

## *Alles Verbunden – Von UPnP zur Geräte-Föderation*

Andreas Heil, Christopher Thiele  
{andreas.heil | christopher.thiele}@stud.uni-karlsruhe.de  
Universität Karlsruhe (TH)  
Fakultät für Informatik

Betreuer der Arbeit: Dr.-Ing. Martin Gaedke  
Art der Arbeit: MWRG Projekt  
GI-Fachbereich: Infrastruktur

### **Zusammenfassung**

Universal Plug and Play (UPnP) ist ein Industriestandard, mit dem Ziel die Benutzung und Vernetzung von elektronischen Geräten und Computern zu verbessern und zu vereinfachen. Der durch UPnP-Funktionalität entstehende Mehrwert kann in der Regel jedoch nur innerhalb des UPnP-eigenen Sub-Netzes genutzt werden. Diese Funktionalität auch außerhalb des UPnP-eigenen Sub-Netzes zugänglich zu machen, erlaubt die hier vorgestellte, auf Web-Technologie basierende, generische Management-Komponente.

## **1. Einführung**

Universal Plug and Play (UPnP) [Mi00] ist ein weit verbreiteter Industriestandard. Über 770 Firmen aus allen Bereichen der Unterhaltungselektronik, Haus-Automation, Netzwerktechnik, Fotografie und der Computerbranche erstellen im Rahmen des UPnP-Forums [UP05] Gerätespezifikationen auf Grundlage der UPnP-Architektur [UP03]. Ein solches Gerät zu besitzen ist jedoch nicht damit gleichzusetzen von dessen zusätzlicher Funktionalität zu profitieren. Unweigerlich stellt sich die Frage, worin der Mehrwert liegt, ein UPnP-fähiges Gerät in eine bestehende Infrastruktur zu integrieren.

UPnP-Funktionalität lässt sich erst dann nutzen, sobald sich mehrere UPnP-fähige Geräte oder Programme innerhalb eines Netzwerkes befinden. Das UPnP Forum hat sich zum Ziel gesetzt, die Integration von Geräten in Netzwerke zu vereinfachen. Hierfür sind UPnP-fähige Geräte und Programme in der Lage, ein auf TCP/IP basierendes Peer-To-Peer-Netzwerk aufzubauen. Dieses Sub-Netzwerk und die darin angebotene Funktionalität sind jedoch ausschließlich UPnP-Geräten vorenthalten. In der Praxis ist daher der Zugriff auf UPnP-Funktionen in einer heterogenen Geräte- und Service-Landschaft, bestehend aus nicht-UPnP-fähigen Geräten, nicht möglich. Andererseits bleiben Geräte die sich nicht in UPnP-Netzwerk eingliedern lassen ungenutzt, obwohl gerade deren Funktionalität unter Umständen im Bereich der Haus-Automation wünschenswert wäre, auch wenn dies nicht vom Hersteller so vorgesehen ist. In dem Paper werden die folgenden Schwerpunkte präsentiert:

- Die Beschreibung eines Lösungsansatzes, wie sich UPnP-Funktionen in einer heterogenen Geräte- und Dienste-Umgebung integrieren lassen und UPnP-Geräte Zugriff auf die außerhalb des eigenen Netzes befindliche Geräte erlangen.
- Die Realisierung des Ansatzes und dessen Einsatz in der Praxis. Eine funktionsfähige Implementierung zeigt, wie sich die UPnP-spezifischen Funktionen in einem heterogenen Hardware-Umfeld nutzen lassen. Darüber hinaus wird gezeigt, dass sich die Funktionen sogar nahezu mit jedem Gerät oder Programm nutzen lassen, selbst wenn es nicht in der Absicht des Herstellers lag, das Gerät in einem Netzwerk einzusetzen. Einen Schritt weiter erlaubt das System sogar die Funktionalität in einem föderierten Ansatz außerhalb des lokalen Systems zugänglich zu machen und somit eine föderationsübergreifende Integration der Funktionalität zu gestatten.

## 2. Geräte Föderation

### 2.1. UPnP Integration

Um auf die spezifischen Funktionen eines UPnP-Gerätes zuzugreifen, kann auf die ausführlichen Spezifikationen des UPnP-Forums zurückgegriffen werden. Um ein Gerät oder ein Programm diese Funktionalität nutzen zu lassen, muss dieses die entsprechenden Protokolle kennen und große Teile der Spezifikation implementieren. Interessante UPnP-Geräte für die Haus-Automatisierung im Bereich der Unterhaltung wären der Media Renderer zum Wiedergeben als auch der Media Server zum Bereitstellen und Durchsuchen von Mediadaten. Im Bereich der Haussicherheit und -wartung könnte sowohl die Digital Security Camera als auch das HVAC<sup>1</sup> Anwendung finden. Die Herausforderung liegt nun darin, die Funktionen dieser verschiedenen Geräte außerhalb des automatisch aufgebauten und in sich geschlossenen Peer-To-Peer-Netzwerks zur Verfügung zu stellen.

### 2.2. FDX

Zur Lösung des Problems wird auf das Konzept des Federated Device Assembly (FDX) zurückgegriffen [He05]. Hierbei wird eine integrierte, auf Web-Technologien basierende Management-Komponente erstellt, über die sich beliebige Geräte ansprechen lassen können. Das FDX stellt eine generische Schnittstelle (STAIVE) zur Verfügung, mittels derer sich auf gerätespezifische Funktionen zugreifen lässt. Die Schnittstelle wurde zunächst mittels der Web Services Description Language (WSDL) [W301] als Vertrag für alle Teilnehmer spezifiziert. Hierdurch ist die Schnittstelle sowohl sprach- als auch plattformunabhängig definiert. Aufgrund der Realisierung als Web Service erweist sich auch die Implementierung als sprach- und plattformunabhängig, um das System so wenig wie möglich einzuschränken und die Wiederverwendung der Komponenten zu fördern.

### 2.3. Aufbau des UPnP-FDX

Abb. 1 zeigt die Architektur bestehend aus vier Elementen. Um neue Geräte im UPnP-Netzwerk zu erkennen und Abmeldungen vorhandener Geräte zu empfangen, ist eine Anwendung notwendig, die die Multicast-Adresse 239.255.255.250:1900 abhört und alle notwendigen Informationen an das UPnP-Gateway weiterreicht. So ist sichergestellt, dass das System stets über die vorhandenen Geräte informiert ist. Diese Anwendung ist in der Abbildung mit dem Namen UPnP Rootdevice Finder gekennzeichnet.

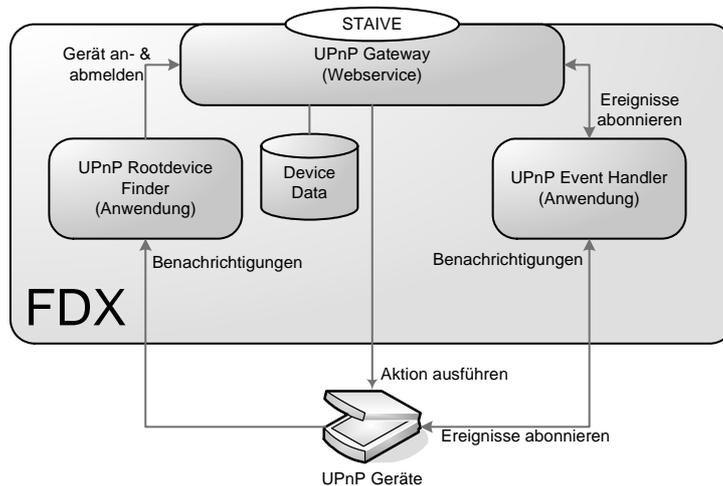


Abb. 1: UPnP-FDX

<sup>1</sup> Die Abkürzung HVAC steht für Heating, Ventilating und Air-Conditioning.

Das UPnP Gateway implementiert die STAIVE-Schnittstelle und stellt die Verbindung zu den UPnP-Geräten her. Will der Benutzer eine Aktion auf einem UPnP-Gerät ausführen, so wird diese direkt vom Gateway aus aufgerufen. Die Ereignisbehandlung wurde in einen eigenständigen Prozess ausgelagert. Dieser UPnP Event Handler verwaltet und erneuert die jeweiligen Ereignisabonnements. Erhält der Prozess eine Ereignisbenachrichtigung, so leitet er diese an das Gateway weiter, das wiederum die registrierten Aktionen ausführt. Sämtliche Informationen, die aus den Geräte- und Dienstbeschreibungen der UPnP-Geräte hervorgehen, werden in der Device Data zwischengespeichert. Durch diese Maßnahme wird ein mehrfaches Laden und Verarbeiten der Beschreibungen vermieden, wenn beispielsweise eine Aktion ausgeführt werden soll.

Um das jeweilige Gerät und die entsprechende Funktion genau zu spezifizieren, wird dem Aufruf ein maschinen-lesbarer Kontext übergeben. Formuliert wird dieser Kontext plattformübergreifend in Form von XML. Einmal an das FDX übergeben, wird die Anfrage ausgewertet und die entsprechende Funktion des UPnP-Geräts aufgerufen. Weiterhin unterstützt das FDX die Abfrage von Statuswerten und das Abonnement von Ereignissen, die durch UPnP-Geräte ausgelöst werden.

Als direktes Ergebnis können nun sämtliche Funktionen, die von UPnP-Geräten bereitgestellt werden, von jeder beliebigen Applikation aufgerufen werden, da lediglich der Endpunkt des einen Dienstes bekannt sein muss, der durch das FDX realisiert wird. Zusätzlich unterstützt jedes FDX die Möglichkeit Funktionen anderer FDX direkt aufzurufen. Weitere FDX, die wiederum die Funktionalität anderer Geräte kapseln, können somit leicht mit den UPnP-Geräten kombiniert werden. Die Beschreibung der Funktionen eines jeden FDX erfolgt bei diesem Ansatz ebenfalls über ein maschinen-lesbares XML Format. Das UPnP-FDX kann somit in ein UPnP-Netzwerk integriert werden und stellt dessen gesamte Funktionalität- an einem zentralen Endpunkt bereit. Aufgrund der Realisierung als Web Service lässt sich das FDX und somit die gesamte UPnP-Funktionalität in eine bestehende Föderation aus Diensten integrieren.

Unter dem Einsatz der Programmiersprache C# wurde das UPnP-FDX basierend auf dem Microsoft .NET Framework 2.0 [Mi06] entwickelt und getestet. Als Validierung wurde das UPnP-FDX mit unterschiedlichen UPnP-Geräten getestet. Bei den dabei eingesetzten Geräten handelte es sich sowohl um Hardware als auch um in Software simulierte Geräte [In05].

### **3. Stand der Technik**

Mehrere Ansätze existieren bereits mit dem Ziel weitere Geräte in UPnP-Netzwerken zu integrieren. Die Open Service Gateway Initiative [Ma01] stellt eine Plattform zur Verfügung, die es erlaubt, externe Dienste mittels einer so genannten Service Platform innerhalb eines UPnP-Netzwerkes zur Verfügung zu stellen. Im Gegensatz hierzu haben die Forschungsprojekte Shaman [Sc04] und Sindrion [Go04] das Ziel, Geräte, die die UPnP-Spezifikation nicht implementieren, in UPnP-Netzwerke zu integrieren. Alle drei Ansätze ermöglichen den Zugriff auf externe Dienste und Geräte-Funktionen innerhalb eines UPnP-Netzes. Allerdings bleibt diese Funktionalität auch weiterhin für Dienste und Geräte außerhalb dieses Netzes verborgen.

### **4. Ergebnis und weiterführende Arbeiten**

Durch die Implementierung weiterer FDX für andere Geräteklassen wurde der Aufbau eines Systems verschiedenster Hard- und Softwareklassen möglich. Als weiteres FDX wurde beispielsweise ein FDX für Phidget-Komponenten [Ph05] entwickelt. Hierbei handelt es sich um elektronische Bausteine in Form von Sensoren und Motoren, die direkt über den Universal Serial Bus (USB) programmiert werden. Aufgrund der STAIVE-Schnittstelle ist es möglich auf sämtliche Geräte und Dienste gleichermaßen zuzugreifen und deren Funktionalität, auch untereinander zu nutzen. Darüber hinaus wurde dieses System in eine existierende Infrastruktur integriert. Das Management aller so eingebundenen heterogenen Geräte wurde aufgrund des jeweiligen FDX als integrierte Management-Komponente möglich. Weiterhin bietet die Integration mittels des von der IT-Management und Web Engineering Research Group entwickelten WebComposition Architecture Models (WAM) die Möglichkeit, das Modell der Infrastruktur in Echtzeit zu verwalten [Me05]. Basierend auf dem Ansatz der Federated Device Assembly und der STAIVE-Schnittstelle lassen sich in Zukunft weitere Management-Komponenten entwickeln um die Interaktion von UPnP-Geräten mit bisher noch nicht abgedeckten Klassen von Geräten auszuweiten.

**Literatur:**

- [Go04] Gsottberger, Y., Shi, X., Stromberger, G., Weber, W., Sturm, T., Linde, H., Naroska, E. und Schramm, P., Sindrion: A Prototype System for Low-Power Wireless Control Networks, 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Oktober 2004
- [He05] Heil, A. Föderatives Dienste- und Gerätemanagement mit Web Technologien, Diplomarbeit, Karlsruhe, 2005
- [In05] Intel Corporation, Intel Tools for UPnP Technologies, <http://www.intel.com/cd/ids/developer/asmona/eng/downloads/upnp/tools/218896.htm?desturl=218892> (letzter Zugriff 27.12.2005), 2005
- [Ma01] Marples, D., Kriens, P. The Open Service Gateway Initiative: An Introduction Overview, IEEE Communications, Vol. 39 No. 12, Dezember 2001, S. 110-114, 2001
- [Me05] Meinecke, J., Gaedke, M., Modeling Federations of Web Applications with WAM, Proceedings of the Third Latin American Web Congress (LA-WEB 2005), ISBN 0-7695-2471-0, S. 23-31, Buenos Aires, Argentina, 31. Okt. – 2. Nov, 2005
- [Mi00] Microsoft Corporation, Understanding Universal Plug and Play, Whitepaper, 2000
- [Mi06] Microsoft Corporation, Microsoft .NET Framework Developer Center <http://msdn.microsoft.com/netframework/> (letzter Zugriff 06.02.2006), 2006
- [Ph05] Phidgets USA, Produkt Webseite, <http://www.phidgetsusa.com/> (letzter Zugriff 06.01.2005), 2005
- [Sc04] Schramm, P., Naroska, E., Resch, P., Platte, J., Linde, H., Stromberg, G. und Sturm, T., A Service Gateway for Networked Sensor Systems, IEEE Pervasive Computing, Vol. 3 No. 1, Januar-März 2004, S. 66-74
- [UP03] UPnP Forum, UPnP Device Architecture 1.0, Version 1.0.1, 2. Dezember 2003
- [UP05] UPnP Forum, <http://www.upnp.org/> (letzter Zugriff 02.01.2006), 2005
- [W301] W3C, Web Services Description Language (WSDL) 1.1, W3C Note, 15. März 2001