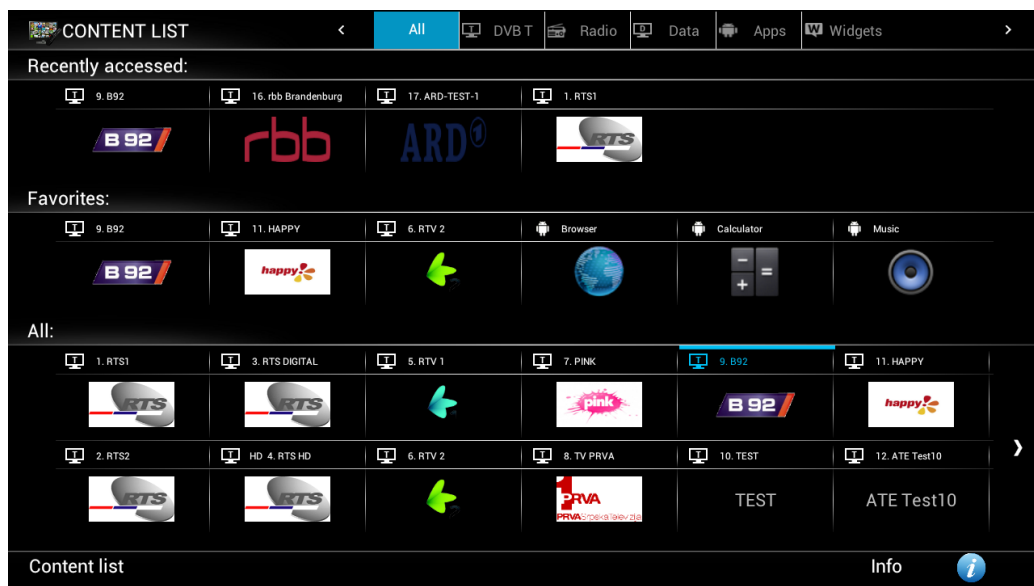
5.1 Prikupljanje i grafički prikaz multimedijalnog sadržaja

Ispitivanje ove funkcionalnosti obuhvata ispitivanje više različitih ispitnih slučajeva, tj. različitih modula koji su zaduženi za prikupljanje odgovarajućeg tipa multimedijalnog sadržaja.

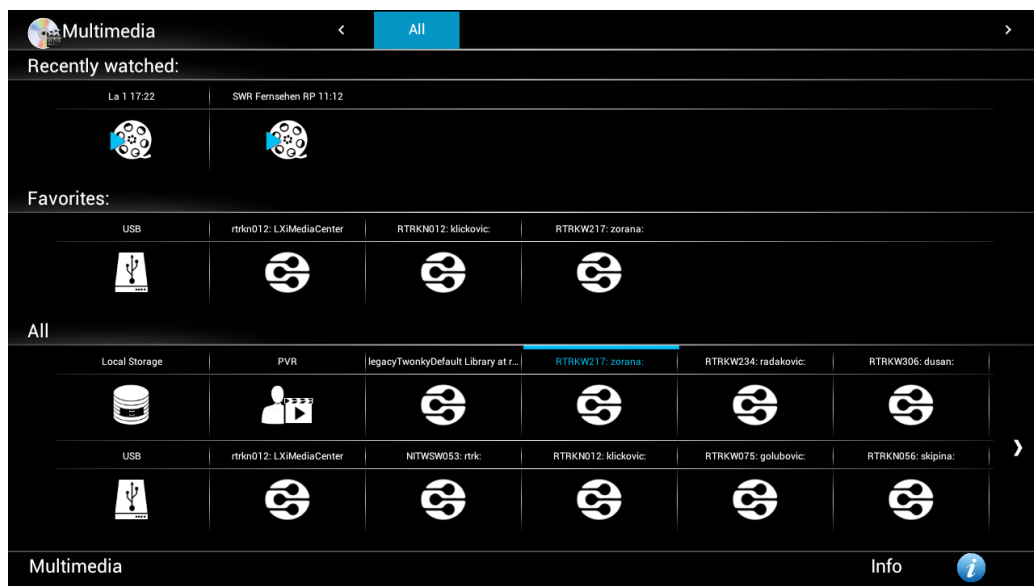
Procedura prikupljanja dostupnog multimedijalnog sadržaja podrazumeva ispravnu funkcionalnost realizovanih modula na različitim nivoima programske podrške.

Kako bi se utvrdilo da li programska podrška ispravno funkcioniše potrebno je ispitati funkcionlanost modula na aplikativnom nivou, kao i funkcionalnost nižih modula programske podrške (aplikativni servis, DTV servis, programsku podršku DTV prijemnika).

Zadavanjam komande za prikaz dostupnog multimedijalnog sadržaja unutar korisničke aplikacije, inicira se komunikacija između različitih nivoa programske podrške.



Slika 9 Grafička sprega sa prikazanim Android i DVB sadržajem



Slika 10 Grafička sprega sa prikazanim sadržajem sa sistema podataka i DLNA mreže

Korisnička aplikacija prikazuje grafičku spregu u zasebnoj grafičkoj ravni koja prikazuje dostupan multimedijalni sadržaj.

Sadržaj je raspoređen u tri definisane podliste, koje prikazuju prethodno korišćeni sadržaj, omiljeni sadržaj i sav dostupan sadržaj iz odedene kategorije.

Sadržaj može biti filtriran po odgovarajućim podržanim tipovima.

Slike 9 i 10 prikazuju grafičku spregu za prikaz dostupnog multimedijalnog sadržaja.

Analizom prikazanog sadržaja na prethodnim slikama može se utvrditi:

- Programsko rešenje prepoznaje različite tipove multimedijalnog sadržaja.
- Filtriranje željenog tipa multimedijalnog sadržaja ispravno funkcioniše.
- Rukovanje različitim listama dostupnog multimedijalnog sadržaja daje očekivane rezultate.

Nakon ispitivanja ove funkcionalnosti programske podrške, na osnovu dobijenih rezultata, može se zaključiti da svi moduli ispravno funkcionišu i da je obezbeđen ispravan mehanizam komunikacije među njima.

U daljim ispitnim slučajevima podrazumeva se da moduli za prikupljanje dostupnog multimedijalnog sadržaja ispravno funkcionišu.

5.2 Reprodukcija multimedijalnog sadržaja

Kako je jedan od glavnih zahteva koji se stavlja pred ovo programsko rešenje da objedini različite dostupne multimedijalne

sadržaje u jedinsteni skup, implementacija modula koji podržavaju njihovu reprodukciju se razlikuje od tipa i formata multimedijalnog sadržaja.

Kako bi se u potpunosti ispitala ispravnost modula za reprodukciju, potrebno je ispitati njihovu ispravnost za sve dostupne tipove multimedijalnog sadržaja.

Iniciranje reprodukcije odvija se tako što korisnička aplikacija na osnovu zadate komande od strane krajnjeg korisnika, u zavisnosti od tipa multimedijalnog sadržaja započinje komunikaciju sa odgovarajućim modulima iz aplikativnog servisa ili programske podrške televizijskog prijemnika.

Slika 11 prikazuje izgled korisničke aplikacije pri reprodukciji televizijskog DVB i IP sadržaja.



Slika 11 Izgled korisničke aplikacije pri reprodukovanju televizijskog sadržaja

Na osnovu prethodne slike i prikazanih podataka možemo zaključiti da moduli za reprodukciju DVB sadržaja, realizovanih na različitim nivoima programske podrške, poseduju ispravnu funkcionalnost.

Prilikom reprodukcije PVR sadržaja ili sadržaja prikupljenog sa DLNA posluživača ili nekog od dostupnih sistema podaka, korisnička aplikacija obezbeđuje i osnovne kontrole za rukovanje reprodukcijom (start, pauza, stop, kontrole za premotavanje).

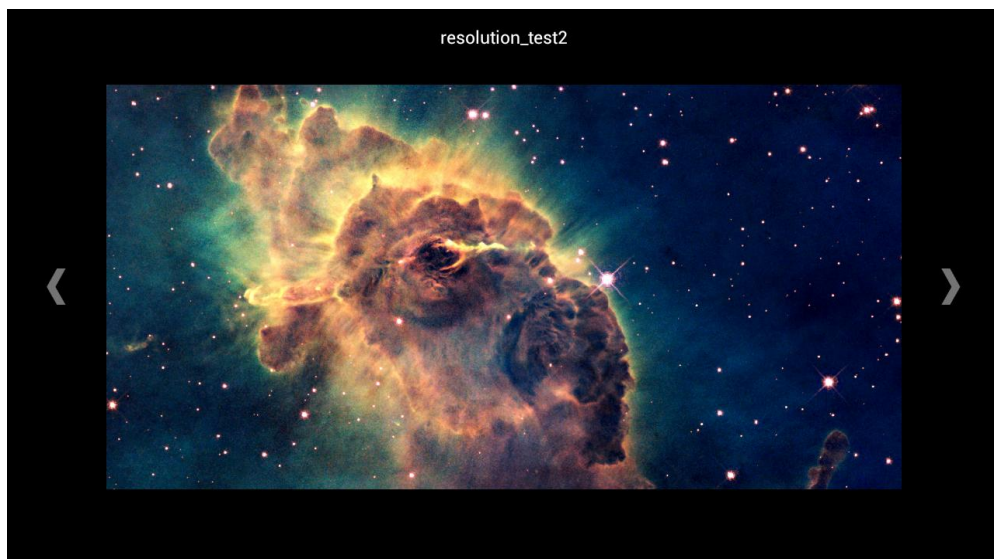


Slika 12 Izgled korisničke aplikacije pri reprodukovanju snimljenog PVR sadržaja

Prethodna slika prikazuje izgled korisničke aplikacije nakon iniciranja reprodukcije PVR ili DLNA sadržaja.

Na osnovu dobijenih rezultata i ispravnog ponašanja korisničke aplikacije, može se zaključiti da moduli zaduženi za reprodukciju i kontrolu ovog tipa sadržaja ispravno funkcionišu.

Prikaz galerije slika pomoću Image Viewer korisničke sprege prikazan je na sledećoj slici.



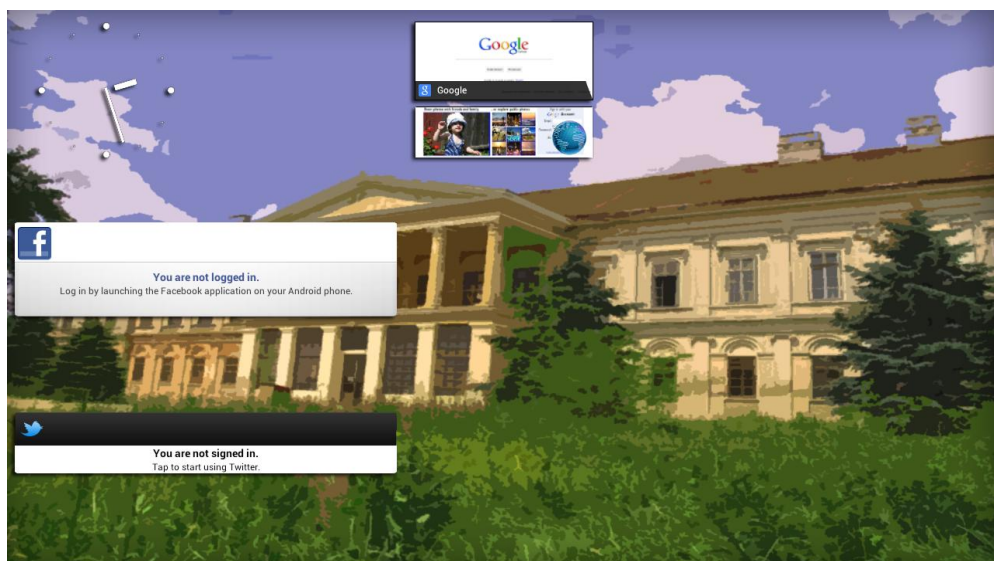
Slika 13 Izgled grafičke sprege za prikaz galerije slika

Slika 13 prikazuje ispravnu funkcionalnosti ImageViewer programskog modula koji je zadužen za dekodovanje i prikaz slika sa različitih izvora podaka.

Uočava se da prikazana slika ne sadrži dodatne artefakte, što dokazuje ispravnu funkcionalnost algoritma za skaliranje i dekodovanje slike.

Reprodukcija Android aplikacija i namenskih grafičkih kontrola, podrazumeva mogućnost njihovog pokretanja, odnosno prikaza na ekranu u zasebnoj grafičkoj ravni.

Slika 15 prikazuje korisničku aplikaciju sa pokrenutim Android namenskim grafičkim kontrolama.



Slika 14 Izgled korisničke aplikacije sa aktivnim Android namenskim grafičkim kontrolama

Na osnovu prethodne slike može se zaključiti da korisnička aplikacija poseduje mogućnost ispravnog prikaza Android namenskih grafičkih kontrola.

Dobijanjem očekivanih rezultata potvrđuje se ispravnost modula za rukovanje Android tipovima podataka kao i ispravnu komunikaciju korisničke aplikacije sa nižim slojevima programske podrške.

5.3 Zaštita od neželjenog pristupa sadržaju

Kako se količina različitog multimedijalnog sadržaja drastično povećava u današnje vreme, a televizijski prijemnik je namenjen korišćenju većem skupu ljudi, zaštita pristupa određenim sadržajima je veoma važna osobina ovakvog programskog rešenja.

Ispitivanje ove funkcionalnosti obuhvata više različitih ispitnih slučajeva:

- Zaključavanje željenog sadržaja
- Prikaz zaključanog sadržaja unutar grafičke sprege
- Reprodukcijska zaključanog sadržaja

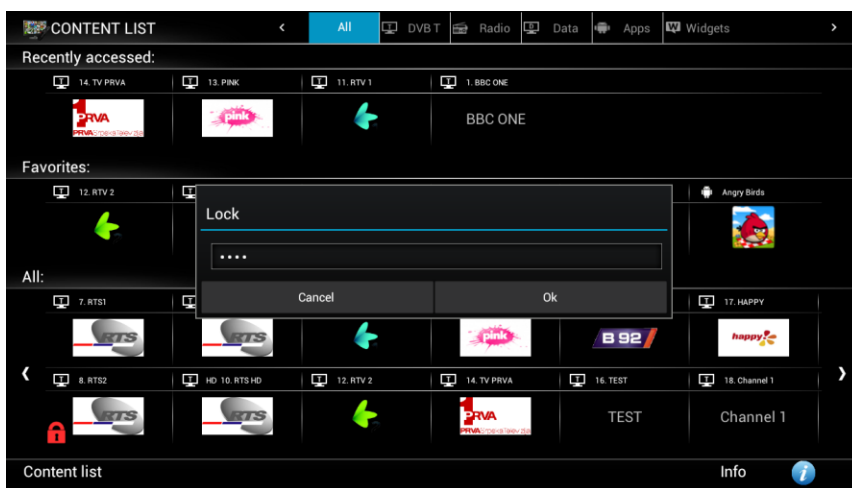
5.3.1 Zaključavanje željenog sadržaja

Korisnička aplikacija pruža grafičku spregu koja omogućuje zaključavanje željenog sadržaja i postavljanje određene šifre.

Obradom komande koju zadaje krajnji korisnik, korisnička aplikacija prikazuje odgovarajuću grafičku spregu koja omogućuje zaključavanje željenog sadržaja.

Nakon uspešnog postavljanja željene šifre, ta informacija se prosleđuje nižim slojevima programske podrške gde se dalje obrađuje i čuva.

Na osnovu informacija iz nižih slojeva programske podrške korisnička aplikacija prikazuje indikaciju da je neki sadržaj zaštićen.



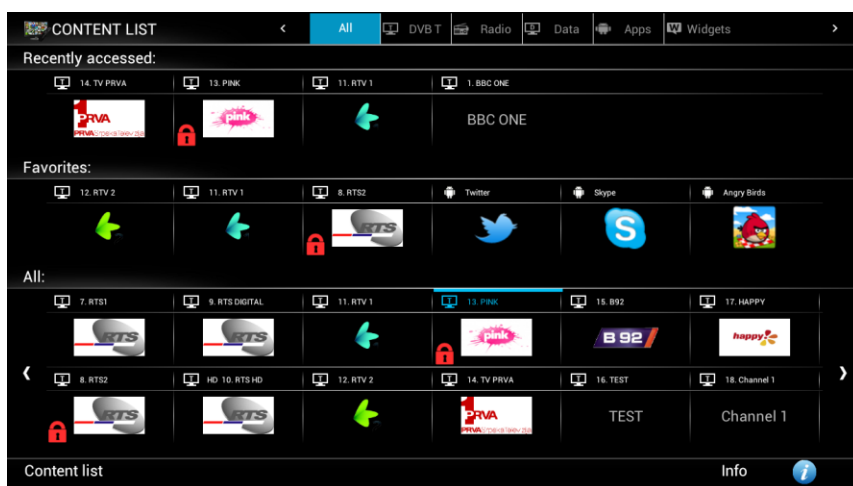
Slika 15 Korisnička sprega za zaključavanje multimedijalnog sadržaja

Prethodna slika prikazuje izgled korisničke sprege koja krajnjem korisniku omogućuje zaključavanje željenog sadržaja.

5.3.2 Prikaz zaključanog sadržaja unutar grafičke sprege

Zaključani sadržaji se razlikuju od ostalih u okviru korisničke sprege uz pomoć posebne grafičke indikacije (ikonica katanca).

Sledeća slika prikazuje izgled grafičke sprege u korisničkoj aplikaciji, koja sadrži “zaštićene” sadržaje.



Slika 16 Korisnička sprega sa zaključanim multimedijalnim sadržajem

Na osnovu prethodne slike može se zaključiti da je obezbeđen ispravan mehanizam komunikacije između modula za zaštitu neželjenog pristupa.

Prikazana je ispravna funkcionalnost modula koji iniciraju zaključavanje željenog sadržaja, kao i modula koji prikupljaju informacije o zaključanim sadržajima iz nižih slojeva programske podrške.

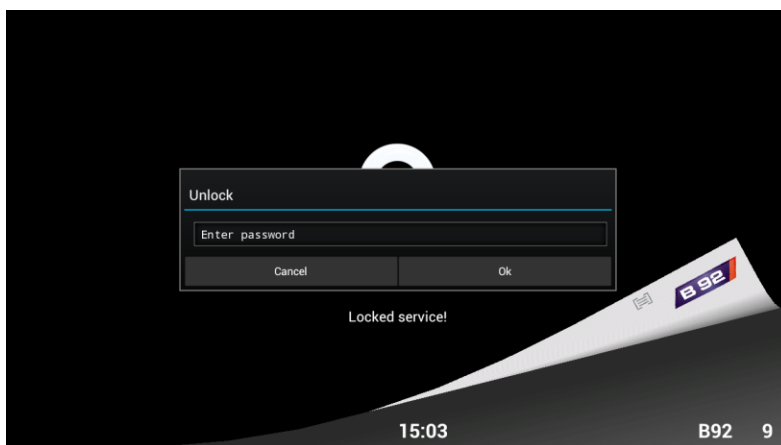
5.3.3 Reprodukcija zaključanog sadržaja

Nakon uspešnog zaključavanja sadržaja, za iniciranje njegove reprodukcije, korisnik mora da unese odgovarajuću šifru, koja je prethodno postavljena.

Nakon uspešnog unosa šifre, korisnička aplikacija će inicirati reprodukciju odgovarajućeg sadržaja.

Ukoliko korisnik unese pogrešan niz karaktera korisnička aplikacija će sprečiti reprodukciju i obavestiti korisnika o nastaloj situaciji.

Sledeća slika prikazuje grafičku spregu koja se prikazuje ukoliko korisnik želi da inicira reprodukciju zaključanog multimedijalnog sadržaja.



Slika 17 Korisnička sprega za iniciranje zaključanog sadržaja

Analizom prethodno prikazanih slika i očekivanog ponašanja korisničke aplikacije, može se potvrditi ispravnost modula za zaštitu od neželjenog pristupa multimedijalnog sadržaju.

6. Zaključak

Ovim radom prikazano je jedno rešenje rukovanja različitim izvorima multimedijalnih podataka u okviru digitalnih televizijskih prijemnika.

U doba sve bržeg razvoja informacionih tehnologija, popularizacije društvenih mreža i različitog interaktivnog sadržaja na globalnoj mreži, količine dostupnog multimedijalnog sadržaja koje stoje na raspolaganju običnom korisniku konstantno rastu.

Uzevši u obzir porast tehnološke pismenosti stanovništva, zaključak je da potrebe običnog korisnika televizijskog prijemnika sve više rastu.

U okviru rada prikazan je način na koji se pogradska podrška Android platforme proširuje sa DTV funkcionalnošću.

Ovo rešenje realizuje različite module koji su zaduženi za prikupljanje, grafički prikaz, rukovanje i reprodukciju različitog

multimedijalnog sadržaja na prijemniku digitalnog televizijskog signala zasnovanog na Android platformi.

Takođe prikazane su različite metode prilagođavanja funkcionlanosti programskog rešenja ograničenjima ciljne platforme.

Na ovaj način objedinjeni su različiti multimedijalni sadržaji sa stanovišta implementacije i njihovog formata u jedinsteni skup, formirajući apstrakciju sa stanovništva krajnjeg korisnika, gde se televizijski sadržaj, Android aplikacije i namenske grafičke kontrole, DLNA poslužioc, USB uređaji, PVR datoteke, slike, video i audio sadržaji tretiraju podjednako sa stanovništva televizijske aplikacije i koriste pomoću jedinstvene grafičke sprege.

Realizacijom ovakvog sistema i demonstracijom njegovih mogućnosti, uočava se i postojanje prostora za dalji napredak.

Jedan od prostora za napredak je svakako smanjenje količine redundantnih podataka prilikom odvijanja komunikacije između različitih slojeva programske podrške. Smanjenjem redundancije dodatno bi se dobilo na performansama i boljem vremenu odziva čitavog sistema. Takođe moguće je unaprediti algoritme dekodovanja i ispitivanja formata multimedijalnih tokova podataka i dodati podršku za neke od protokola za ramenu podataka (SIP, VoIP, FTP, BitTorrent i sl.), što bi dodatno proširilo količinu dostupnih multimedijalnih sadržaja.

Daljim razvijanjem tehnologije, multimedijalna podrška digitalnih televizijskih prijemnika predstavljaće još veći značaj za potrošačko tržište, a zaključuje se da Android platforma pruža velike mogućnosti u pogledu razvoja najrazličitijih multimedijalnih aplikacija i interakcije sa krajnjim korisnikom, što predstavlja veoma svetlu budućnost.

7. Literatura

- [1] RT-RK : <http://stb.rt-rk.com/products/android-stb-line-cat/series-2000/rk-2010/>, učitano 28.04.2013
- [2] Vladimir Kovačević, Miroslav Popović: Sistemska programska podrška u realnom vremenu, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2002
- [3] Vladimir Kovačević: Logičko projektovanje računarskih sistema I – projektovanje digitalnih sistema, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2001
- [4] M.Vidakovic, N.Teslic, T.Maruna, and V.Mihic: Android4TV: aproposition for integration of DTV in Android devices, IEEE 30th International Conference on Consumer Electronics (ICCE), Las Vegas, January 2012
- [5] W. Fisher, Digital Audio and Video Broadcasting Technology, A Practical Engineering Guide, Third Editionv

-
- [6] Android Developers, official Google Android documentation, developers.google.com
 - [7] Digital Network Alliance (DLNA) – Home Network Interoperability Guidelines 1.5
 - [8] Nenad Jovanović, Miloš Milanović, Veljko Ilkić, Borivoj Ivetić, “One proposal of usage of DLNA services on DTV devices and implementation on Android based DTV platform”, IEEE Second International Conference on Consumer Electronics – Berlin, 2013.
 - [9] N. Kuzmanovic, T. Maruna, M. Savic, G. Miljkovic, D. Isailovic, „Google's Android as an application environment for DTV decoder system“, IEEE 14th International Symposium on Consumer Electronics (ISCE), Braunschweig, June 2010.