

Poznań University of Technology
Institute of Computing Science



inż. Paweł Szafer

UPNP DLNA

Master's Thesis

Supervisor: prof. dr hab inż. Jan Brzeziński
Poznań, 2012

Spis treści

Wstęp	3
I Opis technologii	4
1 Standardy UPnP i DLNA	5
1.1 Opis standardu UPnP	5
1.2 Stos protokołów UPnP	6
1.2.1 Adresowanie	8
1.2.2 Wykrywanie	8
1.2.2.1 Wykrywanie: Ogłaszanie	9
1.2.3 Opis urządzeń i usług	12
1.2.4 Kontrola	13
1.2.5 Zdarzenia	14
1.2.6 Prezentacja	15
1.3 Klasy urządzeń multimedialnych UPnP	16
1.3.1 Serwer multimedialny	18
1.3.2 Odtwarzacz multimediiów oraz odtwarzacz renderujący	19
1.3.3 Punkt kontrolny oraz kontroler multimedialny	21
1.3.4 Odmiany mobilne	21
2 Multimedia w UPnP AV/DLNA	23
2.1 Odtwarzanie multimediiów	24
2.2 Klasy multimediiów UPnP	25
2.2.1 Klasa <i>object</i> oraz klasa <i>item</i>	26
2.2.2 Klasa kontener	27
2.2.3 Klasa obraz	27
2.2.4 Klasa audio	28
2.2.5 Klasa wideo.	30
2.3 Rodzaje multimediiów	31
2.3.1 Muzyka	32
2.3.2 Filmy	32
2.3.3 Kodowanie filmów	33
II Implementacja serwera UPnP	34
3 Oprogramowanie serwera UPnP AV	35
3.1 Wymagania wstępne	35

3.2	Wykorzystane narzędzia	36
3.2.1	Framework'i UPnP	37
3.3	Opis programu	37
3.4	Funkcje programu	38
3.4.1	Biblioteka multimediiów	38
3.4.2	Miniatury multimediiów	40
3.4.3	Zmiana opcji	40
3.5	Budowa wewnętrzna serwera	40
3.5.1	Klasa demona	41
3.5.2	Klasa główna serwera UPnP.	41
3.5.2.1	Obiekt MediaStore	41
3.5.2.2	Obiekt MediaItem	41
3.5.3	Modyfikacja frameworka UPnP Coherence	41
3.5.4	Klasa serwera RPC	41
3.5.5	Klasa obsługi zmiany parametrów.	41
4	Testy	42
4.1	Sposób testowania aplikacji.	42
4.2	Współpraca z odtwarzaczami w oprogramowaniu telewizorów	42
4.3	Polecenia UPnP AV i DLNA w praktyce	42
4.4	Napotkane problemy	42
4.5	Wyniki testów	42
5	Przegląd oprogramowania UPnP AV i DLNA	43
5.1	Porównanie oprogramowania serwerów UPnP AV i DLNA	43
5.2	Porównanie oprogramowania odtwarzaczy DLNA i UPnP AV	43
5.2.1	Różnice działania odtwarzaczy różnych producentów	43
5.3	Obsługa standardu.	43
	Bibliografia	44

Wstęp

Technologie udostępniania multimediów w sieciach, zarówno lokalnych, jak i sieci Internetowej stają się coraz bardziej powszechne. Jakkolwiek dostęp do multimediów znajdujących się w Internecie przy pomocy komputera jest bardzo prosty, to problem powstaje w przypadku chęci odtworzenia na ekranie telewizora filmu z wakacji, który znajduje się na dysku komputera. Dotychczas trzeba było nagrać taki materiał na dysk DVD, bądź w nowszych telewizorach “najprostszym” rozwiązaniem było podłączenie dysku USB. Sieci komputerowe LAN zostały stworzone w celu łatwej komunikacji między stacjami roboczymi. Udostępnianie plików było i jest jedną z podstawowych usług, do których można wykorzystywać sieć lokalną. Jednakże wykorzystywane do dzisiaj protokoły udostępniania plików nie zostały zaprojektowane do odtwarzania multimediów i do ich oddzielnego rozpoznawania. Odtwarzacz sieciowy DLNA potrafi odtworzyć film, muzykę, obraz, jednak w większości przypadków nie poradzi sobie skutecznie z dokumentami, arkuszami kalkulacyjnymi i innymi nietypowymi dla niego plikami. Natomiast użytkownik takiego odtwarzacza w przypadku filmów oczekuje możliwości wyboru ścieżki dźwiękowej, napisów oraz tak jak w przypadku odtwarzaczy kaset VHS czy płyt DVD możliwości przewijania materiałów. W ostatnich latach wraz z upowszechnianiem się w gospodarstwach domowych różnego rodzaju urządzeń multimedialnych typu komputery PC, laptopy, netbooki, tablety, smartfony, odtwarzacze Blu-Ray, DVD, telewizory, ramki do zdjęć, przenoszenie multimediów między wieloma urządzeniami stało się uciążliwe. Co gorsza odtworzenie

I Opis technologii

Standardy UPnP i DLNA

Technologia UPnP, której skrót rozwijany jest jako Universal Plug and Play, co w łatwy sposób można zrozumieć jako podłącz i graj (lub odtwarzaj) jest zbiorem protokołów sieciowych, których celem powstania była możliwość wzajemnego wykrywania się przez urządzenia sieciowe takie jak komputery PC, drukarki, punkty dostępu WiFi, routery czy urządzenia mobilne.

UPnP ma za zadanie umożliwić użytkownikowi na łatwe wykrywanie urządzeń sieciowych oraz wykorzystywanie ich możliwości. Jest ono dedykowane dla małych sieci domowych w celach takich jak udostępnianie danych, komunikacja, drukowanie czy szeroko pojęta rozrywka. Niniejsza praca jest poświęcona w głównej mierze ostatniemu zagadnieniu (rozrywka), którego specyfikacja nazywana UPnP A/V (ang. UPnP Audio/Video) jest rozszerzeniem podstawowego standardu UPnP. UPNP AV została stworzona do odtwarzania plików multimedialnych. Definiuje ona takie pojęcia jak serwery multimedialne (ang. Media Servers), Media Renderers??, odtwarzacze multimedii (ang. Media Players). Określone zostały w bardzo elastyczny sposób metody przesyłania multimedii. Z tego powodu UPNP AV nie rozpropagowało się w swojej powszechnej wersji wśród producentów sprzętu elektronicznego. Każdy programista i każda firma mogła zaimplementować odtwarzacz lub serwer w taki sposób, że mogły one ze sobą nie współpracować (pomimo spełnienia wymogów standardu) lub powinny być nad wyrost skomplikowane. DLNA (ang. Digital Living Network Alliance) jest międzynarodową organizacją, powstałą w 2003 roku z inicjatywy firmy Sony. Stworzyła ona standard o tej samej nazwie (DLNA), który korzysta z protokołu UPNP A/V. Standard ten jest jednak bardziej restrykcyjny (np w postaci określenia formatów multimedii odtwarzanych przez urządzenia), dzięki czemu łatwiejszy do implementacji w większości urządzeń, a ponadto wprowadził kilka innowacyjnych cech takich jak ochrona DRM (ochrona przed kopiowaniem).

1.1 Opis standardu UPNP

W 1999 została założona, w głównej mierze dzięki dotacjom firmy Microsoft, fundacja UPnP. Powołana została w celu stworzenia standardu, który umożliwi interakcję urządzeń oraz usług w obrębie sieci lokalnej oraz proste określanie ich możliwości (np. drukowanie).

UPnP jest oparty o architekturę klient - serwer, gdzie urządzenia - serwery, oferują swoje usługi dla klientów (UPnP Control Points). UPnP główny nacisk kładzie na brak konfiguracji urządzeń przez użytkownika, zgodnie z częścią nazwy zarówno fundacji, jak i standardu - plug and play (podłącz i działaj). Opisywana fundacja ustandaryzowała sposób komunikacji szeregu urządzeń, począwszy od włączników światła, drukarek, kończąc na odtwarzaczach audio-wideo. Architektura UPNP AV została stworzona do przesyłania materiałów multimedialnych. UPNP AV określa sposób komunikacji, między serwerem multimedii, a odtwarzaczem. Zbiór protokołów został zaprojektowany tak, aby być niezależny od rodzaju urządzenia, odtwarzanej treści i sposobu przesyłania danych.

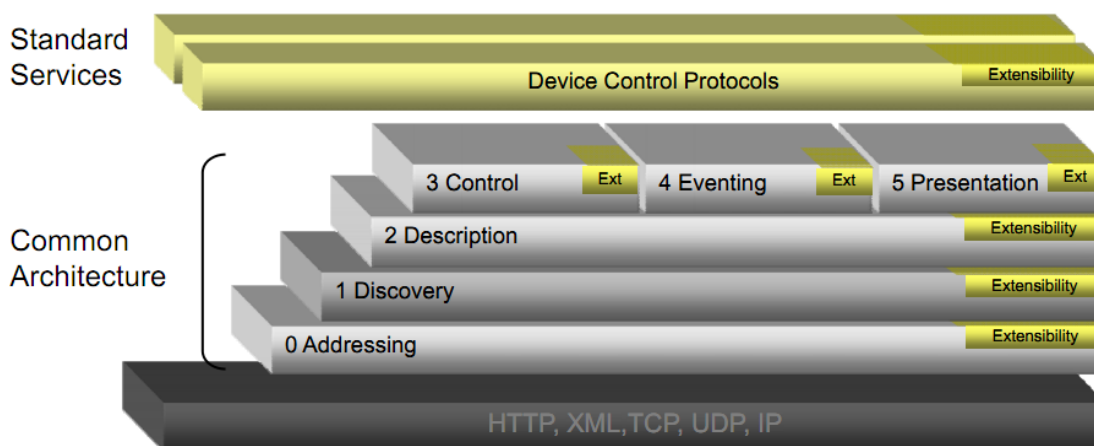
UPnP AV określa sposób komunikacji, między serwerem multimediiów, a odtwarzaczem. Zbiór protokołów został zaprojektowany tak, aby być niezależny od rodzaju urządzenia, odtwarzanej treści i sposobu przesyłania danych.

1.2 Stos protokołów UPnP

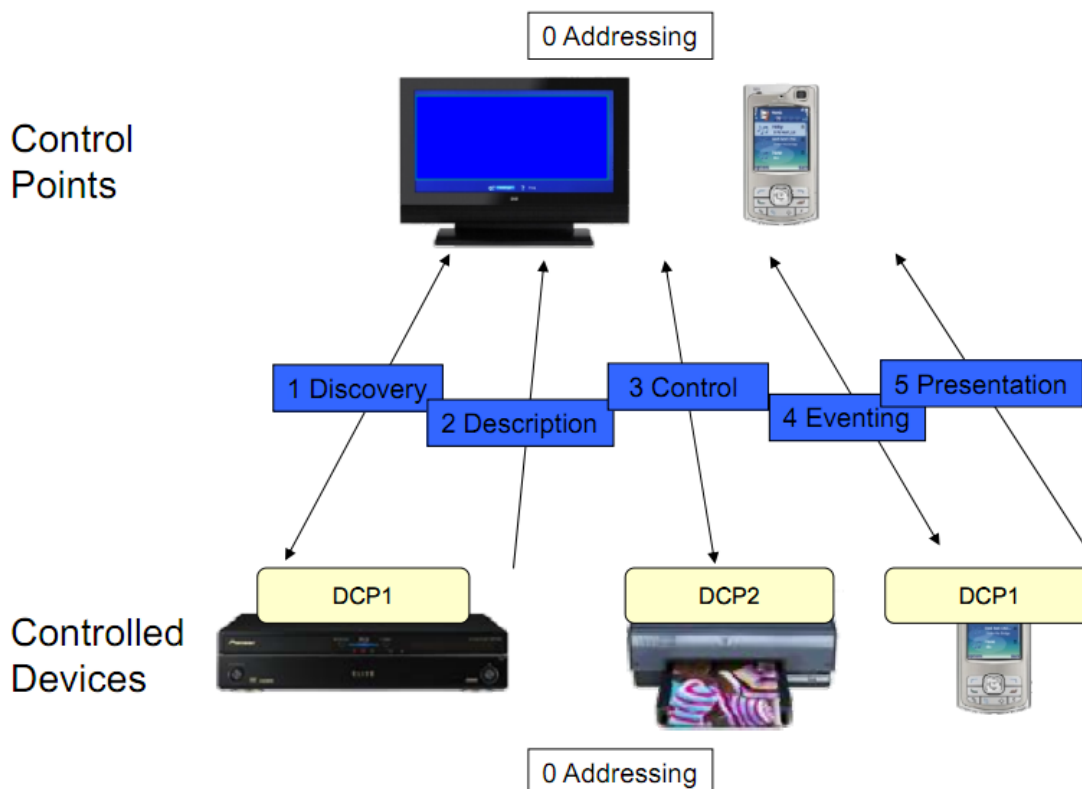
Głównym wymaganiem UPnP jest działanie w obrębie sieci IP, niezależnie od sposobu komunikacji. Połączenie może zostać zrealizowane poprzez sieć Ethernet, połączenia bezprzewodowe takie jak WiFi 802.11abgn, Bluetooth (pomijany jest tu fakt braku możliwości technicznych niektórych protokołów do szybkiego przesyłania danych, co w przypadku niektórych typów multimediiów, takich jak filmy czy muzyka, jest niezbędne).

Raport techniczny UPnP definiuje jedynie sposoby działania w obrębie sieci obsługiwanej przez protokół IPv4. Aneks [10] do głównego raportu określa różnice jakie występują, gdy urządzenie działa w obrębie sieci IPv6. Jeżeli nie napisano inaczej, w niniejszej pracy autor jako pierwszą pozycję określa przypadek sieci IPv4, natomiast drugi przypadek to sieć zgodna z protokołem IPv6.

Rys. 1.1: Protokoły



Rys. 1.2: Komunikacja



Urządzenia UPnP można sklasyfikować w dwóch kategoriach, punktów kontrolnych (ang. Control Points) oraz urządzeń kontrolowanych (ang. Controlled Devices).

W protokole UPnP definiujemy następujące czynności wykonywane przez urządzenia:

- adresowanie IP (ang. IP addressing),
- wykrywanie (ang. discovery),
- opis usługi (ang. description),
- kontrola (ang. control),
- powiadomienia o zdarzeniach(ang. eventing lub event notification),
- prezentacja (ang. presentation).

(<http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20080424.pdf>)

Architektura protokołu wykorzystuje stos protokołów przedstawiony w tabeli

Tabela 1.1: Test tabeli

UPNP Vendor			
UPNP Forum			
UPNP Device			
SSDP		GENA	SOAP
HTTPMU	HTTP U	HTTP	HTTP
IP			

Pierwsza, najwyższa warstwa są to wiadomości zawierające specyficzne informacje dostarczane przez producenta urządzenia dotyczące jego funkcjonalności czy możliwości. Kolejnym el-

ementem są informacje zdefiniowane przez organizację UPnP Forum. Wszystkie te informacje są przekazywane poprzez następujące protokoły:

- SSDP (ang. Simple Service Discovery Protocol),
- GENA (ang. General Event Notification Architecture),
- SOAP (ang. Simple Object Access Protocol).

Do wzajemnego wykrywania się urządzeń UPnP wykorzystuje protokół SSDP. Urządzenia kontrolowane, które można nazwać serwerami usług wykorzystują SSDP do rozgłaszania swojego istnienia. W przypadku punktów kontrolnych SSDP jest wykorzystywany do wykrywania urządzeń pierwszego typu.

W protokole SSDP wykorzystuje się rozgłaszanie HTTP typu unicast lub multicast, z wykorzystaniem UDP (ang. User Datagram Protocol) jako protokołu transportu. Standardowo adresem multicast jest adres 239.255.255.255 oraz port numer 1900.[1]

Drugi protokół transmisji tj. GENA wykorzystywany jest do przekazywania powiadomień o zdarzeniach. W tym przypadku typowo wykorzystywany jest protokół HTTP w sposób “tradycyjny” (z wykorzystaniem protokołu TCP) oraz transmisja UDP typu multicast.

Punkty kontrolne UPnP do wywoływania pożądanej usługi i kontrolowania urządzeń wykorzystują protokół SOAP.

1.2.1 Adresowanie

Zarówno serwer, jak i klient UPnP musi komunikować się z wykorzystaniem protokołu IP. Organizacja UPnP Forum zdefiniowała, że każde urządzenie, które spełnia wymagania specyfikacji musi posiadać oprogramowanie serwera lub klienta protokołu DHCP (ang. Dynamic Host Configuration Protocol). Podczas podłączania się do sieci podejmuje próbę uzyskania adresu IP od serwera DHCP (jeżeli urządzenie nie jest serwerem DHCP). Jeżeli urządzenie nie otrzyma adresu IP powinno wykorzystać tzw. autoadresowanie IP (urządzenie wybiera adres z określonej puli np. 169.254/16 lub w przypadku IPv6 FE80::/64). Raport techniczny stworzony przez organizację UPnP Forum określa dokładnie w jaki sposób urządzenie przydziela automatycznie adres IP, tak aby uniknąć kolizji.

Zarówno w praktyce w większości urządzeń, jak i w aplikacji, którą opisuje niniejsza praca problem adresowania nie jest najważniejszy, gdyż jest rozwiązywany przez oprogramowanie dołączone do sprzętu, bądź systemu operacyjnego.

Wywołania punktów kontrolnych odbywają się poprzez wykorzystanie adresów SOAP SSDP UPNP_FORUM_PUBLIC_RYSUNKI

1.2.2 Wykrywanie

Proces wykrywania polega na rozgłaszaniu przez nowo podłączone urządzenia informacji o swoim istnieniu. Wykorzystywany jest do tego protokół SSDP oraz adres 239.255.255.250 oraz port 1900 (dla IPv6 - [FF02::C]:1900).

Na potrzeby protokołu UPnP wyróżniane są następujące komunikaty SSDP (**rys nr 2.3**):

- ogłaszanie (ang. *advertise*),
- szukanie (ang. *search*),
- odpowiedzi (ang. *responses*).

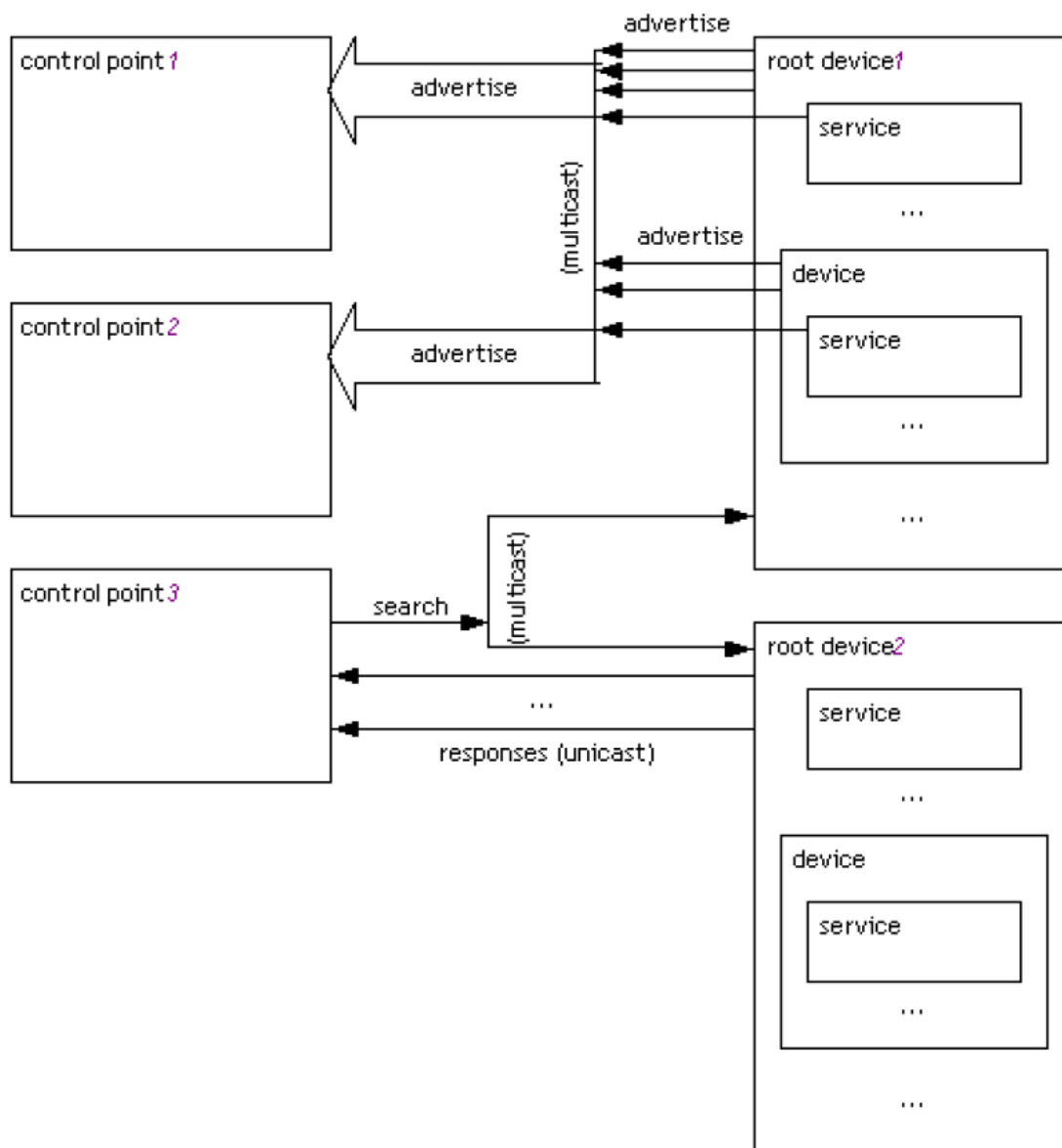
Punkty kontrolne nasłuchują urządzeń na porcie 1900 lub rozgłaszają pakiet z danymi o typie, identyfikatorzeżądanego urządzenia lub usługi. Urządzenia oferujące swoje usługi wysyłają na wcześniej wspomniany adres (239.255.255.250:1900) pakiety SSDP przy pomocy rozgłaszania typu *multicast*.

Odpowiedzi są wysyłane przy pomocy rozgłaszania typu *unicast*.

Urządzenie nowo dołączone do sieci wysyła rozgłoszenie o swoim istnieniu, urządzenia wbudowane w nie również wysyłają rozgłoszenie o sobie oraz o usługach, które świadczą. Kontrolowane odłączenie od sieci wiąże się z obowiązkiem ogłoszenia zdarzenia przez urządzenie. Przyjęta jest w tym przypadku zasada, że powinno ono wysłać tyle komunikatów odłączenia, ile wysłało komunikatów ogłaszających swoje działanie.

Rozgłaszanie wiąże się niestety z obciążaniem sieci, często niepotrzebnym. W tym celu zalecany czas życia pakietu (*TTL* - ang. *time-to-live*) dla każdej wiadomości wynosi 4 (powinna istnieć możliwość modyfikacji tej wartości).

Rys. 1.3: Sposoby wykrywania urządzeń



1.2.2.1 Wykrywanie: Ogłaszanie

W głównej części tego podrozdziału został opisany zarys sposobu ogłaszania się urządzeń w sieci. Jest to jednak bardzo newralgiczna cecha tego protokołu, gdyż bez możliwości wzajemnego wykry-

wania nie ma możliwości, aby urządzenia efektywnie się ze sobą komunikowały. Każde urządzenie/usług wysyła pakiety ogłaszające (*advertise*) swoje działanie na adres 239.255.255.250:1900. Punkty kontrolne nasłuchują na porcie 1900, oczekując na komunikaty wysyłane przez urządzenia. Opisywane komunikaty muszą być wysyłane często oraz co określony interwał czasowy, gdyż zarówno urządzenie, jak i punkt kontrolny mogą zostać dołączone do sieci w różnym czasie.

Każda wiadomość “ogłaszająca” zawiera możliwości urządzenia (np. odtwarzanie muzyki). Ilość wiadomości jest zależna od ilości wbudowanych urządzeń/usług w główne urządzenie. Urządzenia wbudowane w komunikatach dołączają informacje o urządzeniu nadrzędnym (np. informacja o systemie operacyjnym). Oprócz wcześniej przedstawionych cech, komunikaty powinny zawierać czas do wygaśnięcia ogłoszenia (jeżeli po czasie wygaśnięcia urządzenie nadal chce oferować swoje usługi, wówczas powinno wysłać nowy komunikat z nowym czasem wygaśnięcia). Dzięki takiemu podejściu, pomimo, że urządzenia powinny ogłaszać swoje odłączenie od sieci, to nawet, gdy komunikat wyłączenie nie może zostać wysłany, wiadomość rozgłaszająca wygaśnie i nie zostanie przedłużona. Ogranicza to możliwość istnienia w sieci usług już nie działających.

Możliwe jest wysyłanie i odbieranie ogłoszenia przez sprzęt zgodny ze specyfikacją UPnP, jedynie w przypadku, gdy komunikat jest stworzony według schematu z tabeli 1.2

Tabela 1.2: Stos protokołu UPnP w przypadku rozgłaszania – do poprawienia

producent oprogramowania UPnP
organizacja UPnP
architektura urządzenia UPnP
komunikat SSDP
HTTPMU (<i>multicast</i>)
UDP
IP

Pierwsza warstwa opisywanego stosu zawiera informacje specyficzne dla dostawcy urządzenia. Najczęściej producenci dołączają w tym miejscu adres URL do opisu urządzenia, bądź też identyfikator urządzenia (UUID - *ang. Universally unique identifier*). Kolejną warstwą stosu są informacje dostarczane przez organizację UPnP Forum, np. typ urządzenia. Kolejnym dwie warstwy zostaną dokładnie opisane w niniejszym dokumencie, a ich pełen opis znajduje się w specyfikacji protokołu UPnP.

Rys. 1.4: Przykładowy komunikat przyłączający do sieci:

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = seconds until advertisement expires
LOCATION: URL for IPP Printer with 'ipp' scheme
NT: search target
NTS: ssdp:alive
SERVER: OS / version, IPP / 1.1, product / version
USN: advertisement UUID

Trzy pozostałe warstwy nie są tematem niniejszej pracy. Warto jedynie zwrócić uwagę na fakt, iż komunikaty HTTP nie są przesyłane typowo przez protokół TCP, a przez protokół UDP. Jak zostało wspomniane wcześniej urządzenie dołączone do sieci rozgłasza swoje istnienie. Komunikat rozgłaszający wykorzystuje metodę NOTIFY protokołu HTTP. W każdym komunikacie dołączane są informacje w następujących nagłówkach:

- NT - Notification Type - typ ogłoszenia,
- NTS - Notification Subtype - podtyp ogłoszenia,
- USN - Unique Service Name - Unikalna nazwa usługi / UUID
- LOCATION - URL z rozszerzonymi informacjami o urządzeniu,
- CACHE-CONTROL - czas do wygaśnięcia ogłoszenia

Opisywana metoda protokołu HTTP (NOTIFY) wykorzystywana jest przy ogłaszaniu podłączenia, bądź też odłączenia od sieci. W komunikacie podłączenia pole *NTS* jest podtypu *ssdp:alive*, natomiast rozłączenie cechuje podtyp *ssdp:byebye*.

Organizacja stanowiąca zasady protokołu UPnP w przypadku wykrywania urządzeń, poza wzajemnym informowaniem się o obecności (bądź też nieobecności), zagwarantowała również szukanie się wzajemne urządzeń UPnP. Do wyszukiwania służy metoda M-SEARCH, której ogólny format przedstawia **tabela nr.** Pola *HOST* i *MAN* nie mogą zostać zmienione i zawsze muszą mieć taką wartość jak podana w tabeli. Pole *MX* określa maksymalny czas oczekiwania w sekundach. Jego wartość powinna mieścić się w przedziale 1, a 120. Ostatnie pole, *ST* (*Search Target*) określa rodzaj wyszukiwanego urządzeń. Dostępne rodzaje to:

- ssdp:all - wyszukiwanie wszystkich urządzeń i usług,
- upnp:rootdevice - wyszukiwanie jedynie urządzeń głównych,
- uuid:device-UUID - wyszukiwanie konkretnego urządzenia,
- urn:schemas-upnp-org:device:deviceType:v - wyszukiwanie urządzenia według typu. Lista typów możliwych do odnalezienia znajduje się w ...
- urn:schemas-upnp-org:device:serviceType:v - wyszukiwanie urządzenia według typu usługi, którą świadczy. Lista typów **jak wyżej**
- urn:domain-name:device:deviceType:v - takie jak przeszukiwanie urządzenia według typu, z tym, że nazwę domeny typ urządzenia i jego wersję definiuje producent urządzenia UPnP,
- urn:domain-name:device:serviceType:v - takie jak przeszukiwanie urządzenia według typu usługi, z tym, że nazwę domeny typ urządzenia i jego wersję definiuje producent urządzenia UPnP.

Bezpośrednio powiązana z komunikatem M-SEARCH jest odpowiedź na nią, którą urządzenie odpowiadające wysyła na adres źródłowy wcześniejszego pakietu. Odpowiedź jest zgodna z protokołem HTTP. Jej schemat widnieje w **tabeli nr 4.** Odpowiedź nie zawiera treści (*HTTP BODY*), lecz musi posiadać jedną pustą linię poniżej nagłówka. Zarówno dla metody *M-SEARCH*, jak i odpowiedzi na nią nie ma potrzeby określania czasu życia pakietu *TTL*.

Tabela 1.3: Metoda M-Search

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: „ssdp:discover”
MX: seconds to delay response
ST: search target
```

Tabela 1.4: Odpowiedź na metodę M-SEARCH

HTTP/1.1 200 OK
 CACHE-CONTROL: max-age = seconds until advertisement expires
 DATE: when response was generated
 EXT:
 LOCATION: URL for UPnP description for root device
 SERVER: OS/version UPnP/1.0 product/version
 ST: search target **OPISAĆ STRONA 21**
 USN: advertisement UUID

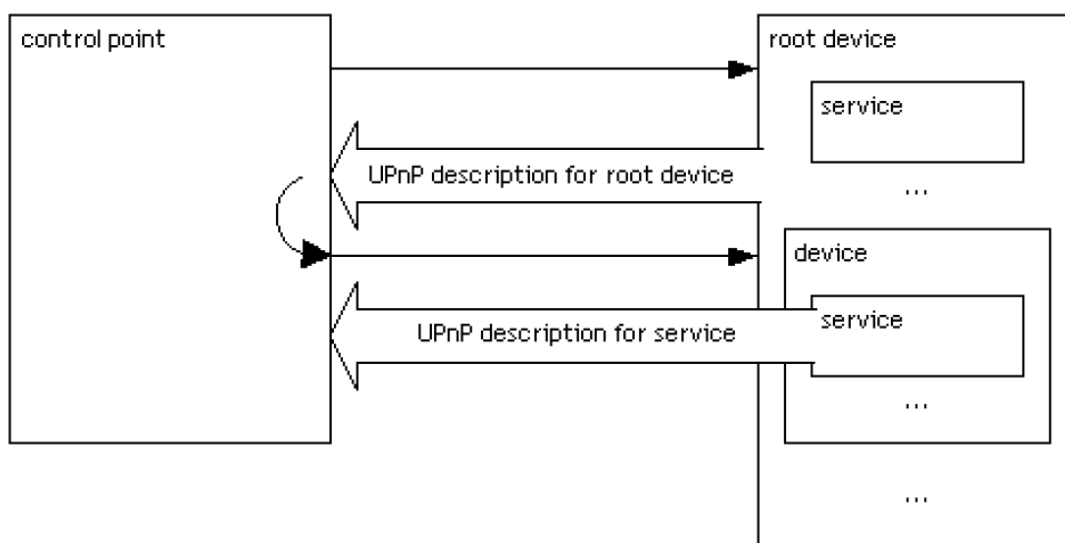
1.2.3 Opis urządzeń i usług

Po udanym wyszukaniu urządzenia, punkt kontrolny posiada jedynie podstawowe informacje o urządzeniu, które zostały zawarte we wcześniejszych komunikatach (Wykrywanie). Do interakcji z urządzeniem punkt kontrolny potrzebuje więcej danych, niż te zawarte w pierwszych komunikatach. W tym celu może pobrać dokładne informacje z adresu URL przesyłanego wraz z komunikatem o przyłączeniu do sieci (ang. *advertising*). Pobiera informacje nazywane są opisem usługi (ang. *description*).

Uproszczony schemat pobierania informacji o urządzeniu znajduje się na rysunku 1.5. Pojedynczy schemat pobierania opisu można podzielić na dwie operacje, uzyskania informacji o urządzeniach oraz o usługach, które one oferują. W najprostszym schemacie cały proces wykonywany jest w kilku krokach:

1. Pobranie informacji o głównym urządzeniu (ang. *root device*),
2. Pobranie informacji o usługach udostępnianych przez urządzenie (ang. *services*).

Opis urządzenia zawiera informacje o fizycznych i logicznych kontenerach (**opisać termin kontener**). Natomiast opis usług określa w sposób bardzo dokładny oferowane przez urządzenie funkcje. Pojedyncze urządzenie fizyczne może zawierać wiele urządzeń logicznych, każde z nich może być skonfigurowane jako główne (ang. *root device*). Ponadto każde urządzenie logiczne może oferować jedną lub wiele usług.

Rys. 1.5: Schemat pobierania opisu usługi

Opis urządzenia/usługi jest napisany przez dostawcę oprogramowania. Jest napisany w języku XML i zalecane jest, aby był zgodny z wytycznymi przedstawionymi w szablonie opisu urządzenia. [11] Najważniejszym elementem z punktu widzenia autora jest sposób opisu usługi, który powinien być stworzony tak, aby punkty kontrolne jak największej liczby producentów (stan idealny - wszystkich producentów) sprzętu i oprogramowania potrafiły uzyskać przydatne dla nich informacje.

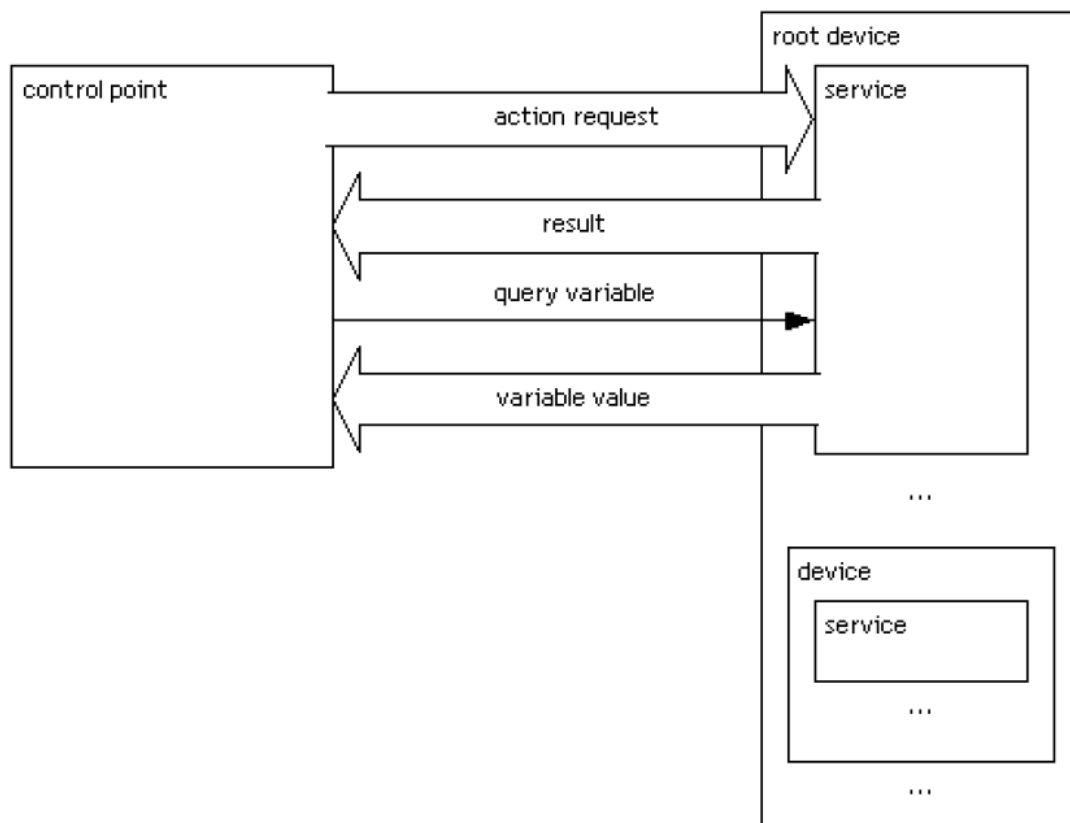
W celu otrzymania opisu urządzenia (bądź usługi) wykorzystywana jest metoda GET protokołu HTTP. Adres URL znajduje się w wiadomościach z ogłoszeniem o przyłączeniu do sieci.

1.2.4 Kontrola

Po uzyskaniu opisu urządzenia oraz usługi, punkt kontrolny może wywołać akcje, które mogą wykonywać dostawcy usług. Wywoływanie usługi przez punkt kontrolny można określić jako rodzaj wywołań zdalnych RPC (ang. *Remote Procedure Call*). Każde polecenie, niezależnie czy wywoływane przez punkt kontrolny czy jako odpowiedź zwrotna od usługi, jest enkapsulowane w komunikatach SOAP (ang. *Simple Object Access Protocol*) poprzez wywołania protokołu HTTP. Wywoływanie procedur przez punkt kontrolny może (lecz nie musi) być przedstawione poprzez modyfikację zmiennych określających stan usługi. O każdym takim zdarzeniu powinny zostać poinformowane wszystkie zainteresowane punkty kontrolne.[12] Punkt kontrolny może wykorzystywać wywołania zdalne, tak długo, aż wcześniej wysłane ogłoszenie nie wygaśnie lub nie zostanie odwołane (przesłanie nowego ogłoszenia, wydłuża możliwość wykonywania wywołań kontrolnych).

Przykładowe wywołanie akcji do usługi znajdującej się w urządzeniu UPnP zostało umieszczone na rysunku 1.6

Rys. 1.6: Wywołanie akcji przez punkt kontrolny



1.2.5 Zdarzenia

W opisie urządzenia UPnP przesyłana jest lista akcji możliwych do wykonania przez usługę oraz lista zmiennych modelujących aktualny stan usługi (w trakcie jej działania). Jako zdarzenie (ang. *event*) została zdefiniowana zmiana wartości wcześniej wspomnianych zmiennych. Każdy punkt kontrolny może zasubskrybować (z ang. *subscribe*) otrzymywanie informacji o takich zmianach. Punkt kontrolny, który zamówił subskrypcję zdarzeń nazywany subskrybentem (ang. *subscriber*), natomiast usługę, która takową subskrypcję realizuje (wysyła informację o zmianie stanu zmiennych) określamy jako wydawcę (ang. *publisher*). Wiadomości przesyłane są poprzez protokół GENA (ang. *General Event Notification Architecture*), który jest rozszerzeniem protokołu HTTP zaprojektowanym na potrzeby organizacji UPnP Forum.

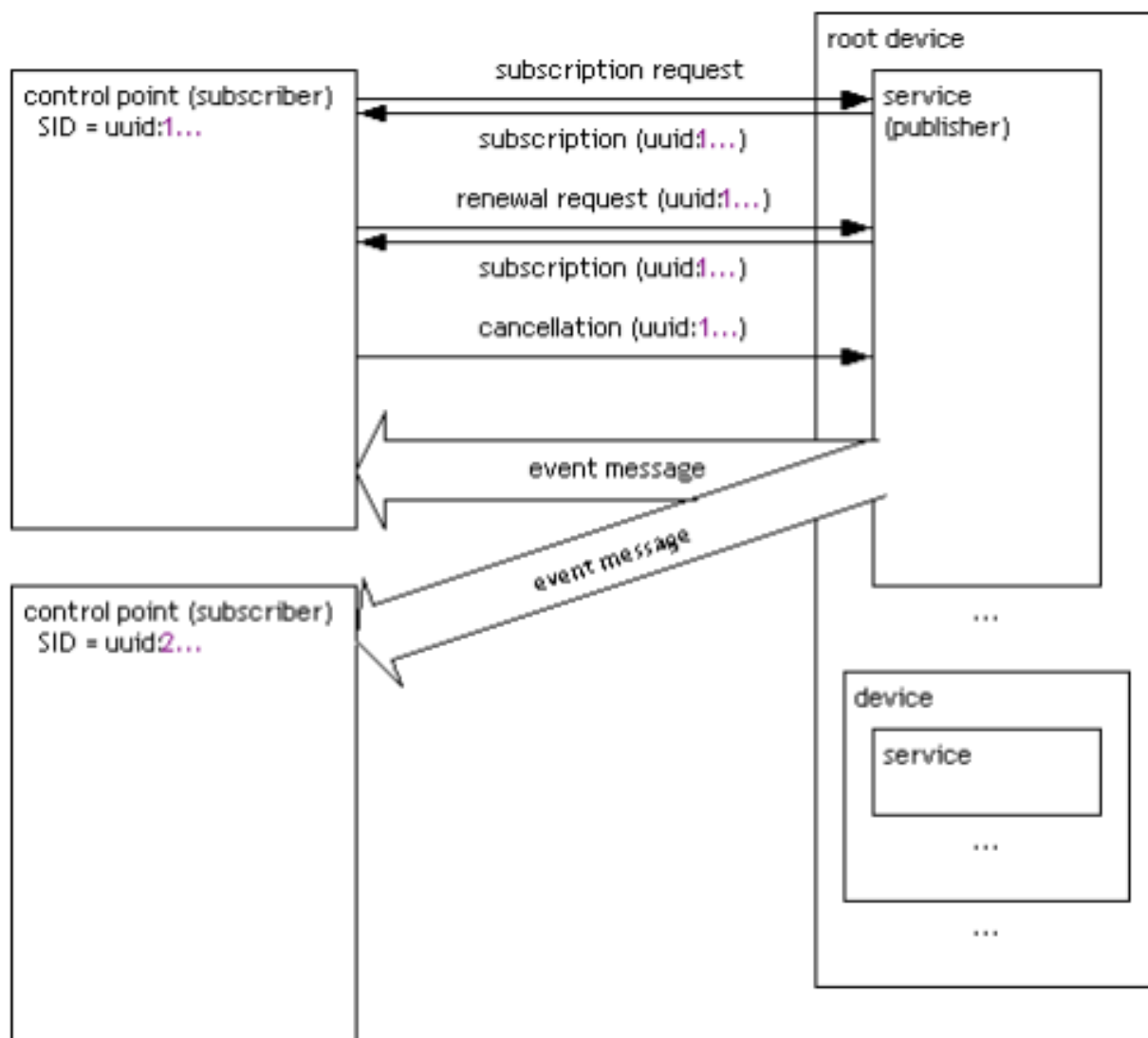
Punkt kontrolny w celu otrzymywania subskrypcji wysyła zgłoszenie subskrypcji (ang. *subscription message*). Jeżeli zgłoszenie jest zaakceptowane wydawca zwraca potwierdzenie w postaci czasu ważności subskrypcji. Subskrypcja musi być odnawiana przez subskrybenta poprzez wysłanie kolejnych zgłoszeń. Punkt kontrolny, który otrzymuje informacje o zmianie stanu zmiennych może w każdej chwili zrezygnować z ich prenumeraty poprzez wysłanie wiadomości unieważniającej (ang. *cancellation message*) do wydawcy. [12]

Do wydawcy należy zadanie wysyłania informacji o zdarzeniach. Komunikat zdarzenia (ang. *event message*) zawiera jeden lub więcej zmiennych, wraz z ich aktualnymi wartościami i jest zapisany w notacji języka XML (ang. *Extensible Markup Language*). Pierwsza wiadomość wysłana do subskrybenta jest nazywana wiadomością inicjującą (ang. *initial event message*) i zawiera wszystkie zmienne (wraz z ich wartościami). Wydawca rozsyła zawsze wszystkie zmienione wersje do każdego punktu kontrolnego, który zgłosił chęć otrzymywania powiadomień (nie ma możliwości przesyłania jedynie niektórych zmiennych). [12]

Zmienne, których wartości są zbyt duże lub których wartości ulega bardzo często zmianie, przez co ich wysłanie w komunikacie zdarzenia jest bezcelowe lub niemożliwe, wydawca może oznaczyć jako tagiem *non-evented*, czyli jako niezgłaszane w rozsyłanych pakietach. [1] Rozmiar (wielkość) zmiennych jest zgodna ze schematem dokumentu XML [15].

Przykładem zdarzeń może być zmiana poziomu głośności urządzenia UPnP, które to zdarzenie jest wysyłane do zainteresowanego punktu kontrolnego. Innym przykładem, dotyczącym multimediiów, jest dodanie pliku, bądź plików multimedialnych i wysłanie powiadomienia o tym fakcie do subskrybentów. Dzięki temu punkty kontrolne mogą wykonać natychmiastową aktualizację listy plików w sposób wygodny i nie wymagający interwencji użytkownika.

Rys. 1.7: Komunikacja pomiędzy punktem kontrolnym, a usługą w przypadku zapisywania do subskrypcji

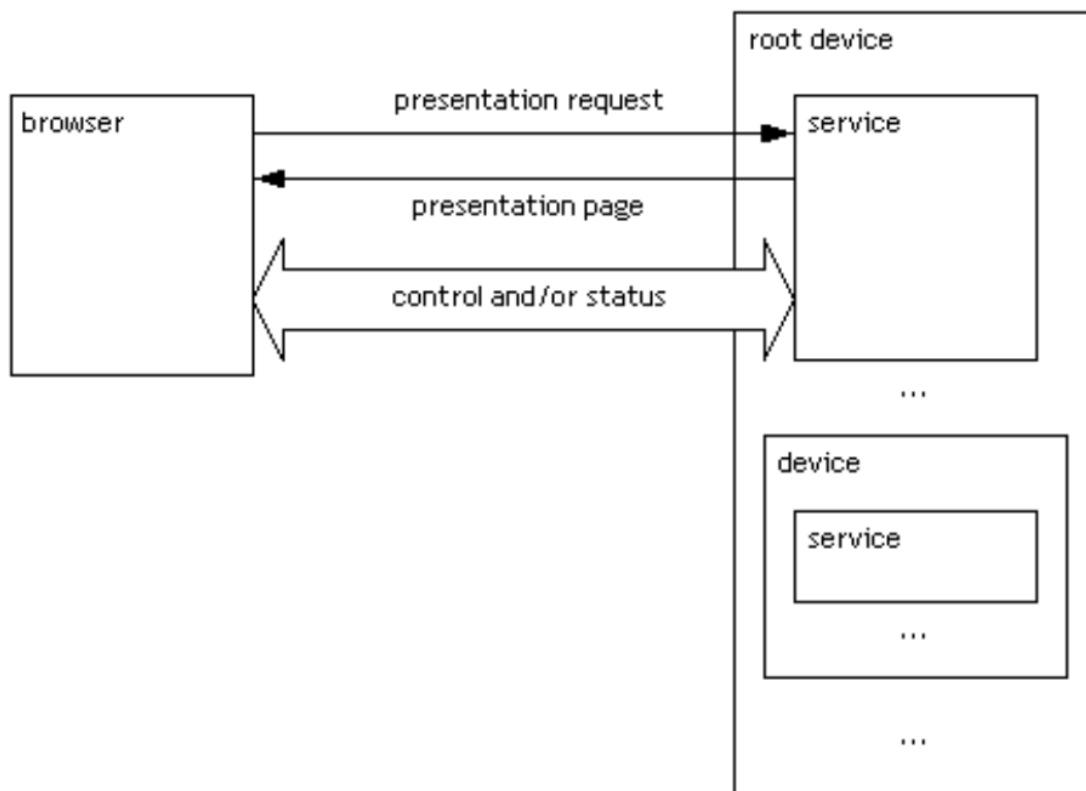


1.2.6 Prezentacja

Warstwa prezentacji jest ostatnią w stosie protokołu UPnP. Wyświetla ona interfejs użytkownika w postaci strony HTML, który umożliwia kontrolę i sprawdzanie statusu urządzenia oraz usług wbudowanych w to urządzenie. Dostęp do interfejsu użytkownika urządzenia umożliwia adres URL, dostarczany wraz z opisem urządzenia (warstwa opisu) 1.2.3. Dostarczanie prezentacji nie jest obowiązkowe i wiele urządzeń takiego interfejsu nie zapewnia. Uzyskanie opisywanej strony HTML urządzenia odbywa się zgodnie ze standardem HTTP (wykorzystując metodę HTTP GET).[1]

Wygląd prezentacji nie jest narzucony w żaden sposób przez zrzeczenie UPnP Forum i jest zależna w pełni od preferencji producenta urządzenia. Wymanagowane jest jednakże, aby strona była zgodna ze standardem HTML i powinna być w wersji 3.0 lub późniejszej.[1]

Często do kontroli urządzenia poprzez warstwę prezentacji wykorzystywane są pozostałe warstwy stosu UPnP. Jednakże nie każdy interfejs UPnP umożliwia modyfikację wszystkich danych (rozdział dotyczący warstwy zdarzeń 1.2.5), co zmusza programistów do tworzenia kilku interfejsu kontroli usług UPnP.[1]

Rys. 1.8: Żądanie strony prezentacji od urządzenia UPnP przez punkt kontrolny

1.3 Klasy urządzeń multimedialnych UPnP

Specyfikacja UPnP definiuje kilka rodzajów urządzeń, które dotyczą protokołu UPnP w sieci IP. W niniejszym rozdziale zostaną opisane rodzaje spotykanych urządzeń UPnP ze szczególnym ukierunkowaniem na rodzaje przeznaczone do współpracy z materiałami multimedialnymi. Urządzenia te są zgodne z jedną z kategorii opisywanego protokołu nazywanego UPnP AV (od ang. *Audio / Video*).

Organizacja DLNA (ang. *Digital Living Network Alliance*) zaczerpnęła multimedialne rodzaje urządzeń ze specyfikacji UPnP oraz dodała elementy wyróżniające DLNA (np drukarkę multimedialną). W niniejszym rozdziale zostaną przedstawione zarówno nazwy urządzeń DLNA, jak i multimedialnych urządzeń UPnP AV.

Jak zostało opisane w poprzednim rozdziale wyróżniamy trzy główne rodzaje urządzeń:

- punkty kontrolne,
- usługi,
- urządzenia.

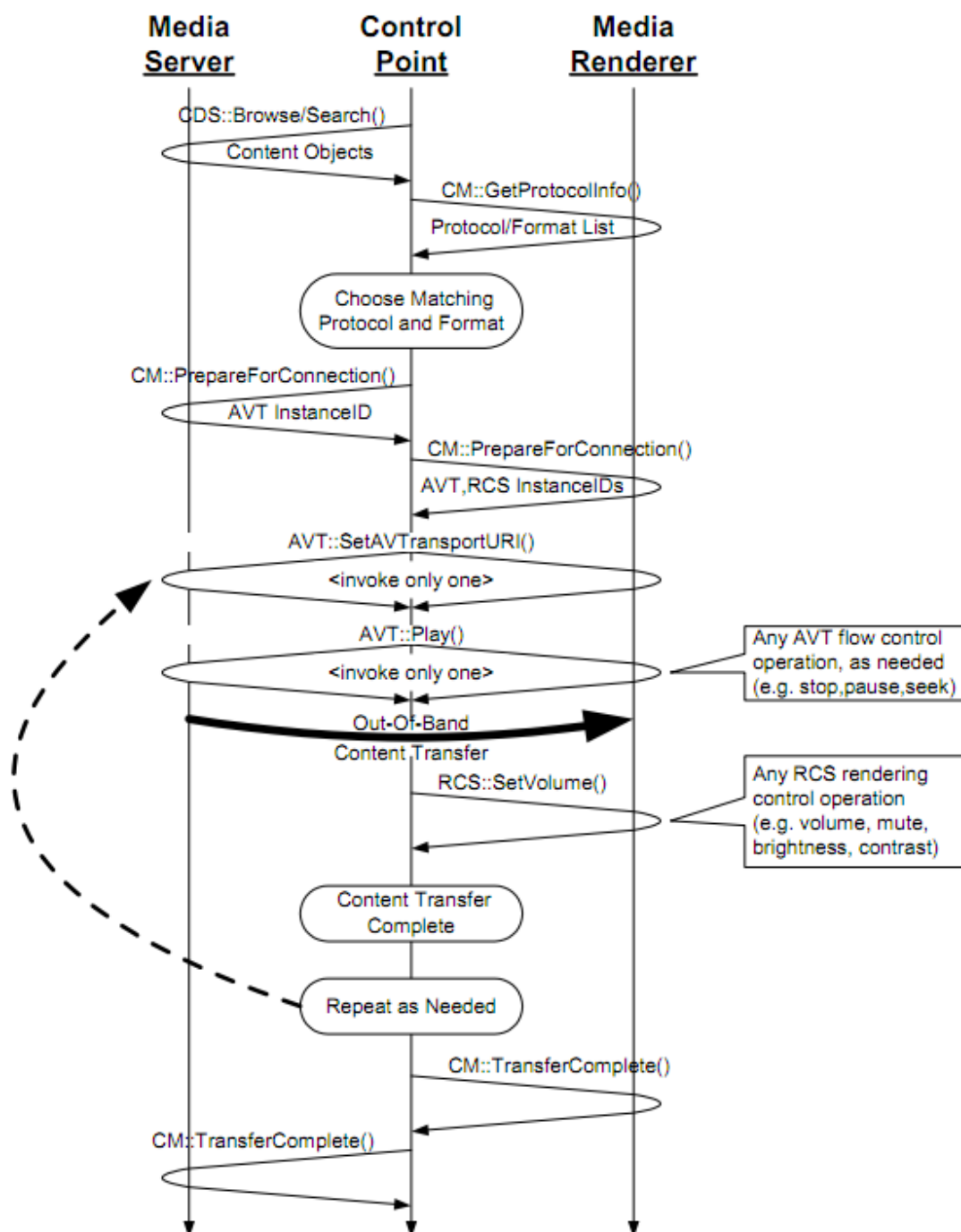
Urządzenia multimedialne podzielone zostały na następujące kategorie:

- serwer multimedialny (ang. *Media Server*) / cyfrowy serwer multimedialny DMS (ang. *Digital Media Server*),
- * / cyfrowy odtwarzacz multimedialny DMP (ang. *Digital Media Player*),
- serwer renderujący (ang. *Media Renderer*) / cyfrowy serwer renderujący DMR (ang. *Digital Media Renderer*),

- punkt kontrolny / cyfrowy kontroler multimediiów DMC (ang. *Digital Media Controller*),
- drukarka multimediiów DMP_r (ang. *Digital Media Printer*).

W powyższej liście pierwsza nazwa jest nazwą nadaną przez organizację UPnP, druga przez organizację DLNA. W przypadku, gdy odpowiednik, w którymś z protokołów nie występuje, na miejscu nazwy usługi lub urządzenia znajduje się znak myślnika.

Rys. 1.9: Schemat odtwarzania w UPnP AV

Playback: General Interaction Diagram**1.3.1 Serwer multimedialny**

Serwer multimedii w protokołach UPnP AV oraz DLNA jest niezbędnym elementem sieci. Serwer udostępnia stworzone, pobrane, nagrane pliki multimedialne. Jest to jego główne zadanie. W protokole DLNA jego rolą jest również zapewnienie ochrony treści (praw autorskich). Serwer

multimediów UPnP AV nie zapewnia ochrony treści. Serwer multimedialny jest to zazwyczaj oprogramowanie zainstalowane na urządzeniach podłączonych do sieci Ethernet, takich jak:

- komputer typu PC,
- nagrywarka cyfrowa,
- smartfon,
- kamera cyfrowa,
- odtwarzacz BluRay / DVD.

Na urządzenie lub oprogramowanie serwera UPnP składają się następujące usługi:

- usługa zarządzania treścią (ang. *Content Directory Service*),
- usługa menedżera połączeń (ang. *ConnectionManager Service*),
- opcjonalna: usługa transportu AV (ang. *AVTransport Service*).

Pierwsza z usług, dotycząca zarządzania treścią umożliwia punktom kontrolnym przeglądanie zasobów udostępnianych przez serwer. Jedną z metod jest przeglądanie - *ContentDirectory::Browse()*, która pozwala uzyskać informacje o pliku wraz z jego metadanymi - nazwa, artysta itd. W metadanych mogą również być zawarte informacje o kodowaniu, transporcie, dzięki którym punkt kontrolny potrafi określić czy dany odtwarzacz ma możliwość odtworzenia zawartości. Druga z usług wykorzystywana jest do zarządzania ustanowionych już połączeń. Ostatnia z opisywanych usług, służy do zarządzania odtwarzanym materiałem. Umożliwia wydawanie poleceń takich jak: odtwarzaj, pauza, stop, przewiń wprzód, przewiń w tył itd.

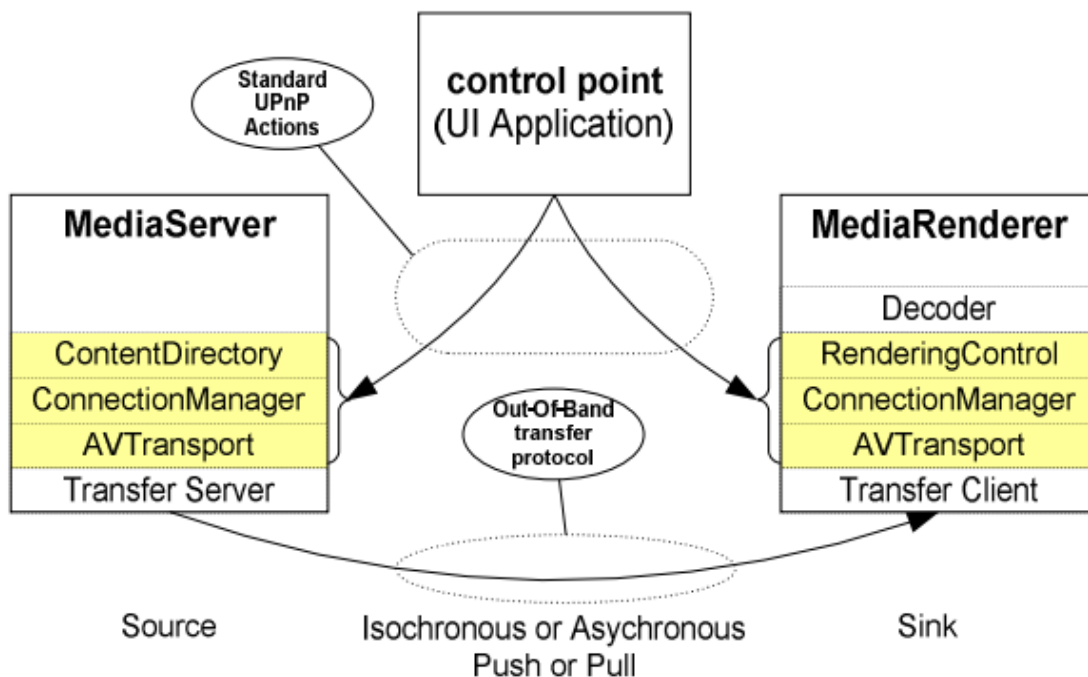
Wszystkie powyżej opisane usługi „wewnętrzne” są również integralną częścią specyfikacji DLNA (ich nazwy w większości publikacji pozostały takie same).[20]

1.3.2 Odtwarzacz multimediów oraz odtwarzacz renderujący

W przypadku opisywanego oprogramowania różnica, między standardami UPnP, a DLNA jest najbardziej widoczna. Zróznicowanie te polega głównie na rozdzieleniu ogólnie pojętego zadania odtwarzania materiałów multimedialnych na mniejsze, pośrednie zadania.

W przypadku UPnP AV w odtwarzaniu treści multimedialnych uczestniczy punkt kontrolny, który wyszukuje materiały udostępniane przez serwer multimediów oraz zleca ich odtworzenie. Polecenie odtworzenia może zostać wykonane w postaci dwóch metod. Metodą *push*, w której serwer multimediów wysyła treść do wybranego przez punkt kontrolny odtwarzacz renderujący, bądź też metodą *pull*, w której odtwarzacz renderujący (ang. *MediaRenderer*) pobiera dane z serwera multimediów, renderuje je i/lub odtwarza (może je renderować i wysyłać do odtwarzania do innego odtwarzacza). Uproszczony schemat opisanej powyżej procedury znajduje się na rysunku 1.10.

Rys. 1.10: Architektura odtwarzania UPnP AV



W przypadku architektury DLNA wyróżniamy dwie usługi potrafiące odtwarzać materiały:

- odtwarzacz multimedialny (ang. *Digital Media Player*),
- odtwarzacz renderujący (ang. *Digital Media Renderer*).

Pierwszy z nich służy do wyszukiwania i renderowania treści multimedialnych. Natomiast zadaniem drugiego jest renderowanie oraz odtwarzanie zadań zleconych przez kontroler multimediiów. Różnica między pierwszym, a drugim jest taka, że odtwarzaczowi renderującemu trzeba zlecić zadanie odtwarzanie, natomiast odtwarzacz multimedialny może podjąć taką decyzję „samodzielnie”.

Urządzeniami tego typu może być:

- telewizor,
- odtwarzacz DVD/BluRay,
- amplituner,
- smartfon,
- komputer PC.

Podobnie, jak w przypadku serwera UPnP, odtwarzacz posiada integralne usługi, które musi, bądź powinien mieć zaimplementowane:

- usługa kontroli renderowania,
- usługa menedżera połączeń (ang. *ConnectionManager Service*),
- opcjonalna: usługa transportu AV (ang. *AVTransport Service*).

Dwie ostatnie usługi komunikują się ze swoimi odpowiednikami w implementacji serwera multimedialnego (ich funkcje się uzupełniają). Usługa kontroli renderowania pozwala punktowi kontrolnemu sterować sposobem przetwarzania treści przez odtwarzacz renderujący. Oznacza to zmianę takich funkcji jak: jasność, kontrast, głośność, wyciszenie itd. Usługa kontroli renderowania może obsługiwać wiele odtwarzaczy renderujących, co umożliwia jednoczesną zmianę ustawień w wielu urządzeniach UPnP.[20]

1.3.3 Punkt kontrolny oraz kontroler multimedialny

W przypadku UPnP A/V nie wyspecyfikowano oddzielnej usługi / urządzenia do kontroli odtwarzacza. Taką rolę spełnia punkt kontrolny w obrębie całej specyfikacji UPnP. Punkt kontrolny w UPnP koordynuje działania serwera multimediiów i odtwarzacza. Zazwyczaj wysyłanie poleceń jest efektem interakcji z użytkownikiem poprzez interfejs użytkownika. Punktu kontrolnego, w opisywanej roli, specyfikacja UPnP AV nie określa jako pełnoprawnego urządzenia UPnP (nie dostarcza żadnych usług UPnP oraz nie jest widoczny jako urządzenie UPnP w sieci), lecz jest niezbędnym pośrednikiem, między użytkownikiem, a urządzeniami UPnP.

Specyfikacja DLNA wyróżnia punkt kontrolny do zarządzania odtwarzaczem multimedialnym, bądź renderującym. Kontroler multimedialny (ang. *Digital Media Controller*) odtwarza zawartość, znaną w udostępnianych zasobach serwera multimedialnego lub zarządza odtwarzaną zawartością odtwarzacza renderującego. Kontrolerami multimedialnymi zazwyczaj są:

- smartfony,
- tablety,
- itd.

Uproszczony sposób działania algorytmu punktu kontrolnego, współpracującego z pojedynczym serwerem multimedialnych oraz serwerem renderującym wygląda następująco:

1. Wykrywanie urządzeń Audio / Video (przy pomocy mechanizmów UPnP).
2. Zlokalizowanie pożądanej treści, przy pomocy metod serwera UPnP, takich jak *ContentDirectory::Browse()* oraz *ContentDirectory::Search()*, które umożliwiają odnalezienie zasobu.
3. Pobierz wspierane, przez odtwarzacz renderujący, formaty danych i protokoły (poprzez metodę *ConnectionManager::GetProtocolInfo()*).
4. Sprawdź możliwości odtworzenia materiału przez pożądany odtwarzacz, wyznacz najlepszy sposób transmisji oraz format danych.
5. Skonfiguruj serwer oraz odtwarzacz renderujący - wykorzystywane są takie polecenia jak: *ConnectionManager::PrepareForConnection()*, *AVTransport::Play()*, *AVTransport::Stop()*, *AVTransport::Pause()*, *AVTransport::Seek()* itd.
6. Wyznacz zawartość do przesłania.
7. Rozpocznij przesyłanie wybranej treści.
8. Możliwe wywoływanie działań na odtwarzaczu (np. pauza).
9. Wybierz opcję: powtórz odtworzenie lub kolejny materiał.
10. Po zakończeniu działania zamknij połączenia z serwerem multimediiów.

1.3.4 Odmiany mobilne

Specyfikacja DLNA dla większości typów urządzeń (wyjątkiem jest drukarka *DMP*) przewiduje możliwość ich odmiany mobilnej. Spełniają one te same funkcje co typy główne. Rozróżnienie dużej ilości typów zostało wprowadzone dla ułatwionej orientacji konsumentów, pośród różnych produktów.[21]

Poza powtarzającymi się typami wyróżniamy dwa, które istnieją jedynie w wersji mobilnej:

- menedżer wysyłania multimediiów (ang. *Mobile Digital Media Uploader M-DMU*),
- menedżer pobierania multimediiów (ang. *Mobile Digital Media Downloader M-DMD*).[21]

Pierwszy typ służy do wysyłania (całości) materiału do serwera multimediiów (mobilnej wersji bądź standardowej) poprzez sieć bezprzewodową. Przykładem takich urządzeń mogą być: cyfrowy

aparat fotograficzny, kamera video, smartfon. Menedżer pobierania multimediiów, również jest urządzeniem działającym w obrębie sieci bezprzewodowej, które pobiera materiały udostępniane przez serwer DLNA (niezależnie od wersji mobilnej, bądź standardowej). Menedżerem pobierania może być smartfon lub przenośny odtwarzacz muzyki.[21]

Odmiany mobilne urządzeń i usług nie są wyróżnione w dokumentacji UPnP.

Multimedia w UPnP AV/DLNA

Specyfikacja UPnP AV została stworzona do umożliwienia oraz ustandaryzowania sposobu komunikacji ze sobą urządzeń multimedialnych w obrębie sieci lokalnej oraz zapewnienie współpracy między produktami różnych dostawców. Cel ten w specyfikacji UPnP AV został w dużej mierze osiągnięty. Specyfikacja UPnP AV pozostawia jednak swobodę w możliwościach obsługiwanych formatów danych. Oznacza to, że pomimo komunikowania się ze sobą różnych urządzeń, cel dla użytkownika końcowego, czyli odtworzenie treści, nie zawsze może być spełniony. Liberalność w kwestii wspieranych formatów zadziałał w tym przypadku na niekorzyść specyfikacji UPnP AV.

Z powyżej opisanego powodu producenci urządzeń powołali konsorcjum DLNA. W specyfikacji DLNA autorzy skupili się w również na formatach odtwarzanych plików. Natomiast ze specyfikacji UPnP zapożyczyli między innymi w prawidłowy sposób działającą komunikację między urządzeniami. Jako format definiuje się możliwy sposób zakodowania pliku.

Certyfikacja DLNA jest stosowana przez większość producentów urządzeń multimedialnych, a spośród w chwili obecnej najprawdopodobniej najpopularniejszego przeznaczenia DLNA (odtwarzanie filmów), większość producentów nowoczesnych telewizorów wyposaża je w oprogramowanie odtwarzacza DLNA.

Zgodnie ze standardem UPnP AV oraz DLNA plik multimedialny można zaklasyfikować do jednego z następujących formatów:

- audio (ang. *audio*),
- obraz (ang. *image*),
- audio-wideo (ang. *AV* / *audio-video*).

W niniejszym rozdziale zostaną opisane obsługiwane typy multimediiów, ich klasyfikacje w opisanych specyfikacjach oraz formaty kodowania plików, przechowywania oraz sposób ich obsługi.

Typy plików (lub ich kodowania) konieczne do obsługi oraz opcjonalne, zdefiniowane przez DLNA, znajdują się w tabelach 2.1, . Objaśnienia do niniejszej tabeli znajdują się w rozdziałach poświęconym poszczególnym klasom multimediiów.

Tabela 2.1: Typy obsługiwanych plików w DLNA - standardowe urządzenia[23]

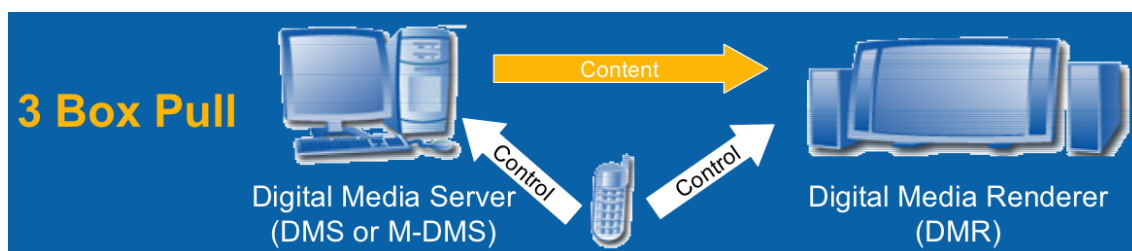
Klasa multimediiów	Wymagane typy plików	Opcjonalne typy plików
Obraz (ang. <i>Image</i>)	JPEG	PNG, GIF, TIFF
Audio	LPCM	AC3, AAC, MP3, WMA9, ATRAC3plus
Wideo (ang. <i>Video</i>)	MPEG2	MPEG1, MPEG4, WMV9

Tabela 2.2: Typy obsługiwanych plików w DLNA - urządzenia mobilne[23]

Klasa multimediów	Wymagane typy plików	Opcjonalne typy plików
Obraz (ang. <i>Image</i>)	JPEG	PNG, GIF, TIFF
Audio	MP3, MPEG4 AAC LC	MPEG4 (HE AAC, AAC LTP, BSAC), AMR, ATRAC3plus, G.726, WMA, LPCM
Wideo (ang. <i>Video</i>)	MPEG4 AVC (AAC LC)	VC1, H.263, MPEG4, MPEG2, MPEG4 AVC (np. BSAC)

2.1 Odtwarzanie multimediów

W odtwarzaniu multimediów w DLNA (analogicznie w UPnP AV) muszą brać udział minimum dwa urządzenia. Jedno to serwer multimediów udostępniający materiał, drugie to odtwarzacz DLNA (rysunek 2.1). Istnieje również schemat w którym udział bierze więcej urządzeń, a trzecim urządzeniem jest kontroler multimedialny pełniący rolę „pilota” zarządzającego, co ma być odtwarzane oraz zarządzać odtwarzaniem (2.2).

Rys. 2.1: Uproszczony schemat komunikacji 2 urządzeń**Rys. 2.2:** Uproszczony schemat komunikacji 3 urządzeń

Pierwsze dwa kroki w celu odtwarzania materiału to podłączenie urządzenia do sieci:

1. Urządzenie dołącza do sieci lokalnej (pobiera adres z serwera DHCP lub autorytatywnie przydziela sobie adres IP - metoda Auto-IP),
2. Urządzenie wysyła informacje o sobie z wykorzystaniem protokołu SSDP na adres 239.255.255.255:1900.

W dalszej kolejności następuje prawidłowa procedura odtwarzania:

1. Punkt kontrolny (w notacji DLNA - odtwarzacz multimedialny lub kontroler multimedialny) wysyła żądanie opisu urządzenia i usługi (usług) do serwera multimedialnego.
2. Serwer multimedialny odsyła opis urządzenia i usługi (usług) w postaci pliku XML.

3. Punkt kontrolny za pomocą komunikat SOAP wysyła do serwera akcje, którą chce wykonać. W przypadku odtwarzania mogą być to komunikaty przeglądania multimedialnych (ang. *browse*), ich wyszukiwania (ang. *search*). Ilość komunikatów jest zależna od drzewa wyszukiwania pożądanego pliku przez użytkownika.
4. Wyszukiwanie kończy się wyborem pliku z wykorzystaniem metod protokołu HTTP np. HTTP GET.

Odtwarzacz multimedialny ma możliwość, działania w typowy sposób dla tradycyjnego odtwarzacza muzyki lub video/DVD. Może przewijać materiał w przód, w tył. Przewijanie jest wykonywane przez podawanie pożądanego bajtów pliku (np. bajt 50 000 - 150 000) przez odtwarzacz multimedialny. Wykonywanie przewijania i skoków w filmie jest możliwe tylko wtedy gdy serwer multimedialny udostępnia rozmiar pliku lub długość pliku (niektóre odtwarzacze pożądamy jednostek w minutach dla długości muzyki czy filmu, pomimo, że prawidłowo wykorzystują pobieranie tzn. pobierają określoną ilość bajtów).

Odtwarzacz multimedialny zgodnie z przytaczanymi specyfikacjami mają możliwość układania list odtwarzania (ang. *playlist*), jednakże autor nie odnalazł odtwarzacza, który umożliwiałby jej utworzenie.

2.2 Klasy multimedialnych UPnP

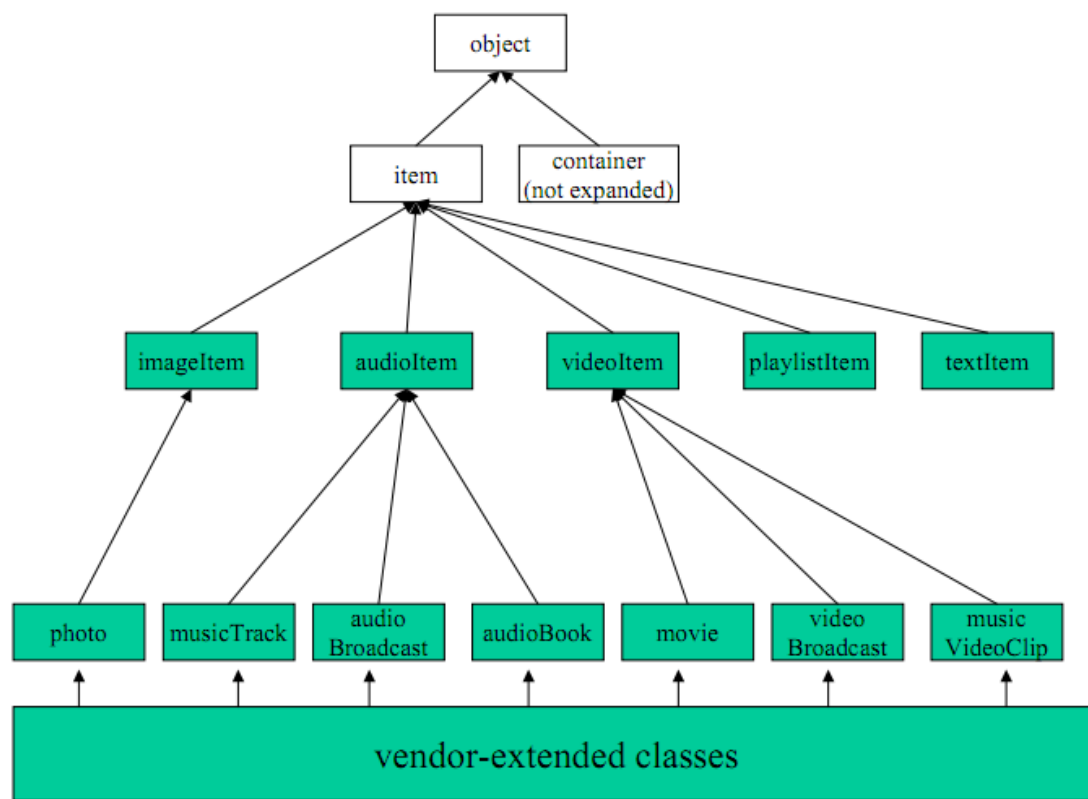
Klasą główną urządzeń UPnP jest *item*, w którym są zawarte następujące podtypy:

- rysunek (*imageItem*),
 - obraz (*photo*),
- audio (*audioItem*)
 - utwór muzyczny (*musicTrack*),
 - nadawanie materiału audio (*audioBroadcast*),
 - audiobook (*audiobook*),
- wideo (*videoItem*),
 - film (*movie*),
 - nadawanie materiału wideo (*videoBroadcast*),
 - teledysk (*musicVideoClip*),
- lista odtwarzania (*playlistItem*),
- tekst (*textItem*).[22]

Każdy z powyższych typów, może posiadać podtypy określone przez wystawcę oprogramowania. Schemat typów został dodatkowo przedstawiony na rysunku nr 2.3.

Rys. 2.3: Schemat typu *item* wraz ze standardowymi podtypami

upnp_av_contentdirectory_service:2



Poza typowymi *item* wyróżniamy również tak zwane kontenery (ang. *containers*). Reprezentują one kolekcje obiektów (fizyczną, bądź logiczną np. posortowaną według typów filmów). Dokładny opis obu typów, wraz ze sposobem stosowania, znajduje się w rozdziale implementacyjnym niniejszej pracy magisterskiej.

W przypadku DLNA zrezygnowano w materiałach z przyrostka *item* i pozostały nazwy takie jak: *audio*, *AV*, *image*. Jednakże w implementacji DLNA (w postaci XML) pozostawiono nazwy, takie jak w UPnP AV, dlatego w niniejszej pracy autor używa nazw pochodzących z UPnP AV. Wyjątek stanowi sytuacja, gdy nazwa została zdefiniowana jedynie w specyfikacji DLNA, a w specyfikacji autorstwa UPnP Forum jej odpowiednik nie występuje.

Typ pliku jest określany poprzez opcję *res* (ang. *resource*).

2.2.1 Klasa *object* oraz klasa *item*

Wyróżniamy dwie klasy główne: *object* i *item*. Pierwsza z nich to klasa główna, w której znajduje się cała udostępniana odtwarzaczowi UPnP lub DLNA treść multimedialna (ang. *ContentDirectory*). Klasa *object* określa właściwości, które są wspólne dla każdego z udostępnianych plików, jak i dla ustrukturyzowanej logicznej kolekcji materiałów. Niniejsza lista przedstawia wymagane właściwości dostarczane wraz z obiektem:

- @id - unikalny identyfikator klasy głównej,
- @parentID - identyfikator klasy nadrzędnej,
- @restricted - ...???
- dc:title - nazwa obiektu,

- upnp:class - klasa zawartości zgodna z UPnP[22].

Dodatkowymi właściwościami opisywanej klasy są:

- dc:creator - ...?
- res - URL ????
- upnp:writeStatus - ????????????[22]

Klasa *item* jest typem określającym każdy indywidualny materiał multimedialny np. zdjęcie w albumie. Obiekt ten nie może zawierać w sobie innych obiektów (nie może być „katalogiem”). Klasa dziedziczy wszystkie opcje po klasie głównej *object* oraz wprowadza następujące właściwości (wszystkie są opcjonalne):

- @refID - odniesienie do pliku głównego (w przypadku gdy plik znajduje się w logicznym kontenerze). Jeżeli udostępniany zasób jest odniesieniem, wówczas pole opisywana opcja jest obowiązkowa,
- upnp:bookmarkID -?

2.2.2 Klasa kontener

Klasa kontener (ang. *container*) dziedziczy po typie głównym *object* i reprezentuje kolekcję innych obiektów (innych kontenerów lub innych podtypów dziedziczącej po klasie *object*). Nie określa ona żadnych obowiązkowych właściwości, które trzeba dostarczyć drugiej stronie komunikacji. Opcjami dodatkowymi są natomiast pola:

- @childCount - liczba obiektów, które znajdują się w kontenerze,
- upnp:CreateClass -?
- upnp:SearchClass -?
- @searchable - wartość 0 lub 1???????
- @neverPlayable - ????????????

2.2.3 Klasa obraz

Typ obraz (ang. *imageItem*) wywodzi się z klasy głównej *item* i dziedziczy od niej wymagane opcje. Ponadto typ obraz powinien zawierać następujące opcje (minimum jedno wystąpienie):

- res - ścieżka do obrazu
 - res@resolution - rozdzielczość obrazu,
- upnp:longDescription - długi opis obrazu,
- upnp:storageMedium - rodzaj pamięci,
- upnp:rating - ocena,
- dc:description - opis obrazu,
- dc:publisher - wydawca,
- dc:date - data powstania obrazu,
- dc:rights - prawa do obrazu (???) [22]

Gdy przedstawianym obrazem jest zdjęcie może ono być podtypu zdjęcie *photo:imageItem*, które dziedziczy wszystkie właściwości typ obraz oraz umożliwia układanie zdjęć w albumy, dzięki opcjonalnej właściwości:

- *upnp:album* - nazwa albumu.

Obraz lub zdjęcie może być udostępnione w kilku rozmiarach oraz według kilku typów. Zdjęcie może być udostępnione w następujących rozmiarach:

- SM (ang. *small*) - małe,
- MED (ang. *medium*) - średnie,
- LRG (ang. *large*) - wielkie,
- TN (ang. *thumbnail*) - miniatura.

Obsługiwany obowiązkowo typ grafik to JPEG_SM. Opcjonalnymi typami, które może obsługiwać odtwarzacz są JPEG_MED, JPEG_LRG, JPEG_TN, PNG_TN. Ponadto miniaturami obrazów, urządzenie może udostępniać ikony (np. ikona z logiem producenta oprogramowania). Ikony mogą być formatowane zgodnie z następującymi rozmiarami: PNG_SM_ICO, PNG_LRG_ICO, JPEG_SM_ICO, JPEG_LRG_ICO. W przypadku ikon, jeżeli urządzenie wspiera ich udostępnianie, wówczas musi udostępniać ich wszystkie rozmiary.

Na podstawie dokumentacji DLNA autor opracował tabelę 2.3 zestawiającą formaty plików oraz ich właściwości. Tabela zawiera standardowe formaty danych i nie uwzględnia formatów opcjonalnych (takich jak PNG czy BMP), które mogą być obsługiwane przez urządzenie. Jednakże bez względu na typ pliku, właściwości dotyczące rozmiaru (na podstawie którego klasyfikowana jest nazwa skrócona obrazu np. MED) muszą być uwzględnione, a obraz udostępniony z odpowiednią nazwą profilu (np. PNG_MED).

Obrazy udostępniane są również w postaci miniatur. Miniatury mogą odnosić się nie tylko do obrazów i zdjęć, lecz do pozostałej udostępnianej zawartości, czyli muzyki i filmów. Każdy album muzyczny może mieć miniaturę z okładką albumu, zaś film miniaturę z plakatem, wybraną sceną lub inny dowolny obraz.

Tabela 2.3: Tabela porównawcza standardowych formatów obrazów w technologii DLNA

Typ MIME	Profil*	Właściwości
image/jpeg	JPEG_SM	maksymalny rozmiar**: 640 x 480
	JPEG_MED	paleta kolorów: niekalibrowane lub sRGB
	JPEG_LRG	maksymalny rozmiar: 1024 x 768
	JPEG_TN	maksymalny rozmiar: 4096 x 4096
	JPEG_SM_ICO	maksymalny rozmiar: 160 x 160
	JPEG_LRG_ICO	maksymalny rozmiar: 48 x 48
	PNG_TN	maksymalny rozmiar: 120 x 120
image/png	PNG_SM_ICO	maksymalny rozmiar: 160 x 160
	PNG_LRG_ICO	maksymalny rozmiar: 48 x 48
		paleta kolorów: odcienie szarości, 24 bitowa paleta kolorów
		maksymalny rozmiar: 120 x 120

*na profil składa się typ pliku (np. JPEG) oraz rozmiar: SM - mały, MED - średni, LRG - wielki, TN - miniatura, SM_ICO - mała ikona, LRG_ICO - duża ikona

** rozmiar podawany w pikselach

2.2.4 Klasa audio

Typ audio służy do udostępniania plików audio, które są przeznaczone jedynie do słuchania. Oznacza to, że klasa ta nie dotyczy ścieżki dźwiękowej w materiałach wideo, takich jak filmy czy transmisja telewizyjna. Klasą nadrzedną dla audio, jest klasa *item*, od której dziedziczone są wszystkie jej właściwości. Podobnie jak w klasie opisującej obraz, powinna ona zawierać opcję

res wraz z danymi dotyczącymi zasobu. Ponadto opcjonalnymi właściwościami możliwymi do załączenia są:

- *upnp:genre* - gatunek pliku audio,
- *dc:description* - opis pliku,
- *upnp:longDescription* - rozszerzony opis pliku,
- *dc:publisher* - wydawca,
- *dc:language* - język,
- *dc:relation* -?,
- *dc:rights* -?

Klasa audio posiada następujące podtypy, określające rodzaj dostępnego materiału dźwiękowego:

- muzyka (ang. *musicTrack*),
- transmisja audio (ang. *audioBroadcast*),
- książka audio (ang. *audioBook*).

Pierwszy z podtypów jest jednocześnie najbardziej popularnym do udostępniania w klasie audio. Muzykę można klasyfikować w albumy, listy odtwarzania, udostępniać informacje o artyście. Wszystkie te atrybuty są opcjonalne, a ich pełna lista wygląda następująco:

- *upnp:artist* - wykonawca,
- *upnp:album* - album muzyczny,
- *upnp:originalTrackNumber* - oryginalny numer utworu,
- *upnp:playlist* - lista odtwarzania,
- *upnp:storageMedium* - rodzaj pamięci??,
- *dc:contributor* - ????
- *dc:date* - data??????

Transmisja audio daje bardzo dużo możliwości rozszerzenia funkcji odtwarzaczy o wzbogacenie ich o takie opcje jak radio internetowe, dzięki przekazywaniu transmisji na żywo przez serwer. Opisany podtyp można przesyłać wraz z opcją *res* zawierającą dane o zasobie oraz opcjonalnymi atrybutami, takimi jak:

- *upnp:region*
- *upnp:radioCallSign*
- *upnp:radioStationID*
- *upnp:radioBand*
- *upnp:channelNr*
- *upnp:signalStrength*
- *upnp:signalLocked*
- *upnp:tuned*
- *upnp:recordable* .????????????????????

Ostatni z typów dziedziczący po klasie audio, czyli audiobook, służy do określania charakterystyki tego typu książek. Pole z zasobem (*res*) powinno być zawarte w przesłanych danych co najmniej raz. Poza nim mogą zostać dołączone następujące atrybuty:

- *upnp:storageMedium*,

- upnp:producer,
- dc:contributor,
- dc:date .????????

2.2.5 Klasa wideo

Klasa wideo opisuje materiały audiowizualne udostępniane w sieci. Są to materiały, których udostępnianie jest najtrudniejsze, zarówno z perspektywy programisty, jak i użytkownika. Jest to spowodowane bardzo dużą rozległością w sposobach kodowania materiałów oraz wciąż niepełną kompatybilnością wielu urządzeń z wieloma formatami plików. W plikach wideo warto zwrócić uwagę na dwa podstawowe pojęcia: kontener multimedialny oraz format pliku wideo. Pojęcie te występuje również w plikach graficznych i audio, jednakże stopień komplikacji dla sprzętu komputerowego występuje dopiero w plikach zawartych w opisywanej klasie.

W ogólnie rozumianym kontenerze multimedialnym może być przechowywanych wiele strumieni obrazów, audio, wideo, napisów, informacji o strumieniach i innych metadanych. Pozwala on na przechowywanie wszystkich tych danych w jednym pliku[25]. Natomiast jako format wideo definiowany jest sposób kodowania strumienia wideo. Analogiczne definicje dotyczą formatów audio oraz obrazów.

Podstawowy typ wideo (ang. *videoItem*) dziedziczy po klasie głównej *item*. Zawiera minimum jedną informację o zasobie *res*. Ponadto z materiałem mogą być przekazane następujące atrybuty:

- upnp:genre,
- upnp:genre@id,
- upnp:genre@type,
- upnp:longDescription,
- upnp:producer,
- upnp:rating,
- upnp:actor,
- upnp:director,
- dc:description,
- dc:publisher,
- dc:language,
- dc:relation,
- upnp:playbackCount,
- upnp:lastPlaybackTime,
- upnp:lastPlaybackPosition,
- upnp:recordedDayOfWeek,
- upnp:srsRecordScheduleID.????????

Jak jest widoczne na powyższej liście ilość atrybutów możliwych do przesłania jest bardzo duża. Jest to bazowa klasa dla pozostałych typów wideo, którymi są:

- film (*movie:videoItem*),
- transmisja wideo (*videoBroadcast:videoItem*),
- teledysk / klip muzyczny (*musicVideoClip:videoItem*).

Klasa film reprezentuje materiały filmowe i umożliwia załączenie dodatkowych danych odnośnie filmu w wyspecyfikowanych, przez autorów dokumentacji UPnP AV, atrybutach. Na atrybuty składają się:

- *upnp:storageMedium* - nośnik pamięci,
- *upnp:DVDRegionCode* - kod regionalny materiału DVD,
- *upnp:channelName* - nazwa kanału,
- *upnp:scheduleStartTime* - planowany start transmisji,
- *upnp:scheduleEndTime* - planowany koniec transmisji,
- *upnp:programTitle* - tytuł programu (odcinka),
- *upnp:seriesTitle* - tytuł serii,
- *upnp:episodeCount* - liczba odcinków,
- *upnp:episodeNr* - numer odcinka.

Wszystkie z powyższych opcji są możliwe do udostępnienia w postaci atrybutów XML, lecz nie są obowiązkowe.

2.3 Rodzaje multimedialnych

Filmy, a dokładnie ich pełne wsparcie i obsługa, były najciekawszym i zarazem najtrudniejszym tematem do realizacji dla autora niniejszej pracy. Z tego powodu niniejszy podrozdział zostanie głównie poświęcony opisowi, w jaki sposób w erze komputerów przechowywane są tego rodzaju materiały.

Ilość formatów danych multimedialnych jest bardzo duża. Najbardziej pożądanymi formatami są formaty kompresujące dane. Formaty plików zapewniające tzw kompresję bezstratną (formaty *RAW* - surowe) są formatami pozwalającymi zachować najwyższą jakość obrazu lub dźwięku. Przykładami takich formatów są dla plików graficznych - *RAW*, natomiast audio - format *FLAC* (ang. *Free Lossless Codec*). Innym popularnym formatem dźwięku, który jest uważany za bardzo bliski oryginałowi, dzięki pokrywaniu całego zakresu pasma częstotliwości słyszalnych jest kodowanie *PCM*, stosowane bardzo często w formacie *WAV* (<http://pl.wikipedia.org/wiki/PCM>). Spośród plików wideo można wyróżnić formaty danych o niskiej kompresji stratnej. Przykładem takich plików są formaty *MPEG-2* oraz *MPEG-4*. Drugi z nich pozwala na przesyłanie materiału HD i posiadają kompresję w stosunku 4:1 (skąd to wiem). Wykorzystywany jest współcześnie w przesyłaniu cyfrowej telewizji naziemnej oraz częściowo satelitarnej.

W przypadku zdjęć prym wiodzie format *JPEG*, który jest ustandaryzowany przez organizację *ISO/IEC* [26], jest formatem kompresji stratnej. Jednakże kompromis między kompresją, a jakością obrazu (kompresja 10:1) jest na tyle zadowalający, że stał się najpopularniejszym formatem danych graficznych (z pominięciem zastosowań profesjonalnych, gdzie często wykorzystywane są formaty bezstratne, bądź inne specjalistyczne, specyficzne dla oprogramowania obróbki grafiki).

Najpopularniejszym formatem audio dla przechowywania muzyki wśród końcowych użytkowników jest format *MP3* (ang. *MPEG-1/MPEG-2 Audio Layer 3*). Jest standardem *RFC*.... W swojej pierwotnej, a zarazem najpopularniejszej wersji pozwala na zapis muzyki w postaci dźwięku jednokanałowego (mono) lub dwukanałowego (stereo) i oferuje najwyższy stopień kompresji, przy zachowaniu dobrej jakości dźwięku. W przypadku ścieżek dźwiękowych do filmów częściej stosowane ma standard *AC3* [wikipedia]

W przypadku materiałów filmowych trudnym zadaniem jest jednoznaczne określenie najpopularniejszego formatu tego typu danych. W kolejnych podrozdziałach został dokładnie opisany ten rodzaj multimedialnych.

2.3.1 Muzyka

Obrazy

2.3.2 Filmy

Film można rozumieć zgodnie z następującym schematem:

- Kontener filmowy (np. AVI, MKV)
 - Format wideo (np. MPEG-2),
 - Format audio (np. AC3),
 - Napisy,
 - Rozdziały,
 - Metadane opisujące powyższe dane.

Istnieje wiele rodzajów kontenerów. Część z nich ma przeznaczenie do udostępniania multimedialnych w sieci internetowej (Flash), inne są dedykowane dla urządzeń mobilnych (MP4). Najpopularniejszymi kontenerami multimedialnymi są:

- AVI (rozszerzenie AVI),
- Matroska (rozszerzenie MKV),
- Flash (rozszerzenie FLV lub F4V).
- MP4,
- DivX.

Kontenery różnią się metadanymi opisującymi pliki oraz możliwościami we wspieraniu przechowywanych formatów danych. Tabela porównawcza formatów AVI i MKV pokazuje podstawowe różnice między tymi formatami (tabela 2.4).

Materiał wideo w filmach może być kodowany w formatach, takich jak:

- MPEG-1,
- MPEG-2
- MPEG-4 AVC / H.264,
- WMV (ang. *Windows Media Video*),
- RealVideo.

Kontenery filmowe pozwalają przechowywać różne formaty wideo (część z nich widoczna jest powyżej), różne formaty audio (jeden plik może zawierać kilka ścieżek audio w różnych formatach np. MP3 lub AC3). Niektóre umożliwiają przechowywanie w kontenerze plików z napisami (we wspieranych formatach), posiadają możliwości podziału materiału na sekcję lub przechowywania materiału trójwymiarowego 3D. Wszystkie dane co przechowywane jest w danym pliku zawierają się metadanych pliku. Każdy rodzaj konteneru posiada (niestety) zróznicowany sposób tworzenia metadanych. Utrudnia to w znaczący sposób łatwe wsparcie jak ich największej ilości.

Oddzielnym tematem są formaty płyt Blu-Ray, bądź DVD. Formaty te posiadają specjalną strukturę plików oraz schemat katalogów, według których umieszczone są odpowiednie dane. W uproszczeniu można powiedzieć, że nośnik Blu-Ray jest kontenerem multimedialnym, takim jak AVI czy MKV. Podobna sytuacja ma miejsce w przypadku płyt DVD.

Płyta Blu-Ray, zgodna ze standardem posiada drzewo, składające się z dwóch plików „index.bdmv” (zawierający informacje o zawartości katalogów płyty BD), MovieObject.bdmv (zawierający informacje o obiektach filmowych) oraz katalogów: PLAYLIST (lista odtwarzania filmu / rozdziały), CLIPINF (informacje o klipach wideo), STREAM (strumień audio wideo w kontenerze

m2ts - ang. MPEG-2 Transport Stream), AUXDATA (pliki dźwiękowe oraz czcionki dla plików z napisami), BACKUP (katalogów zawierający kopie zapasowe plików index.bdmv oraz MovieObject.bdmv).

Tabela 2.4: Porównanie formatów AVI i MKV

Format	Rozdziały	Napisy	Formaty wideo	Formaty dźwiękowe
AVI	Tak	Nie (poza standardem tak)	MPEG-1, MPEG-2, WMV, RealVideo	MP3, WMA, AAC
MKV	Tak	Tak	MPEG-1, MPEG-2, MPEG-4 AVC, WMV, RealVideo	MP3, WMA, AAC, DTS, PCM, RealAudio

Film może posiadać wiele ścieżek dźwiękowych np. różne wersje językowe lub lektora. Płyty DVD oraz Blu-Ray w standardzie swoim mają możliwość zamieszczania wielu ścieżek dźwiękowych. Podobna sytuacja ma miejsce w przypadku kontenerów danych takich jak AVI oraz Matroska. Oprogramowanie udostępniające materiały multimedialne powinno zapewniać możliwość wyboru ścieżki dźwiękowej. W tym celu konieczne jest poprawne odczytywanie metadanych dostarczanych wraz z kontenerem multimedialnym. Filmy DVD oraz Blu-Ray ponadto, że umożliwiają zmianę ścieżki dźwiękowej, wprowadziły do zestawów kina domowego możliwość wyświetlania napisów w filmach (przedtem napisy były cechą wyróżniającą kina). W przypadku płyt DVD napisy są przechowywane w formacie bitmapowym, co wykracza poza standardy UPnP oraz DLNA (dla napisów), dlatego też nie są one obsługiwane do udostępniania. Płyty Bluray posiadają standard zbliżony do płyt DVD (napisy PGS - ang. *Presentation Graphic Stream*). Łatwiejszym przypadkiem są kontenery plików. Napisy w przypadku tego typu filmów mają postać pliku tekstowego. Plik z napisami może być wbudowany w kontener, a może być udostępniany jako oddzielny plik (zazwyczaj w tym samym folderze i tej samej nazwie co film).

Istnieje wiele standardów napisów do filmów, które mogą być w prosty sposób przechowywane na dysku twardym komputera stacjonarnego. Głównymi standardami spotykanymi powszechnie w Internecie są MicroDVD oraz SubRip. Istnieje wiele formatów, jednakże standard DLNA pomija te cechy filmów i nie specyfikuje jakie rodzaje napisów powinny być obsługiwane oraz w jaki sposób. Informacje o sposobie udostępniania napisów zostały określone jedynie w specyfikacji UPnP AV dla urządzeń renderujących [<http://www.upnp.org/specs/av/UPnP-av-RenderingControl-v3-Service-20101231.pdf>].

Brak jednoznacznych rozwiązań w kwestii napisów do filmów, sprawił, że producenci w wielu przypadkach we własnym zakresie zapewniają ich wsparcie. Odtwarzacze multimedialnych podłączonych przez złącze USB w telewizorach ma zazwyczaj możliwość odtwarzania napisów w wersji tekstowej, w formatach MicroDVD lub SubRIP. Takie samo podejście zostało zastosowane w odtwarzaczach DLNA. Obsługują one jeden z dwóch rodzajów napisów, a w gestii serwera leży, aby w odpowiednim formacie takie napisy udostępnić (lub samodzielnie przekonwertować do obsługiwanego formatu).

2.3.3 Kodowanie filmów

II Implementacja serwera UPnP

Oprogramowanie serwera UPnP AV

Większość współczesnych odtwarzaczy Blu-ray, telewizorów, dekodek i tym podobnych urządzeń RTV, posiada złącze Ethernet (lub moduł *WiFi*) i umożliwia podłączenie urządzenia do sieci lokalnej i Internetu. Takie urządzenia posiadają oprogramowanie umożliwiające wyszukiwać oraz odtwarzać materiały udostępniane przez serwery UPnP AV oraz DLNA. Autor na potrzeby niniejszej pracy stworzył w pełni funkcjonalny serwer UPnP AV współpracujący z urządzeniami UPnP i DLNA. Głównym celem aplikacji było umożliwienie kooperacji dwóch urządzeń ze sobą, w taki sposób, aby odtwarzacz potrafił wykorzystać jak największą możliwą ilość funkcji określoną w specyfikacjach UPnP AV i DLNA.

3.1 Wymagania wstępne

Zdefiniowanie środowiska do implementacji oprogramowania było bardzo ważnym wyborem. Autor do codziennej pracy używa systemu operacyjnego z rodziny Linux, jednakże większość użytkowników komputerów typu PC używa systemu operacyjnego Microsoft Windows. Dlatego też środowisko implementacji ma zapewniać kompatybilność z oboma systemami operacyjnymi. Pierwszym wyborem ze względu na biegłą znajomość języka było środowisko programistyczne języka Java.

W dzisiejszych czasach coraz więcej urządzeń sieci lokalnej nie jest oparta o architekturę PC. Przykładami takich urządzeń mogą być routery lub smartfony. Pierwsze z nich, coraz częściej są wyposażone w złącza USB oraz posiadają załączone oprogramowanie do udostępniania multimedialnych. Z doświadczeń własnych autora, wynika jednak, że zakres możliwości oraz wydajność takiego oprogramowania pozostawia wiele do życzenia. Skutecznie udostępniane są jedynie zdjęcia, natomiast użytkownicy natrafiają na problemy w kwestii odtwarzania filmów oraz muzyki.

W tabeli 3.1 zostały zdefiniowane platformy konieczne do obsługiwanego przez serwer UPnP AV.

Tabela 3.1: Platformy obsługiwane przez serwer UPnP AV

Klasa urządzenia	System operacyjny
Router	DD-WRT
	Tomato
PC	Windows
	Linux

W przypadku środowiska Java wymienione we wcześniejszych akapitach wymagania mogą być z łatwością realizowane. Jednakże obciążenie procesora (system operacyjny Ubuntu 10.10) dla prostej aplikacji UPnP wynosiło około 5%. Analiza materiałów udostępnianych w Internecie oraz doświadczenie z użytkowania aplikacji Java, poddawała w wątpliwość wydajną pracę serwera zaprojektowanego w tym języku na mało wydajnych urządzeniach (np. routery). Po kilku próbach

uruchomienia serwera Tomcat na routerze firmy Linksys E3000, autor podjął decyzję o dalszym szukaniu środowiska do stworzenia aplikacji. Ostateczny wybór padł na język Python do realizacji oprogramowania.

Pozostałe wymagania zostały narzucone przez standardy UPnP AV i DLNA, co prosto można określić jako wymaganie pełnej współpracy z certyfikowanymi urządzeniami DLNA lub UPnP. Głównymi wymaganiami są:

- współpraca z podstawowymi urządzeniami DLNA - telewizorami, odtwarzaczami BluRay itd.,
- odtwarzanie (o ile podstawowy odtwarzacz urządzenia ma taką możliwość) kontenerów muzycznych i filmowych np. plików Matroska - MKV,
- odtwarzanie napisów w filmach i muzyce (np. do karaoke, specjalny tryb serwera), zarówno z plików tekstowych, jak i z kontenerów (w tym przypadku tylko pliki Matroska, MKV),
- przewijanie materiału,
- udostępnianie materiału w pełnej jakości,
- dodawanie / usuwanie udostępnianej zawartości bez konieczności ponownego uruchamiania oprogramowania,
- tworzenie albumów, klasyfikacji treści w logicznych katalogach.

W oprogramowaniu autor jako opcjonalne wymaganie zdefiniował renderowanie filmów do gorszej jakości w locie. W chwili obecnej większość nowoczesnego sprzętu na rynku obsługuje dużą ilość formatów i mimo swoich możliwości to oprogramowanie odtwarzacza DLNA jest przestarzałe lub stworzone z minimalnymi wymaganiami certyfikacji, co powoduje komunikat o niekompatybilności materiału, pomimo, że w rzeczywistości taka sytuacja nie ma miejsca. Z tego powodu autor skupił się podczas tworzenia oprogramowania serwera na ominięciu przeszkód postawionych przez większość producentów i wymuszenie odtwarzania materiałów na każdym urządzeniu.

Nie wszystkie urządzenia posiadają interfejs graficzny użytkownika. Interfejs użytkownika w serwerze UPnP AV/DLNA jest dostarczane wraz z oprogramowaniem serwera www i interfejsem strony internetowej do zarządzania serwerem oraz uruchamiania go.

3.2 Wykorzystane narzędzia

Aplikacja serwera UPnP AV została napisana w języku Python. Jej szkielet został oparty o zmodyfikowaną wersję framework'a *Coherence*, opisanego w rozdziale 3.2.1. Druga aplikacja, zapewniająca zarządzanie serwerem została napisana w języku Python, HTML oraz Javascript z wykorzystaniem popularnego framework'a Django.

Aplikacja jest zarządzana przez narzędzie kontroli wersję Git na darmowym serwerze Github.com

Wykorzystane narzędzia na potrzeby opisywanego zestawu aplikacji to:

- platforma Eclipse 3.6,
 - języki programowania: Python, HTML, Javascript, **Django**, T-SQL,
 - framework'i
 - * *Coherence DLNA*,
 - * *Django*,
 - * *JSONRPC*,
 - * *Storm*.
- GIMP do tworzenia grafiki aplikacji
- odtwarzacze UPnP AV / DLNA do testów
- GIT - do zarządzania wersjami oprogramowania.

3.2.1 Framework'i UPnP

W internecie można odnaleźć bardzo dużo framework'ów, które obsługują UPnP AV oraz DLNA. Przed wyborem języka programowania autor skupił się na poszukiwaniu odpowiedniego środowiska, który wspiera komunikację w sieci multimedialnej, posiada darmową, otwartą do modyfikacji licencję oraz nie obciąża w znaczący sposób systemu operacyjnego. Autor nie odnalazł framework'a, który natywnie wspierałby specyfikację DLNA. Jest to spowodowane prawdopodobnie koniecznością ponoszenia kosztów licencji DLNA. Liczba framework'ów, które wspierają protokół UPnP oraz UPnP AV, pomimo że nie jest duża, to jest w zadowalający sposób zaimplementowana, dzięki czemu autor nie musiał zaprogramować niższych warstw opisywanego protokołu.

Pierwszym, prawdopodobnie najnowszym i najczęściej aktualizowanym jest oprogramowanie *Cling* [27]. Jest to biblioteka, której zadaniem docelowym (jest wciąż w fazie rozwoju i wiele funkcji nie jest jeszcze zaimplementowanych) jest pełne wsparcie protokołu UPnP. Nie implementuje zmian wprowadzonych przez zrzeszenie DLNA. Jest napisana w języku Java, możliwe jest jej wykorzystanie zarówno w oprogramowaniu PC, jak i oprogramowaniu Javy dla smartfonów z systemem operacyjnym Android. Autor odrzucił jej wykorzystanie, gdyż w ówczesnym stadium rozwoju implementowała zbyt małą liczbę koniecznych do komunikacji funkcji oraz niemożliwe byłoby jej wykorzystanie na urządzeniach o bardzo małych możliwościach wydajnościowych (np. routery).

Drugim brany pod uwagę jako podstawę do aplikacji serwera UPnP framework'iem był *CyberLink for Java* [28]. Jest to pakiet oprogramowania umożliwiający twórcom aplikacji komunikację w sieci UPnP. Posiada wiele popularnych dla oprogramowania UPnP przykładów, np. włączanie i wyłączanie żarówki. Niestety w chwili porównywania możliwości framework'ów opisywany projekt nie był rozwijany i jego ostatnia aktualizacja datowana była na 2003 rok. Natomiast w trakcie pisania niniejszej pracy (już po stworzeniu oprogramowania) autor zauważył wznowienie projektu poprzez dodanie wsparcia dla pisania aplikacji UPnP do obsługi urządzeń mobilnych z systemem operacyjnym Android. *CyberLink for Java* jest rozwijany na licencji nie zezwalającej na modyfikacje kodu źródłowego.

Autor rozważał również wykorzystanie bibliotek popularnego w środowisku *Linux* programu MiniUPnP [29] lub MiniDLNA [30]. Niestety oba umożliwiają wykorzystanie we własnym oprogramowaniu bibliotek klienckich, a celem projektu było stworzenie serwera UPnP AV.

Ostatnim opisanym w niniejszym podrozdziale będzie *Coherence DLNA/UPNP Framework for Digital Living* [31]. Pomimo, że ostatni to spełnił najwięcej wymagań stawianych przez autora i dlatego został przez niego wybrany jako szkielet napisanej aplikacji. *Coherence* jest zarówno programem udostępniającym prosty serwer multimedialny, serwer renderujący, posiada wiele wtyczek do popularnych programów (np. Rhythmbox, Totem, Picasa). Jest na bieżąco rozwijany, zaś jego licencja - MIT [32], zezwala na dowolną modyfikację. Jest napisany w języku Python. Dla deweloperów najważniejsze jest, że oprogramowanie zapewnia komunikację (wykrywanie, ogłaszanie urządzeń) oraz posiada „szablony”, to jest przykładowe aplikacje z funkcjonalnością udostępniania albumów, filmów itd. Niestety testy oprogramowania wykazały, że sposób implementacji protokołu nie zapewnia komunikacji z wieloma urządzeniami (w tym mobilnymi). Autor zdecydował się wykorzystać w swoim oprogramowaniu opisywany framework oraz wprowadzić w nim niezbędne modyfikacje, co dzięki liberalnej licencji MIT było w kwestiach prawnych możliwe.

3.3 Opis programu

Zaprojektowana aplikacja do udostępniania materiałów multimedialnych, jest to w rzeczywistości zestaw dwóch niezależnych od siebie programów. Pierwsza aplikacja jest serwerem UPnP AV który komunikuje się bezpośrednio z odtwarzaczami DLNA oraz UPnP AV. Natomiast druga aplikacja służy do zarządzania współdzielonymi zasobami oraz ustawieniami „technicznymi” serwera.

Aplikacja posiada wbudowane wyszukiwanie multimedialnych różnych rodzajów w lokalizacjach, które użytkownik wcześniej zdefiniował. W przypadku podkatalogów stosowane jest przeszukiwanie

rekursywne. Serwer UPnP AV interpretuje zawartość katalogów i określa te pliki, które są filmami, plikami tekstowymi (które są napisami do filmów), muzycznymi oraz zdjęciami. Pliki w odtwarzaczu mogą być przeglądane w katalogach oddających strukturę w systemie plików serwera oraz w kategoriach (wirtualnych katalogach) zdefiniowanych przez użytkownika.

3.4 Funkcje programu

Aplikacja składa się z dwóch modułów:

- serwer UPnP AV,
- aplikacji zarządzającej.

Funkcje pierwszego z modułów to:

- ogłaszanie aktywności dla innych urządzeń UPnP,
- wysyłanie, odpowiadanie na aktywność UPnP,
- wyszukiwanie multimediiów: filmów, plików z napisami, muzyki, obrazów,
- tworzenie biblioteki multimediiów:
 - zgodnie ze strukturą katalogów,
 - według typów plików (sortowanie według typów multimedialnych MIME),
 - według typów zawartości:
 - * Muzyka,
 - * Wideo,
 - * Zdjęcia.
- udostępnianie w językach:
 - angielski,
 - polski,
- udostępnianie informacji o plikach
 - rozdzielczość,
 - rodzaj,
 - rozszerzenie,
 - czas trwania,
 - jakość,
 - data dodania,
 - rozmiar pliku,
 - lokalizacja pliku.

3.4.1 Biblioteka multimediiów

Biblioteka multimediiów jest zbiorem plików udostępnianych przez serwer UPnP AV. W opisywanej aplikacji biblioteka została podzielona na kilka głównych kategorii. Pierwszą z nich jest odwzorowanie zgodnie z położeniem w systemie plików (gdy to możliwe wyświetlana jest tylko nazwa katalogu nadrzędnego, w którym znajduje bądź znajdują się pliki multimedialne). Oznacza to, że przy pomocy odtwarzacza UPnP AV lub DLNA użytkownik może nawigować do przeglądania zawartości serwera i do sekcji „Katalogi” (ang. „*Directories*”). W dalszej kolejności wyświetlane są nazwy katalogów z plikami multimedialnymi.

Kolejnym krokiem nawigacji może być „Typy MIME” (ang. „*Mimetypes*”), w którym wyselekcjonowane materiały są dzielone według swoich typów MIME.

Dalszymi dostępnymi ścieżkami są następujące kategorie:

- Wideo (ang. *Video*),
- Muzyka (ang. *Music*),
- Zdjęcia (ang. *Images*),

Pliki są segregowane przez algorytm zgodnie ze swoją kategorią (zgodnie z typem MIME), a następnie umieszczane w odpowiadającej kategorii bibliotece. Typ MIME w swojej nazwie dla każdej z bibliotek w pliku posiada cechę wspólną. W niniejszym przypadku pliki graficzne posiadają w typie MIME słowo *image*, muzyka słowo *audio*, natomiast materiały wideo, słowo *video*. Upraszcza to w dużym stopniu sposób kategoryzowania plików.

Dla każdego odnalezionego katalogu oraz pliku multimedialnego tworzony jest obiekt *MediaItem*, który przechowuje niezbędne informacje dla serwera UPnP AV. Dzięki odczytaniu informacji o obiekcie możliwy jest zwrócenie odtwarzaczom posiadanych informacji o pliku, katalogu lub katalogu wirtualnym.

Parametry możliwe do zwrócenia dla każdego z wcześniej wymienionych typów są umieszczone w tabeli 3.2

Tabela 3.2: Parametry dla kategorii UPnP

Pobierany parametr Rodzaj	Katalog	Wideo	Audio	Zdjęcia
Ścieżka URL	x	x	x	x
Ilość podkatalogów	x			
Podkatalog	x			
Obiekt <i>MediaItem</i>	x			
Katalog nadrzędny	x	x	x	x
Inne katalogi nadrzędne	x	x	x	x
ID	x	x	x	x
Nazwa	x	x	x	x
Okładka (miniatura)		x	x	x
Typ MIME	x	x	x	x
Rozmiar		x	x	x
Czas trwania		x	x	
Bitrate		x	x	
Kanały audio		x		
Napisy		x		
Rozdzielczość		x		x

Biblioteka multimediiów umożliwia dodawanie nowych kategorii w trakcie działania aplikacji bez konieczności przerywania jej działania. Odpowiada za to parametr zdefiniowany w specyfikacji UPnP - parametr aktualizacji (ang. *updateID*), który jest numerem sekwencyjnym określającym ilość zmian w danej bibliotece. Jeżeli odtwarzacz posiada dane o plikach z nieaktualnym parametrem aktualizacji, wówczas pobiera od nowa listę multimediiów. Aplikacja w podobny sposób reaguje na dodanie i usunięcie pliku (plików) do istniejącego już, w ścieżce wyszukiwania, katalogu.

3.4.2 Miniatury multimediów

Miniatury multimediów nie są obowiązkową treścią, którą musi udostępniać serwer DLNA. Jednakże sposób udostępniania ich nie odbiega znacząco od sposobu udostępniania innych materiałów, dlatego autor uznał, że warto tą funkcję również zaimplementować.

Większość współczesnych systemów operacyjnych tworzy miniatury filmów oraz obrazów na własne potrzeby oraz często pobiera okładki albumów muzycznych z Internetu. Autor zdecydował się w pierwszej kolejności korzystać z miniatur stworzonych przez system operacyjny, zaś w dalszej kolejności z miniatur tworzonych przez aplikację (w przypadku gdy miniatura systemu operacyjnego nie istnieje lub nie spełnia wymogów co do rozmiarów). Miniatura tworzona przez aplikację ma rozmiar 120 x 120 pikseli i jest zapisana w standardzie JPEG z tą samą nazwą pliku jaką posiada główny plik multimedialny, z przedrostkiem TH (od ang. *Thumbnail* - miniatura).

3.4.3 Zmiana opcji

3.5 Budowa wewnętrzna serwera

Serwer multimedialny UPnP AV / DLNA został wykonany w języku Python. Do jego realizacji wykorzystany został framework *Coherence DLNA/UPNP Framework for Digital Living* którego opis znajduje się w rozdziale 3.2.1. **Zestaw oprogramowania został podzielony na dwie niezależne aplikacje, serwer multimedialny UPnP AV oraz serwer zarządzający. Każdy z nich składa się z wielu podmodułów.**

Serwer multimedialny UPnP AV udostępnia swoje funkcje poprzez protokół UPnP w wersji tylko do odczytu, natomiast zarządzanie nim następuje przy pomocy poleceń RPC wysyłanych do wewnętrznego serwera RPC. Polecenia do integralnego serwera RPC wysyłane są z drugiej aplikacji, z serwera zarządzającego. Poprzez zarządzanie rozumiane są takie funkcje jak:

- uruchomienie / zatrzymanie serwera UPnP,
- uruchomienie / zatrzymanie / uruchomienie ponowne serwera RPC,
- zatrzymanie / uruchomienie ponowne serwera zarządzającego,
- zmiana interfejsu sieciowego serwera UPnP, bądź RPC,
- zmiana portu zarządzającego serwerem RPC,
- dodawanie, modyfikacja, usuwanie ścieżek, z których są udostępniane multimedia,
- tworzenie, modyfikacja, usuwanie wirtualnych bibliotek UPnP AV,
- wymuszanie ponownego przeskanowania folderów w celu znalezienia plików multimedialnych,
- zmiana loga serwera UPnP, widocznego w odtwarzaczu UPnP AV bądź DLNA.

W całej aplikacji wykorzystywane są bazy danych *SQLite* (zawierającymi np. ścieżki do przeszukiwania). Jest to wybór stosunkowo oczywisty w kontekście umożliwienia działania na urządzeniach o niskiej wydajności. Baza danych *SQLite* została stworzona z myślą o przechowywaniu małej ilości rekordów. Obsługuje się ją poprzez zewnętrzną bibliotekę. Przy dużej ilości danych powoduje to nadmierne obciążenie aplikacji. Jednakże w przypadku wielu małych aplikacji, jak i niniejszego serwera UPnP, rozwiązanie to eliminuje konieczność instalowania dodatkowego oprogramowania bazodanowego (np. *MySQL*) działającego zazwyczaj w trybie klient - serwer (wprowadza to dodatkowy narzut komunikacyjny). [33]

Główna aplikacja, która udostępnia multimedia w sieci UPnP AV i/lub DLNA ma możliwość uruchomienia w tle (w trybie demona zgodnie z nazewnictwem systemu Linux lub usługi, zgodnie z nazewnictwem systemu Windows) lub w trybie pierwszoplanowym (ang. *foreground*), w którym wszystkie komunikaty są wyświetlane w oknie wiersza poleceń. W pierwszej kolejności aplikacja

otwiera pliki baz danych, z których wczytywane są ustawienia oraz ścieżki wyszukiwania plików multimedialnych. W przypadku poprawnego wczytania wszystkich danych

Jej uogólniony schemat działania widoczny jest na diagramach stanów oraz komunikacji (rysunku).

3.5.1 Klasa demona

ewwef

3.5.2 Klasa główna serwera UPnP

fewfwe

3.5.2.1 Obiekt MediaStore

ewqedq

3.5.2.2 Obiekt MediaItem

Metadane dotyczące plików są pobierane przy wykorzystaniu różnych narzędzi. Najprostszym przypadkiem jest typ graficzny, dla którego wykorzystywana jest wbudowana klasa języka Python - *Image*. Przy jej pomocy uzyskiwane są informacje o rozmiarze pliku i jego rozdzielczości.

Dla plik

Miniatury mogą być tworzone przy pomocy dwóch narzędzi: *ffmpegthumbnailer* dla plików wideo lub przy pomocy klasy *Image* języka Python dla plików graficznych (klasa ta posiada specjalną metodę do tworzenia miniatur).

3.5.3 Modyfikacja frameworka UPnP Coherence

ewqewq

3.5.4 Klasa serwera RPC

fewfew

3.5.5 Klasa obsługi zmiany parametrów

fwfew

Testy

rerew

4.1 Sposób testowania aplikacji

fewfew

4.2 Współpraca z odtwarzaczami w oprogramowaniu telewizorów

fewfew

4.3 Polecenia UPnP AV i DLNA w praktyce

4.4 Napotkane problemy

4.5 Wyniki testów

Przegląd oprogramowania UPnP AV i DLNA

wrew

dqwdq

5.1 Porównanie oprogramowania serwerów UPnP AV i DLNA

fewefe

5.2 Porównanie oprogramowania odtwarzaczy DLNA i UPnP AV

dqwdwq

5.2.1 Różnice działania odtwarzaczy różnych producentów

ffwefe

5.3 Obsługa standardu

Bibliografia

- [1] UPnP Forum, UPnP Device Architecture 1.0, 2008.
- [2] Digital Living Network Alliance Home Networked Device Interoperability Guidelines Version: 1.0
- [3] Digital Living Network Alliance Home Networked Device Interoperability Guidelines Version: 1.0 Errata Version 2.0
- [4] <http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-v1-Service-20110224.pdf>
- [5] http://www.google.com/url?q=http%3A%2F%2Fwww.pervasive.jku.at%2FResearch%2FPublications%2F_Documents%2F2011_Online%2520UPnP%2520AV%2520Device%2520Database%2520for%2520Quick%2520and%2520Easy%2520Capability%2520Checking_Riener.pdf&sa=D&sntz=1&usg=AFQjCNEMLsCjydhx6Qy1L30d4wZ5HgUVA
- [6] http://www.google.com/url?q=http%3A%2F%2Fwww.territorioscuola.com%2Fwikipedia%2Fen.wikipedia.php%3Ftitle%3DTalk%3ADigital_Living_Network_Alliance&sa=D&sntz=1&usg=AFQjCNFB10Qj3jNXfAqwV_NNQaRIBOk5zg
- [7] http://www.google.com/url?q=http%3A%2F%2Fsuperuser.com%2Fquestions%2F226636%2Fwhat-is-the-difference-between-dlna-and-upnp&sa=D&sntz=1&usg=AFQjCNHUjIpiufrZlgM_YLSNohaL355J_g
- [8] http://www.google.com/url?q=http%3A%2F%2Fjan.newmarch.name%2Fpublications%2Fjini_upnp.isng05.pdf&sa=D&sntz=1&usg=AFQjCNEZCczU2XK8hGsmHe7wU1mAbQpeKA
- [9] Abul Ahsan Md. Mahmudul Haque, UPnP Networking: Architecture and Security Issues, 2007
- [10] UPnP Device Architecture V1.0 Annex A – IP Version 6 Support
- [11] UPnP Device Template
- [12] Networked Digital Media Standards A UPnP / DLNA Overview, Allegro Software Development Corporation, 2006.
- [13] RFC 2616 Hypertext Transfer Protocol – HTTP/1.1, Internet Engineering Task Force, 1999
- [14] SOAP Simple Object Access Protocol, W3C, 2007

- [15] Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C, 2008
- [16] Overview of UPnP AV Architecture, Intel Corporation, 2003
- [17] Abusing Universal Plug and Play, Armijn Hemel, 2008
- [18] Design and Implementation of home network systems using UPnP middleware for networked appliances, Dong-Sung Kim, Jae-Min Lee, Wook Hyun Kwon, Kwan You, 2002
- [19] UPnP Design by Example: A Software Developers's Guide to Universal Plug and Play, Michael Jeromino, Jack Weast, 2003
- [20] UPnP Forum UPnP AV Architecture:2, 2010
- [21] Network Media Content Aggregator for DLNA Server, Jan Kubový, 2011
- [22] ContentDirectory service:2 Service Template Version 1.01 – Document Version 1.00, 2006
- [23] DLNA Overview and Vision Whitepaper 2007
- [24] http://brage.bibsys.no/hia/bitstream/URN:NBN:no-bibsys_brage_9695/1/andreas-haber-phone-based-upnp-control.pdf
- [25] Wikipedia, http://pl.wikipedia.org/wiki/Kontener_multimedialny, 2011
- [26] ISO/IEC 10918, JPEG
- [27] Cling Framework, <http://4thline.org/projects/cling>
- [28] Cyberlink for Java, <http://www.cybergarage.org/twiki/bin/view/Main/CyberLinkForJava>
- [29] MiniUPnP, <http://miniupnp.free.fr/>
- [30] MiniDLNA, <http://sourceforge.net/projects/minidlna/>
- [31] Coherence <http://coherence.beebits.net/wiki/WikiStart#CurrentState>
- [32] Licencja MIT, <http://www.opensource.org/licenses/mit-license.php>
- [33] Test Baz danych Open Source, SDJ Extra nr 17, PHP Starter Kit, <http://www.scribd.com/doc/51087611/2/Test-baz-danych-Open-Source>
- [34] Typy multimedialne MIME, <http://tools.ietf.org/html/rfc2046>