ENABLING WORKFLOW IN UPnP NETWORKS

Andreas Bobek¹, Hendrik Bohn², Frank Golatowski³, Gerd Kachel⁴, Andreas Spreen⁵

¹,²,³ Andreas Bobek, Hendrik Bohn, Frank Golatowski, Institute of Applied Microelectronics and Computer Science, University of Rostock, Richard-Wagner-Strasse 31, 18119 Rostock, Germany, (andreas.bobek, hendrik.bohn, frank.golatowski)@etechnik.uni-rostock.de
⁴,⁵ Gerd Kachel, Andreas Spreen, kachel GmbH, Rolandsweg 65b, 33102 Paderborn, Germany, (eerd.kachel, andreas.spreen)@kachel.biz

Abstract— UPnP networks can be enriched by integrating workflow management systems with UPnP capabilities. Thus, interactions between UPnP devices become manageable and UPnP networks can be used for entire technical or business processes. On the other hand the integration of UPnP capability opens new perspectives for workflow management systems (e.g. controlling devices).

This paper describes how to enable workflow management systems in UPnP networks. We identified two levels of integration: workflow management systems that control UPnP networks and workflow management systems that act as an UPnP device at the same time. In the latter case we propose two UPnP services: WorkflowControl service offering actions for creating process instances, actions for controlling and for observing them and ProcessDefinition service allowing installing/uninstalling process definitions. This approach was successfully implemented and tested within an UPnP audio video network. We are not aware of any other approach integrating workflow services into UPnP enabled embedded systems.

Index Terms—UPnP, workflow, SOA, process management.

I. INTRODUCTION

Service Oriented Architectures (SOA) is a keyword in today’s information technology. Currently it plays an increasing role in desktop and business environments and enterprise systems. In the foreseeable future devices will be integrated in SOAs, too. Appropriate technologies are Universal Plug and Play (UPnP) [1] and Devices Profile for Web Services (DPWS) [21]. Devices are represented by their services which are described in a standardised way. Devices just have to enter the network and are ready to interact and communicate with any other device in the network.

A lot of specifications have to be set up for driving SOAs to full capacity: automatic integration of services (publishing and discovery), communication, security, management, workflow, (semantic-rich) descriptions, and reliability. The Information Technology for European Advances (ITEA) [22] project Service Infrastructure for Real-time Embedded Networked Applications (SIRENA) [23] uses UPnP and DPWS as underlying SOAs. Within SIRENA we define a framework that deals with the before-mentioned SOA requirements. In this paper we concentrate on workflow and UPnP networks.

In June 1999, the Universal Plug and Play (UPnP) Forum [1] was founded by Microsoft to set up a service oriented standard for integrating devices and services in a small network area and to define their usage.

Several implementations (UPnP stacks) for different programming languages and operating systems are available such as the stacks of Siemens [2], Cyberlink [3], and Intel [4]. For a nearly complete overview of obtainable stacks the website of the UPnP Forum is recommended.

There are two levels of specifications: UPnP Device Architecture (UDA) [5] defines a generic protocol that has to be implemented by all UPnP devices, and the Device Control Protocols (DCP) define specific device types and service types. Up to now there are several DCPs available such as Printer, Lighting, MediaServer and MediaRenderer. However, protocols that enable workflow systems as UPnP devices are still missing although process management is already an integral part of other SOAs such as the Business Process Execution Language for Web Services (BPEL4WS) [24] in Business Process Management (BPM).

A workflow is a step-by-step operation of a process. These operations can be executed in a manual or in an automatic manner. When workflow is related to information technology it defines a process that is executed by a workflow management system. When designing a process it can be assigned to external systems and applications. Both are accessible by using workflow management system specific connectors (e.g. plug-ins) that have to be installed on a workflow management system in advance or that have to be available at runtime, respectively. Currently, there are a lot of well-defined workflow standards that are widely used. These standards are related to automation of business processes. However, a standardized way for automating technical processes is still missing. Enabling workflow in UPnP networks is a first step to close the existing gap between technical- and business-oriented processes.

There are several works written on using SOAs for process management. The mostly evolved technology in this context is Business Process Execution Language for Web Services (BPEL4WS) which will be used here as a reference to similar technologies.

BPEL4WS uses an XML based language to define business processes that interact with external entities through
Web Service operations. A set of Web Service orchestration concepts are defined to provide external (abstract) and internal (executable) view of a business process. Furthermore, BPEL4WS supports implicit creation and termination of process instances and uses Web Services as the model for process decomposition and assembly.

We are not aware of any process management approach for UPnP although UPnP is a widely accepted SOA for small networks pushed by the home automation and audio/video entertainment domain. Due to that need of UPnP service orchestration we have integrated a workflow management system into a UPnP network. Thereby we applied techniques which are also used in BPEL4WS.

After introducing relevant technologies in chapter II, we identify possibilities how to integrate workflow systems in UPnP networks in chapter III. Chapter IV then presents an application scenario that has been successfully tested.

II. RELEVANT TECHNOLOGIES

In the following relevant technologies SOA, UPnP and workflow management system are explained for further understanding.

A. Service Oriented Architectures

The concept of Service Oriented Architecture (SOA) exists since the beginning of the nineties. A SOA is a system that contains of loosely coupled resources. Resources are e.g. devices or applications. Resources are made available to other participants in the network as independent services that are accessed in a standardized way. In a SOA services can be added and removed dynamically. Service in terms of SOA is an independent functional unit that can be used remotely by a defined interface typically running outside of the own environment.

Well known SOA implementations are Web Services architecture [6], Java Intelligent Network Infrastructure (JINI) [7], Universal Plug and Play (UPnP) and Home Audio Video interoperability (HAVi) [8].

B. UPnP

UPnP supports ad-hoc networking of devices and services by defining their announcement, discovery and usage. Programming languages and transmission media are not assumed. Only protocols and interfaces are specified instead.

The UPnP specification is divided into six phases: addressing, discovery and description specify automatic integration of devices and services; control, eventing and presentation specify how to use them.

UPnP is based on a client-server concept where UPnP devices (servers) offer their services to a network and UPnP control points (clients) can be utilized to find and use them (Fig. 1). Usage of devices is possible through interconnected IP networks, whereas dynamic integration of devices is limited to local IP networks.

In order to participate in networking a device needs a unique IP address. UPnP devices use discovery to announce themselves and their services (Advertisements). Control points use discovery to search for devices and services (Searching), and use Simple Object Access Protocol (SOAP) [9] messages for invoking actions on services.

Fig. 2 shows required protocols of the UDA (e.g. SSDP, GENA and SOAP) and defined protocols by the UDA (Discovery, Control and so on). These protocols are extended by the Standardized DCPs.

![Fig. 1 UPnP network components](image)

![Fig. 2 UPnP protocol stack](image)

C. Workflow

The term workflow refers to the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [10].

The Workflow Management Coalition (WMC) [11], founded in August 1993, is a non-profit, international organization that develops the use of workflow by defining and promoting standards for software terminology, interoperability and connectivity between workflow products.

In a more general definition workflow can be treated as a manual or automated step-by-step operation of a generally known process. As an example consider eBay's [12] trading and rating system from the seller's point of view: The seller has to set up a web site that offers a product. After purchasing the auction eBay sends email to seller and buyer to notify about the upcoming trade. The buyer specifies the method of payment. According to the buyer's selection the seller sends information such as article number, banking account and the sum. When cashed the seller delivers the goods and waits until the buyer makes a rating. The seller's last activity is to make a rating, too. This is a well-known and typical process for eBay sellers. For a long time most of these steps had to be done manually. Accord-
ing to this defining workflow is not new. That’s why it should be explicitly noted that workflows related to information technology and to WfMC are dealing with business processes in an automatic manner (In the case of eBay AfterBuy [13] is a tool that handles the described process automatically).

For that purpose WfMC announced several specifications that are integrated into the overall workflow system architecture (Fig. 3) as follows: Business processes are defined by process definitions using the XML Process Definition Language (XPDL). Process definitions describe the intended process incrementally by putting together activities (e.g. invoking an application) and control nodes (e.g. and-split, and-join). Process definitions are accessible via the Workflow Process Definition Interface and are interpreted and executed by a Workflow Management System. At execution time the workflow management system creates a workflow instance and controls their interaction with workflow participants by using software components running on one or more workflow engines. Communication between different workflow engines is provided by the Workflow Interoperability Interface for supporting chained processes running on different systems from different vendors. Last but not least workflow integration into desktop systems can be implemented by using the Workflow Client Application Programming Interfaces (WAPIs). These interfaces provide capabilities to access workflow management engine services.

III. INTEGRATING WORKFLOW MANAGEMENT SYSTEMS INTO UPnP NETWORKS

We identified two levels of integration: workflow management systems that control UPnP networks and workflow management systems that act as an UPnP device at the same time. In the latter case we propose two UPnP services.

A. Workflow Management System acting as UPnP Control Point

From the workflow management system's point of view the usual way to access external applications is by connecting them through plug-ins (Fig. 4). Applying this approach to UPnP networks the UPnP network, consisting of UPnP devices and services, can be considered as a collection of further applications that can be accessed via the control point that is part of the workflow management system.

Each UPnP device presents an application.

B. Workflow Management System acting as UPnP Device

The WfMC announced several standards that specify how to communicate with a workflow management system as introduced above. Extending the system with UPnP device and service capabilities can be considered as a further standardized way to extend the interoperability of workflow systems.

When implementing an UPnP device, other control points can use the workflow system via UPnP protocol (Fig. 5). For that purpose we propose two UPnP services: WorkflowControl service and ProcessDefinition service. WorkflowControl service offers actions for creating process instances, actions for controlling and for observing them. This service is mandatory when implementing an UPnP workflow device. Actions and their descriptions can be found in table 1.
ProcessDefinition service is optional. It allows installing/uninstalling process definitions. By implementing this service a workflow management system is not limited to its pre-defined process definition repository. Process definitions are specified by passing a parameter that contains a XPDL stream or by referencing a Uniform Resource Locator (URL) [14] that contains a XPDL file, respectively. Actions and descriptions of ProcessDefinition service can be found in Table 2.

### Table 1 WorkflowControl Service

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartWorkflow</td>
<td>Creates a process instance from a given process definition. Takes all given parameters and starts the execution of the process with them. The process is started asynchronously. A resultId is returned and identifies the started process for further handling. This string must be reused when calling the other actions AbortWorkflow, GetWorkflowStatus or GetWorkflowResult.</td>
</tr>
<tr>
<td>AbortWorkflow</td>
<td>Aborts and deletes a process if it is still running or only deletes it, if it is already done. If the process does not exist anymore or already was deleted, then this action will only give an error which can be ignored.</td>
</tr>
<tr>
<td>GetWorkflowStatus</td>
<td>Returns the status of the given workflow as a string. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• “Done” The workflow has been executed successfully.</td>
</tr>
<tr>
<td></td>
<td>• &quot;Executing&quot; The workflow is still running.</td>
</tr>
<tr>
<td></td>
<td>• &quot;Executable&quot; The workflow has been created but did not start running yet.</td>
</tr>
<tr>
<td></td>
<td>• &quot;Blocked&quot; The workflow has been executed but was aborted and or ended with an error.</td>
</tr>
<tr>
<td>GetWorkflowResult</td>
<td>Returns the formal parameters that are defined in the process to be the output. Like the input, all output parameters are given in one string. If there is more than one value, they are separated by comma.</td>
</tr>
<tr>
<td>RunWorkflow</td>
<td>Executes a process synchronously. First starts workflow by creating a process instance from the given process definition. Then starts the execution and waits for the results.</td>
</tr>
</tbody>
</table>

### IV. Application Scenario

In our scenario we demonstrated dynamic integration of the GINGER [15] workflow management system into an UPnP audio video (UPnP AV) network. For that purpose a control point was used to play back various multimedia items on different devices. The ability to create a picture slide show was not implemented by the control point. However, we defined a process that realizes a slide show at the workflow management system in advance. When advertising the workflow system to the UPnP network the control point recognizes it and offers the slide show ability as a dynamically added feature to the user.

The UPnP Audio Video (AV) Architecture:0.83 [16] is part of the standardized DCPs and specifies how to control multimedia devices within UPnP network. For that purpose two device types (MediaServer V1.0 and MediaRenderer V1.0) and four service types are defined. A MediaServer is a device that manages metadata of multimedia files and streams. A MediaRenderer is a device that is able to play back multimedia content such as TV displays, speakers, beamer or Windows Media Player.

![Fig. 6 Scenario UPnP AV network](image)

Our UPnP network consisted of several devices as shown in Fig. 6: Philips Streamium MX6000i [17] is a usual hi-fi system that is able to play back CDs, DVDs and so on. Furthermore it provides its capabilities via an UPnP interface as an UPnP Media Renderer. We implemented a further Media Renderer running on a tablet PC. It provides displaying pictures of various data format and was written to the UniRo UPnP AV interfaces introduced in [18]. The Media Renderer reference implementation CyberMediaGate [19] was used for offering multimedia items to the network. GINGER was used as workflow management system, and both components were running on separate PCs. A further tablet PC acted as an UPnP control point tailored to control AV networks. It was also implemented by University of Rostock. In our implementations we used Java.
and the Cyberlink UPnP stack.

GINGER is a process management system developed by kachel GmbH for the integration of applications and systems based on configurable workflow definitions. Workflows are used for processing activities in form of graph-like flows. Within these workflows systems and/or applications are accessed for the purpose of function calls, of activation, of data access, of data transport or of any combination of these purposes. GINGER provides an open and flexible plug-in mechanism for accessing systems and/or applications, where plug-ins are used to couple an activity to a system and/or application. For configuration and administration of workflow-based applications GINGER provides tools to support definition-time as well as run-time.

For our scenario we extended the GINGER system by UPnP device capabilities as described above (Fig.5) and implemented the WorkflowControl Service. In the run-up we defined a process definition for the GINGER system. To execute the process the following parameters are required: Unique Device Name (UDN) of the target rendering device, a list of Uniform Resource Identifiers (URIs) [20] that refer to the media items and the number of seconds the slide show has to wait before displaying the next picture.

This scenario was successfully demonstrated at the CeBIT\(^1\) 2005 in Hannover, Germany.

V. FUTURE WORK

The Devices Profile for Web Services (DPWS) [21] was announced in August 2004. Subtitle of this document is "A Proposal for UPnP 2.0 Device Architecture". It specifies discovery, control, eventing and description with respect to web services. Therefore, it is a good candidate for further evaluation. The Information Technology for European Advances (ITEA) [22] project Service Infrastructure for Real-time Embedded Networked Applications (SIRENA) [23] deals with a cross-domain (industrial, automotive, telecommunication and home automation) SOA. The SIRENA framework is based on DPWS. A first implementation of DPWS has already been made for C programming language and is highly applicable for small embedded devices. Future work intends to integrate these embedded devices into workflow management systems.

VI. CONCLUSION

Workflow management systems are not limited to automate business processes by using connected applications only. By extending them with UPnP capabilities they also have access to other UPnP devices and can use their services. At the same moment workflow management systems are themselves available as UPnP devices and therefore accessible by other control points. We presented our approach by giving a scenario example and by defining two UPnP services that can be considered as a further standardized way to make workflow systems interoperable. Thus, a first step is taken to close the existing gap between technical and business-oriented processes.

VII. ACKNOWLEDGEMENTS

This work was supported by the SIRENA project in the framework of the European premium cooperative R&D program "ITEA" and BMBF.

VIII. REFERENCES

[16] UPnP AV Architecture 0.83 http://www.upnp.org/standardizedcpss/mediaserver.asp

IX. APPENDIX

A. WMS Device Description Example

```xml
<xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0"><
colorVersion>
<major>1</major>
<minor>0</minor>
</colorVersion>
<URLBase>http://www.kachel.biz:2005/wms</URLBase>
<device>
<deviceType>
urn:schemas-kacheldevice:WMS1
</deviceType>
<friendlyName>
Workflow Management System
</friendlyName>
<manufacturer>Kachel</manufacturer>
```
<manufacturer:URL>http://www.kachel.biz</manufacturer:URL>
<manufacturer:URL>
<modelDescription>
Workflow Management System
</modelDescription>
<modelName>Ginger WMS Wrapper</modelName>
<UEN>uuid:maz-2345-356945-859797</UEN>
<servicelist>
<service>
<serviceType>
urn:kachel:service:WorkflowControl:1
</serviceType>
<serviceId>
urn:kachel:serviceId:WorkflowControl
</serviceId>
<SCPURL>/wfc.xml</SCPURL>
<controlURL>/control/wfc</controlURL>
<eventSubURL>/event/wfc</eventSubURL>
</service>
<service>
<serviceType>
urn:schema=kachel:service:ProcessDefinition:1
</serviceType>
<serviceId>
urn:kachel:serviceId:ProcessDefinition
</serviceId>
<SCPURL>/pd.xml</SCPURL>
<controlURL>/control/pd</controlURL>
<eventSubURL>/event/pd</eventSubURL>
</service>
</servicelist>
</root>

B. WMS WorkflowControl Service Description wfc.xml (extract)

<specVersion>
<major>1</major>
<minor>0</minor>
</specVersion>
<action>
<name>GetWorkflowStatus</name>
<argumentList>
</argumentList>
</action>
</scp>