2011 Frontiers of Information Technology

Online UPnP A/V Device Database for Quick and Easy Capability Checking

Andreas Riener and Martin Gutenbrunner
Johannes Kepler University, Institute for Pervasive Computing, Linz/Austria
Tel. +43/732/2468–1432, Fax. +43/732/2468–8426
{lastname}@pervasive.jku.at

***Abstract*—Creating and customizing ubiquitous media networks with UPnP A/V compliant components from different manufacturer (brands) is quite a common task, however, poses a few problems. First, it is not possible at time of purchase of such devices to check whether or not all the devices will interoperate as desired. Second, and even more critical is the fact that manufacturer declare their products "UPnP" certified, even if not the complete function set specified mandatory is implemented. Third, it is also not possible to find or identify devices implementing a certain required function effectively. In this work we present our initiative toward a central database of UPnP A/V devices, allowing interested persons to assess the function coverage (i. e., grade of implementation) of a certain device according to the UPnP A/V standard. By providing a client-side device scanner coupled with a server-side repository – both available on the Internet – any person can contribute to the project by submitting information from UPnP devices in his/her local network and can browse the central database for device specific characteristics. Available UPnP A/V information includes implementation details of MediaServer, MediaRenderer, and simple players, as well as information about the level of function coverage of a certain device or service class. This information can be used by developers or end users to select devices matching exactly their requirements or to exclude devices lacking implementation of a necessary function.**

***Keywords*— Heterogeneous devices; UPnP A/V capability database; Interoperability of UPnP devices.**

I. Motivation

Unifying device discovery, event handling and device control makes Universal Plug and Play (UPnP) capable devices appropriate for a wide range of applications. Especially the "Universal Plug and Play Audio/Video" subset (UPnP A/V) is utilized by a vast array of home entertainment devices to benefit from these advantages [1]. Unfortunately, large parts of the functions defined in the technical specification are optional – the label "UPnP A/V" itself reveals nothing about which parts of the specification are implemented – and if so, to what degree. Therefore, building a distributed ubiquitous media network using UPnP A/V components often poses the question, what manufacturer and/or which devices are capable of fulfilling the required functional range. Although reviews in magazines and online information sources (i. e., manufacturer websites) provide some information regarding UPnP A/V functionality, they still fail to offer a holistic overview. The same holds true for (online) bulletin boards – search queries for specific information on function coverage can become very tedious as the threads are highly nested and do mostly not provide structured overviews. From a technical point of view, "Wikis" are the adequate Internet source for offering information in an organized and easily browsable manner. Searching for "UPnP A/V Wikis" currently (October 2011) results in a number of company or product web sites announcing UPnP A/V capable devices. But beside a general definition of UPnP and some configuration examples for the devices, most often not a piece of information about function coverage (on the technical level) is denoted. In addition, and for all of the before mentioned sources of information, it is unclear whether or not one can fully trust the information provided.

From this review it turned out that there is not a single, self-contained source of information dealing with the implementation coverage of the UPnP (or, in particular, the UPnP A/V) specification. This could be a solid issue if a specific UPnP A/V function is required for a problem, and a qualified device needs to be selected to fulfill the task. Even if the UPnP A/V specification seem to contain everything necessary for the desired solution, it is normally unknown or time intensive to gather which devices or products are commercially available to cover the specific functionality. The only Internet source that comes close to answering these questions is Wikipedia®, providing a comprehensive list of UPnP A/V devices [2] and comparing license models, cost, and cross-platform availability [3]. But, as with other sources, a detailed list of covered functions is not provided.

A. Research Approach

The UPnP (A/V) specification lacks some problems in its implementation and we assume that these can be discovered best from data returned from the devices itself. This is a valid approach as UPnP devices convey their capabilities automatically to other "listeners" in a network. Therefore, any application in a local network can gather capabilities of all the other UPnP devices available in the network.

To solve the before mentioned issues, the approach followed in this work to reach a trustful source of function coverage of UPnP (A/V) devices is (i) the distribution of a local network scanner automatically gathering information about the actual set of implemented functions for all UPnP devices in the network, (ii) a functionality to submit collected information to a central device database (one-click submission), and (iii) the central database itself, organizing and maintaining UPnP

978-0-7695-4625-4/11 $26.00 © 2011 IEEE
DOI 10.1109/FIT.2011.65

A/V data sets received from local scanners and providing information to the community. The database of different UPnP A/V devices available online allows for queries such as implemented functions or coverage rate of specific functions or for compatibility checks with other devices. The device database is online since almost two years and the software tool to scan the local network for new UPnP A/V devices and submit collected data sets is also available for free. Resources can be found at http://upnp-database.info.

Outline

The rest of this work is structured as follows: In section II UPnP and in particular UPnP A/V is introduced and problems covered in this work are identified. The software tool for scanning UPnP A/V devices and analyzing UPnP A/V traffic in a LAN is introduced and described in detail in III. In section IV, the central device database is analyzed on service granularity and with respect to function coverage; results are presented and conclusions are drawn and discussed. Section V finally summarizes our results and concludes the paper.

II. Universal Plug and Play (UPnP)

From a consumer's point of view, UPnP enables devices in a network to recognize each other automatically. Additionally, those devices can exchange information about their capabilities and they can control each other within the borders of the specification. From a technical point of view, however, UPnP is a collection of standardized technologies which take care of addressing, device discovery, capability description, event notification, and finally the control of UPnP devices. It is furthermore important to mention that UPnP devices and their services usually consist of more than one component, and that, depending on the category of the device (home entertainment, networking, printing, etc.) these components will differ.

A. UPnP A/V and its Relationship to UPnP

As of today, UPnP defines ten device categories (such as Home Automation, Networking, Telephony, Audio/Video) and several add-on categories [1]. In this work, we are only dealing with the *Audio/Video* subset "UPnP A/V" of Universal Plug and Play, encompassing "*MediaServer*" and "*MediaRenderer*" services in different revisions. (The most recent standardized device control protocols (DCPs) are "*MediaServer:4*" and "*MediaRenderer:3*"; however, as shown later, almost no device available today implements functions higher than standards/versions 1 or 2).

UPnP A/V Services
MediaServer:1 and MediaRenderer:1
MediaServer:2 and MediaRenderer:2
MediaServer:3
MediaServer:4 and MediaRenderer:3

Fig. 1. UPnP A/V service overview and version history.

UPnP A/V: The device control protocol specifies the behavior or functionality of a device, for example for the "*MediaServer*" that it stores and manages all the media while the "*MediaRenderer*" DCP takes care of playback of the media provided by the "*MediaServer*". To control these devices the UPnP specification introduces beside ordinary UPnP devices so called "*control points*", often simply named "*controllers*".

A control point is a component that is active (communicating) in the network. Control points are listening to announcements of devices and have furthermore the ability to issue commands to other devices and are, in general, directly connected to the user interface (UI). Both "*MediaServer*" and "*MediaRenderer*" can not directly interact with each other – a "*control point*" is required in between, querying the "*MediaServer*" for content and showing gathered information on the UI (i. e., display). To give an example, if the user decides for a certain title to be played, the control point takes action by submitting the address of the item (URL) to the "*MediaRenderer*" and issues the playback command. The "*MediaRenderer*" immediately connects to the "*MediaServer*" and starts the playback. Which streaming protocol to use and what types of media being supported is not part of the UPnP A/V specification – it is solely up to server and client devices whether playback is possible or not. Technically, the control point queries all the information on both types of devices required to decide on matching protocol and file type. As long as no other command is issued by the user, the control point does not actively take part in the communication between media server and renderer. (This is not fully the truth, the control point may query information such as transfer status to be able to display the playback progress, remaining time, or other status information; Fig. 2).

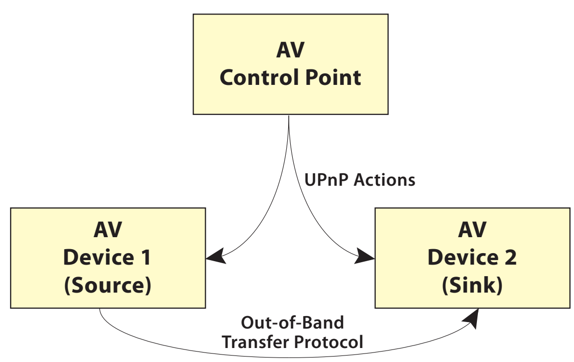

Fig. 2. The control point initiates the transfer between source (MediaServer) and sink (MediaRenderer). Playback itself does not involve the control point.

B. Related Work

As UPnP in its original form has already been used in a wide field of applications [1], there is much support that is suitable also for a broad range of future application scenarios. The UPnP A/V standard as part of UPnP is nothing short as flexible and widespread, and the suitability of UPnP A/V for new solutions depends most likely on the level of function coverage of UPnP capable devices and applications (i. e., they would have no ability to "talk" one to the other if the required function to do so is not implemented on one of the devices).

WiMAC@home [4] is a project aiming a holistic approach for controlling all aspects of a modern home by using Digital Living Network Alliance (DLNA), a standard derived from UPnP as an attempt to normalize media interoperability. The authors conclude that devices such as television sets should be equipped with network connectivity in order to force penetration of UPnP and DLNA services. Tusch and colleagues [5] extended the specification of UPnP A/V for context-aware usage of home multimedia systems. In particular, they dealt with the problem of incompatible file formats between the "*MediaServer*" and the "*MediaRenderer*" and showed a solution by implementing a control point that *transcodes* the exchanged media content.

- **Problem statement:** The problem in these approaches not considered so far is, that the user – at time of purchase of devices such as TV, receiver, NAS storage, etc. – does not have the possibility to check whether or not all the devices will interoperate as desired. Even more critical is that manufacturer of home electronic equipment declaring their products "DLNA" or "UPnP" certified even if not the complete function set as specified is implemented. To solve these issues, we developed the UPnP database (upnp-database.info) described in this work. The information available in this database (UPnP A/V devices and implemented function set) can be used manifold, e. g., to list all DLNA enabled TV sets, to check if a certain device implements a specific function (service) required or which hard disc recorders are "compatible" with a particular receiver.

Park *et al.* [6] proposed a system extension to cope with the issue of source-sink data communication. They developed a transparent protocol selection module on top of the control point. This module automatically selects the "best" protocol for data exchange between a "*MediaServer*" (source) and a specific "*MediaRenderer*" (sink) based on their individual specifications. Kang and colleagues [7] tried to combine UPnP A/V with OSGi and successfully provided UPnP services outside the "closed" home environment.

- **Problem statement:** The extension of control points is somehow problematic for the approach of data gathering as implemented in our work, as we are not able at all to extract information from (or about) control points, thus would not be able to report about such functional extensions. In addition, the combination of UPnP with other "outer" technologies is another aspect we are not able to capture with our tool in the current version. However, these two issues are only theoretically relevant, as these extensions [6], [7] representing research work which would not find its way into commercial products (the focus of our database and scanner) in the next time.

III. UPnP A/V Tool Set

The tool set developed in the course of this work consists of a client-side scanner application and the server-side application connected to a database for retrieving, organizing/analyzing, and presenting content received from the clients. Below, we will give insights into implementation details of the individual parts of the framework.

A. Device scanner (client application)

The device scanner collects data from all UPnP A/V devices in a local network and finally requests the user to submit the results to the central server. It is implemented as a simple, portable Java desktop application. Due to fact that the selected UPnP framework did not contain the required UPnP A/V specific implementations for our evaluation, this part was developed fully in-house [8]. In general, the application can distinguish between, and analyze three different types of devices:

- UPnP A/V MediaServer
- UPnP A/V MediaRenderer
- Plain Players

The capabilities of each and every UPnP A/V device are defined in XML files, describing hardware type, implemented functions, etc. In addition to the information on functions defined in the UPnP standard, product vendors tend to state functions as implemented even if they are not. So they pretend a higher rate of function coverage than what is actually available. By simply analyzing the XML files of the UPnP A/V device, it is possible to identify the functions which have been added accessorily by a manufacturer. This information is – together with information on standard functions – sent to the central server after a user has accepted to do so. One prominent case of additionally implemented functions is Digital Living Network Alliance (DLNA) [9], a standard based on UPnP A/V. The most distinctive difference between UPnP A/V and DLNA is that DLNA devices have to be officially certified. (DLNA certified devices are still full compatible with "plain UPnP A/V" devices.)

1) UPnP A/V MediaServer: UPnP A/V MediaServers are analyzed by browsing their whole media structure. This way the nodes of all possible media types such as music tracks, photo albums, containers, movies, etc. can be analyzed in its full details. This self-contained information is of utmost importance for developers, as they want to know which information could possibly be returned from a certain device. It has to be mentioned here that a "*MediaServer*" can only deliver information it has media files for – the analysis will yield no result for media files for which the running server instance would be capable of, but does not have entries. This is one of the reasons why multiple entries of the same type of "*MediaServer* are not removed from the database or combined to a single entry. Another reason is that large parts of the server software are open source, with software developer often modifying the functional range, resulting in differences for data collection or presentation.

2) UPnP A/V MediaRenderer: UPnP A/V MediaRenderer devices are analyzed by just querying the XML files available, and without carrying out additional actions.

3) Plain (passive) UPnP A/V devices: Plain UPnP A/V device players are the third type of devices; they can be analyzed using the "Device scanning application". The difference between plain players and UPnP A/V MediaRenderers is that the first do not actively participate in UPnP communication, they just call the necessary Application Programming Interfaces (APIs) to be able to acquire URL's and stream media from servers. As they do not broadcast information, they are not visible for other UPnP devices and cannot be analyzed automatically. To grab information from this type of device as well, the analysis tool provides a simple implementation of a UPnP A/V MediaServer with basic content. Plain player devices are aware of MediaServers and try to browse their content. The analysis tool keeps track of this process (by collecting information which of the server actions have been called by the player). This way it is possible to identify at least basic characteristics and behavior of the examined plain player device.

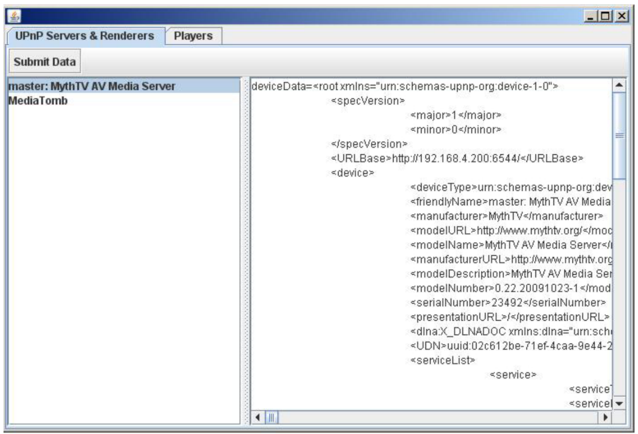

Fig. 3. UPnP device scanner with some collected data in XML format. The left side of the application lists all the detected UPnP-MediaServers in the local network. The right pane shows detailed information of the selected device. After accepting, all the information collected is transferred to the server, stored in the central database, and immediately displayed on the website.

B. Server-side application

The UPnP A/V server (http://upnp-database.info/) hosts a Java web application that listens for data from the client application(s), and presents the collected data together with some statistics on the website. For this purpose, data is provided either in a single list, containing all devices, or optionally filtered by device type. The initial information shown when browsing the database is a presentation of general data about the device like manufacturer or device description (see Fig. 4). One hierarchy level deeper, detailed information on provided/implemented services is presented for the selected device. Each device type has a defined set of services with mandatory or optional functions (see Figures 5, 6). According to the UPnP specification, a device does not only consist of services and actions, but also of state variables. The service data page displays all of this information. Although this information can also be gained from the official UPnP documentation, it is device specific to which extent the specification is implemented. The device scanning application also calls some of the actions implemented to gain additional information. If called, the result of this action is also displayed on this page (e. g., the information which protocol(s) the device is capable of).

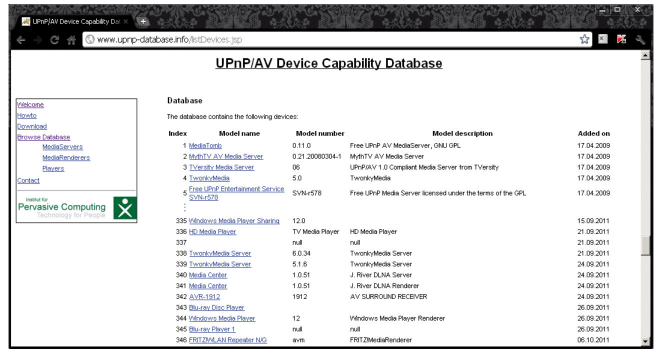

Fig. 4. Server application at http://upnp-database.info.

All the data about UPnP A/V devices submitted by instances of the client application is stored in a MySQL database. The database structure follows the native UPnP object hierarchy, while the table "devices" was extended by additional parameters to be able to store information not contained in the standard UPnP XML structures.

IV. UPnP Database Analysis

The web service providing the central device database is online since June 2009 and accessible at http://upnp-database.info. In the last two years more than 350 device specifications have been collected (i. e., provided by volunteers running our UPnP A/V tool in their local networks). In the next paragraphs, the most interesting results from a deep analysis of the database items will be provided and discussed.

A. List of Services

A.1 RenderingControl

The "*RenderingControl:1*" service is intended to provide control points with the ability to query and/or adjust any rendering attribute that the device supports [10]. Two functions are defined mandatory, and they are implemented by almost all devices in our database (Fig. 5). Most of the over 30 optional commands are implemented by only 10% or less of the devices, and only four functions (*Get/SetVolume*, *Get/SetMute*) have a coverage rate higher than 30% (actually 72 to 74%).

The next generation of the "*RenderingControl*" service (v2) differs only marginally from the original version, as by providing two added methods ("*GetStateVariables*"/"*SetStateVariables*") allowing for easy and conveniently changing state variable values. In the bunch of database rows only a single device implements "*RenderingControl:2*", but even this device does not offer implementation for the two optional functions indicated before. (Not a single device is based on the current "*RenderingControl:3*" specification.)

A.2 AVTransport

The "*AVTransport*" service in its first version is the service with most actions to be implemented mandatory (Fig. 5, right). Furthermore, it is one of two services part of both the "*MediaServer*" as well as the "*MediaRenderer*" specification

RenderingControl:1						AVTransport:1		
Function	*R/O*	*Degree*	*Function*	*R/O*	*Degree*	*Function*	*R/O*	*Degree*
ListPresets	Req	99 %	GetBlueVideoGain	Opt	6 %	GetDeviceCapabilities	Req	100 %
SetPreset	Req	99 %	GetSharpness	Opt	6 %	GetMediaInfo	Req	100 %
SetVolume	Opt	73 %	GetVerticalKeystone	Opt	6 %	GetPositionInfo	Req	100 %
GetVolume	Opt	74 %	GetColorTemperature	Opt	6 %	GetTransportInfo	Req	100 %
SetMute	Opt	72 %	GetGreenVideoBlackLevel	Opt	6 %	GetTransportSettings	Req	100 %
GetMute	Opt	74 %	GetGreenVideoGain	Opt	6 %	Next	Req	100 %
SetVolumeDB	Opt	30 %	SetBlueVideoBlackLevel	Opt	6 %	Previous	Req	100 %
GetVolumeDB	Opt	30 %	GetHorizontalKeystone	Opt	6 %	Play	Req	100 %
GetVolumeDBRange	Opt	23 %	SetBlueVideoGain	Opt	6 %	Stop	Req	100 %
SetLoudness	Opt	19 %	SetColorTemperature	Opt	6 %	Seek	Req	100 %
GetLoudness	Opt	19 %	SetGreenVideoBlackLevel	Opt	6 %	SetAVTransportURI	Req	100 %
SetBrightness	Opt	10 %	SetGreenVideoGain	Opt	6 %	Pause	Req	94 %
GetBrightness	Opt	10 %	SetHorizontalKeystone	Opt	6 %	GetCurrentTransportActions	Opt	77 %
GetContrast	Opt	10 %	SetRedVideoBlackLevel	Opt	6 %	SetPlayMode	Opt	48 %
SetContrast	Opt	10 %	SetRedVideoGain	Opt	6 %	SetNextAVTransportURI	Opt	45 %
GetRedVideoBlackLevel	Opt	6 %	SetSharpness	Opt	6 %	Record	Opt	1 %
GetBlueVideoBlackLevel	Opt	6 %	SetVerticalKeystone	Opt	6 %	SetRecordQualityMode	Opt	1 %
GetRedVideoGain	Opt	6 %						

Fig. 5. UPnP A/V "RenderingControl:1" and "AVTransport:1 service classes.

– making "*AVTransport*" one of the most important parts of the UPnP A/V specification. The entire set of the 12 required functions is implemented by all the devices in the database built-up over the last two years. Interesting is, that the optional functions for recording (i. e., *Record*, *SetRecordQualityMode*) are implemented by very few devices (coverage rate 1%).

Version 2 of the "*AVTransport*" service extends version 1 with three (optional) functions, the two as described for the "*RenderingControl:2*" service above and, in addition, the action "*GetDRMState*". In the central database only a single device implements "*AVTransport:2*"; optional functions/commands are again not implemented.

A.3 ConnectionManager

The service type "*ConnectionManager:1*" (Fig. 6, left) enables modeling of streaming capabilities of audio/video devices and allows binding of these capabilities between different devices [11]. Required functions are implemented by 96 to 98% of devices, optional commands are covered by only 6% of the elements in the device database. Interesting here is, that the newer version "*ConnectionManager:2*", which is in terms of functionality identical to its first version, comes with a higher degree of coverage (required functions are implemented by 100%, optional commands are covered by half of devices; Fig. 6). In general, the enhancement from one version to the next version (in the UPnP A/V standard) is not only defined by a extended function set, but also accompanied by more detailed information for developers [12].

A.4 ContentDirectory

In order to be able to share media within the network, a service is needed providing uniform and consistent mechanisms to user interfaces (i. e., devices connected to a UI) for browsing server content and retrieving detailed information about individual objects (title, artist, length, etc.). This functionality is defined in the "*ContentDirectory:1*" service [13]. Two out of 4 mandatory functions are not implemented by all devices (coverage rate of 97% each), the optional *Search* command is implemented by 82% of devices, and all the other optional functions are covered by less than one-third of devices (Fig. 6, second to the right).

"*ContentDirectory*" in its second version facilitates improved mechanisms for sorting media items and provides additional information/features of objects as well as a basic function for moving objects within the directory. A single device in our database implements "*ContentDirectory:2*". The most current version of this service class is "*ContentDirectory:4*"; however this (and also "*ContentDirectory:3*") is not used by devices so far, at least not by them submitted to our database.

A.5 ScheduledRecording

This service type, released in versions "*ScheduledRecording:1*" and "*ScheduledRecording:2*", is a specialty as it is not implemented in a single device in the central database (neither version 1 nor version 2). From the analysis of more than 350 devices we assume that this service class is, by now, also not implemented by other UPnP A/V devices available.

A.6 "ContentSync"

The "*ContentSync*" service is one of the newest classes (released in 2009). The DCP specifies synchronization mechanisms between devices implementing the "*ContentDirectory*" service. The synchronization of media content between devices in a network was one of our main motivations to start the work presented here. Theoretically, devices implementing the "*ContentSync*" service class should have the capability for media synchronization. Unfortunately, we could not find a single device on the market offering this service; manufacturer websites or other Internet resources provided also no information about available devices or of devices short before introduction on the market. In a deeper analysis of "*ContentSync*" we noticed that it is not part of the central UPnP A/V device control protocols, but specified as add-on service. Furthermore, the specification indicates that the service must be deployed on devices also offering the "*ContentDirectory*" service – as the latter is part of the UPnP A/V DCPs, it can be inferred that the "*ContentSync*" service should be found on UPnP A/V devices.

ConnectionManager:1		
Function	*R/O*	*Degree*
GetProtocolInfo	Req	99 %
GetCurrentConnectionInfo	Req	98 %
GetCurrentConnectionIDs	Req	96 %
ConnectionComplete	Opt	6 %
PrepareForConnection	Opt	6 %

ConnectionManager:2		
Function	*R/O*	*Degree*
GetProtocolInfo	Req	100 %
GetCurrentConnectionInfo	Req	100 %
GetCurrentConnectionIDs	Req	100 %
ConnectionComplete	Opt	50 %
PrepareForConnection	Opt	50 %

ContentDirectory:1		
Function	*R/O*	*Degree*
Browse	Req	100 %
GetSystemUpdateID	Req	100 %
GetSearchCapabilities	Req	97 %
GetSortCapabilities	Req	97 %
Search	Opt	82 %
UpdateObject	Opt	31 %
DestroyObject	Opt	27 %
CreateObject	Opt	26 %
CreateReference	Opt	23 %
GetTransferProgress	Opt	23 %
ImportResource	Opt	23 %
DeleteResource	Opt	11 %
StopTransferResource	Opt	11 %
ExportResource	Opt	10 %

ContentDirectory:2		
Function	*R/O*	*Degree*
GetSortExtensionCapabilities	Opt	100 %
GetFeatureList	Opt	100 %
MoveObject	Opt	100 %

Fig. 6. UPnP A/V service classes "ConnectionManager" (v1, v2) and "ContentDirectory"(v1, v2) with specified functions/actions.

For these reasons, we developed the central device database and distributed the tool for scanning local networks with the aim to collect as much device specifications as possible. Our hope was, to get a list of few devices implementing the "*ContentSync*" service class; however, this hope has not been fulfilled: Not a single entry in the device database implements the "*ContentSync*" service. To conclude, we implemented the service on our own; findings and implementation details are presented elsewhere.

B. Service/Function Coverage Rates

According to our central device database, services indicated mandatory in the UPnP A/V specification are implemented at rates between 94 and 100 percent (Figures 5, 6). The fact that functions required by a standard are not covered by 100 percent of devices certified or specified as standard conform highlights the problem that some of the product manufacturer and vendors interprets such standards more as kind of framework than as binding specification. It becomes obvious that not all of the UPnP A/V implementations (devices) are standard compliant. This supports our initiative for a open database of UPnP A/V devices with details about implemented (or missing) functions even more. Further support is given by the fact that there is no obligation for manufacturer to conduct compliance tests, making it even more problematic, e. g., to customize a home entertainment system with components from different manufacturer/brands, where all the devices cooperates and interoperates with all other devices.

When looking on optional functions, one can see that only few of them reaches a coverage rate of 30% or more, with most are in degree of implementation in the 10s or even less. Optional functions with highest coverage rates correspond to functions often used on media players, and also found on most remote controls. These are *Get/SetVolume* and *Get/SetMute*. In addition the *Search* command, used to browse a directory based on keywords or artist name, achieves high coverage rate of 82%.

UPnP A/V is not a steady standard but continuously enhanced. October 2011, the newest versions released are "*MediaRenderer:3*" and "*MediaRenderer:4*" [1]. However, from more than 350 data sets in the database, only one device (Coherence UPnP A/V MediaServer[1]) implements all of its methods according to version 2 of the UPnP A/V specification and not a single DB entry corresponds to a device implemented based on UPnP A/V v3 or v4. When analyzing contents based on service classes it can be indicated that some classes, such as "*ScheduledRecording:1*", "*ContentSync:1*", "*AVTransport:2*", "*RenderingControl:2*", "*ScheduledRecording:2*", "*ContentDirectory:3*", etc. are not implemented at all, i. e., not a single entry in the device database refers to that service.

Beside the functional range as defined in the specification, device manufacturer may specify additional services or functions. A prominent example of a added service is "*X-MS_MediaReceiverRegistrar*". This additional service class is implemented by 41 devices in the database, or about 11%.

C. Unused potential of UPnP A/V hardware

In the following, we summarize our findings of why the diverse functions have not been implemented by some of the manufacturer at all, or only to a small degree.

- Functions with highest coverage rates in the "*RenderingControl*" service are the one that can be found on remote controls of rendering devices such as TV sets (volume up/down, mute). Functions with a low degree of implementation are the one that are only rarely adjusted (e. g., adjust brightness, sharpness) or those kept untouched/fixed for a longer time [14].
- The recording functionality specified in "*AVTransport*" is, according to our database, only implemented in 1% of the devices. This is of particular interest, as many software and hardware solutions for video recording are on the market, and lot of the software products are open source. Analyzing 529 set-top boxes equipped with an Ethernet connector [15] revealed that 345 devices (65%) are equipped with built-in storage, 91 (17%) have a interface to connect external storage devices, and 64 (12%) can be enhanced with internal storage. As far as hardware is concerned, almost all of the 529 devices are capable

[1] http://www.upnp-database.info/device.jsp?deviceId=149.

of UPnP A/V recording. On the software side, however, it is discoverable that almost half of the set-top boxes (238, 45%) are using Linux®as operating system [16] (open source), and 230 of these devices come with internal storage or can be upgraded with internal/external disk space. To conclude, the low coverage rate of 1% for the *record* function does not follow from lacking hardware support (as nearly 50% of devices have record capabilities), but is caused by lacking software support from device manufacturer.

- Transferring content (resources) between different "*MediaServer*" instances using UPnP A/V functionality should be, in general, no problem, as the corresponding functions required are defined in the standard. One of the reasons for the low coverage rate of these functions in UPnP devices may be the fact that a consumer would normally have only a single "*MediaServer*", and therefore no need to transfer media content to other "*MediaServer*" instances. A problem could emerge in the next time, as mobile media players potentially come with an integrated "*MediaServer*", which would enable users to conveniently synchronize content between stationary and mobile devices with wireless connections and without being bound to proprietary manufacturer standards (wires, jacks) – given that the required UPnP A/V functions are implemented.

V. Conclusions

In this paper we have summarized the tool-based analysis of UPnP A/V devices and applications, where the collected data is submitted to a central database. This enables projects with different goals to get detailed information about the current state of implementation (i. e., function coverage) of a certain device. Based on this detailed information on device granularity, future development of UPnP A/V applications and devices should be significantly simplified. Evaluation of the latest version of the data pool has been presented, discussed, and enriched with personal interpretation of the authors (e. g., on the low rate of function coverage rate).

UPnP A/V Database		
Source	*Visits*	*Proportion*
en.wikipedia.org	9,571	37.40 %
upnp-database.info (*direct*)	8,416	32.89 %
google	4,946	19.33 %
de.wikipedia.org	1,067	4.17 %
ps3mediaserver.org	154	0.60 %
SUM	**25,154**	**100.0%**

Fig. 7. Page hits of *upnp-database.info* and their origin (October 2011).

The number of page hits (Fig. 7) of more than 25,000 over the last two years clearly indicates that there is indeed a need for such an application. Nevertheless, the database could even play a much higher role in UPnP (A/V) development as it does now, e. g., by extending it to a more powerful vendor independent platform for developers and manufacturer. Nokia, for example, provides related information about the experiences with UPnP during the last years in a "lessons learned" document [17].

UPnP devices are (and will be) operated by laymen lacking necessary troubleshooting skills. Although the UPnP forum propagates "Plug&Play" already by it's name, still only experienced users are able to debug and fix failures and the lack of debugging toolkits is another serious problem. This motivates even more to maintain a central database with detailed device information, where people can check (before purchase) whether or not a certain device would work in their network, or has a specific function implemented.

References

[1] UPnP Forum. Announced Certified Product Registry. Online. Last retrieved October 17, 2011. [Online]. Available: http://upnp.org/sdcps-and-certification/standards/sdcps/

[2] Wikipedia. List of UPnP AV media servers and clients. Online. Last retrieved October 13, 2011. [Online]. Available: http://en.wikipedia.org/wiki/List_of_UPnP_AV_media_servers_and_clients

[3] ——. Comparison of UPnP AV media servers. Online. Last retrieved October 13, 2011. [Online]. Available: http://en.wikipedia.org/wiki/Comparison_of_UPnP_AV_MediaServers

[4] M. Schmalohr, "UPnP-AV und DLNA Standards für das vernetzte Wohnen," *rfe-online*, vol. 6/2008, pp. 23–24, June 2008.

[5] R. Tusch, M. Jakab, J. Köpke, A. Krätschmer, M. Kropfberger, S. Kuchler, M. Ofner, H. Hellwagner, and L. Böszörmenyi, "Context-Aware UPnP-AV Services for Adaptive Home Multimedia Systems," *International Journal of Digital Multimedia Broadcasting*, vol. 2008, no. 835438, p. 12, July 2008.

[6] J. Park, H. Lee, and S. Kim, "A Transparent Protocol Scheme Based on UPnP AV for Ubiquitous Home," in *ISPA 2007 Workshops*, P. Thulasiraman et al., Ed. Springer-Verlag Berlin Heidelberg, 2007, pp. 153–162.

[7] D.-O. Kang, K. Kang, S.-G. Choi, and J. Lee, "UPnP AV architectural multimedia system with a home gateway powered by the OSGi platform," in *International Conference on Consumer Electronics (ICCE 2005)*, ser. Digest of Technical Papers, January 2005, pp. 405–406.

[8] S. Konno, "CyberLink for Java," Online, September 2011, last retrieved October 13, 2011. [Online]. Available: http://www.cybergarage.org/cgi-bin/twiki/view/Main/CyberLinkForJava

[9] DLNA Organization, "The Digital Living Network Alliance (DLNA)," Online, 2011, last retrieved October 13, 2011. [Online]. Available: http://www.dlna.org

[10] S. Kou, T. Matsui, J. Moonen, Y. Rasheed, and J. Ritchie, "RenderingControl:1 Service Template Version 1.01," Online, UPnP Forum, June 2002, last retrieved October 13, 2011. [Online]. Available: http://upnp.org/specs/av/UPnP-av-RenderingControl-v1-Service.pdf

[11] S. Chan, A. Dara-Abrams, M. Dawson, J. K. Fu, F. Matsubara, J. Moonen, Y. Rasheed, D. Sather, and E. Shteyn, "ConnectionManager:1 Service Template Version 1.01," Online, UPnP Forum, June 2002, last retrieved October 13, 2011. [Online]. Available: http://upnp.org/specs/av/UPnP-av-ConnectionManager-v1-Service.pdf

[12] A. Presser, G. Langille, G. Shults, J. Ritchie, M. Walker, C. Kim, S. Ahn, M. Hori, M. Ma, J. Unverferth, W. Bronnenberg, G. Knapen, R. Berkoff, I. Shen, N. Kikkawa, J. Tourzan, and Y. Morioka, "ConnectionManager:2 Service Template Version 1.01," Online, UPnP Forum, May 2006, last retrieved October 13, 2011. [Online]. Available: http://upnp.org/specs/av/UPnP-av-ConnectionManager-v1-Service.pdf

[13] K. Debique, T. Igarashi, S. Kou, J. Moonen, J. Ritchie, G. Schults, and M. Walker, "ContentDirectory:1 Service Template Version 1.01," Online, UPnP Forum, June 2002. [Online]. Available: http://upnp.org/specs/av/UPnP-av-ContentDirectory-v1-Service.pdf

[14] P. Fiekers, "Flachbild-Fernseher sind ab Werk falsch eingestellt," Online, March 2010, last retrieved October 13, 2011. [Online]. Available: http://www.digitalvd.de/dvd-heimkino-news/10902,Flachbild-Fernseher-sind-ab-Werk-falsch-eingestellt.html

[15] [Online]. Available: http://geizhals.at/eu/?cat=satrecv&xf=282_Ethernet

[16] [Online]. Available: http://geizhals.at/eu/?cat=satrecv&xf=282_Ethernet~275_Linux

[17] Franklin Reynolds, Nokia Research Center, Cambridge, "The Ubiquitous Web, UPnP and Smart Homes," Online, 2 2006, presentation slides, pp. 17, last retrieved October 13, 2011. [Online]. Available: http://www.w3.org/2006/02/slides/reynolds.pdf