A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

The Internet of Things – Context-based Device Federations

Andreas Heil[1], Mirko Knoll[2], and Torben Weis[2]

Microsoft Research Cambridge
Cambridge, United Kingdom
v-aheil@microsoft.com

Universität Stuttgart
Stuttgart, Germany
{knoll,weis}@ipvs.uni-stuttgart.de

Abstract

Locating physical devices and accessing their functionality in pervasive environments is strongly restricted by their locality. Since there are more and more devices available, waiting to serve users, a simple way is required to access, connect, and most important to locate them in the real world. In this paper we propose an approach to connect and access arbitrary devices by federating them related to their geographical location. This approach allows us to create context-aware federations of devices – the internet of things.

1. Introduction

Modern life is strongly affected by an increasing number of computing devices in work and domestic environments. Such devices usually belong to individuals or are bound to a specific location such as a certain office space. Consequently, the usage of these devices is predominantly restricted to the proximate neighborhood of these devices. Even if the configuration of such a neighborhood is usually static, it changes from the user's point of view, every time he or she enters or leaves a certain area. Since the user's personal devices are not aware of the nearby setup of devices, locating, pairing, connecting, and thus using these devices and their services is almost not possible. Furthermore, it is usually limited to the direct physical usage of the corresponding appliance. Techniques beyond common management and administration systems, directly linked to location-based services (LBS), are necessary to satisfy the increasing amount of devices in our work and domestic environments, ready to be interconnected to each other.

Advances in technology have led to an increasingly interconnected world. As a global backbone for information exchange, the internet provides a vast amount of various services. Despite this potential, the internet only moves slowly from a static archive of information to a dynamic interactive system. At the same time, the development in interconnecting devices is constantly increasing. Established industrial home automation standards, such as EIB/KNX [9] or LonWorks [3] using proprietary communication, have recently adapted technologies compatible with global networking standards such as Ethernet. Also standards such as UPnP [20] focus on interconnecting a wide variety of devices and services by providing an overlay network on existing infrastructure such as a domestic network or corporate intranet. To exclude competitors and increase market share, manufacturers design these industrial standards mostly incompatible. Furthermore, bus-systems as the examples given above are usually restricted by physical boundaries such as buildings. To interconnect devices among different buildings direct wiring is required but limited to a certain distance by the physical decrease of signal strength. Even if connected, the problem remains the same as for systems driven by software: Exposing and consuming functionality of devices among logical structures e.g., beyond organizational boundaries has not been solved yet. This paper aims to federate functionality with various spanning domains. Unlike pure services, devices are bound to their physical place. In the following we want to investigate how one can federate these devices and their functionality while presuming they are located somewhere in our physical world.

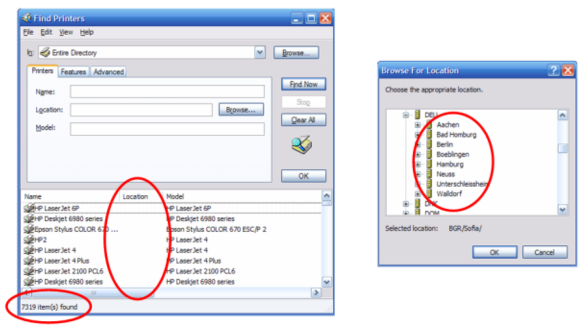
The paper is organized as follows. In Section 2, we motivate our research. In Section 3, we investigate existing technologies suitable to our approach. We discuss possibilities in localizing services and devices using the FDX approach and its decentralized extensions in Section 4. Implementations to verify our research are covered in Section 5, followed by an overview of related works in Section 6. We conclude this paper with a summary of the ongoing research and an outlook on future work.

A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

2. Motivation and Background

Regardless of the technological improvements during the past years, most of the organizational and domestic environments are restricted to a distinct location such as a room, a floor, or a certain building. The common virtual limitations that can be used to map such an affiliation and to locate and allocate resources are organizational boundaries. The drawbacks of such boundaries can clearly be seen in the following example:

Bob, an employee of a large company arrives at a corporate office in a foreign country. In order to print documents, he has to choose and install an appropriate printer, which is located next to him. Browsing the company's printer directory, he can select a printer. In large-scale systems, search queries on metadata (e.g., name or location) support the user in finding the required device.



(a) Find Printers (b) Browse For Location

Figure 1. Locating printer devices

However, these queries can only be performed if the corresponding metadata has properly been entered in the corresponding directory service (e.g., Microsoft's Active Directory). As depicted in Figure 1 (a), without knowing the printer's name and missing metadata about the location, Bob cannot fall back on this search criterion. Thus, h has to browse a directory of several thousand printers connected to the corporate network worldwide.

Figure 1 (b) shows a location browser, allowing Bob to further specify his location. This decreases the amount of printers, but still may result in a larger amount to choose from. Consequently, he has to either trial and error, consult a local employee for the printer's name, or check out the device's name by himself. Still, it is not guaranteed that Bob has been granted rights to print on the selected device even if he was able to install the corresponding device drivers. Despite this, another printer located directly next to him in an office room nearby cannot be accessed at all since the printer belongs to a different organizational unit. Nevertheless, the desire of Bob "Print the document on the printer, located next to me I have access to!" requires a lot of user interaction and is somewhat error-prone. Location-based services might help the user in the situation described before.

3. State of the Art

Federating real-world devices and their services is an upcoming research area since these devices are usually strongly restricted in their functionality with regard to their physical location. Thus, we propose a way of optimizing the federation of physical resources towards their locality to reduce latency, accessibility issues, and security problems. To the best of our knowledge, no other approach applies the localization of devices to the modeling of federations. We discuss various existing technologies suitable to describe the interaction of location-based systems and show the drawbacks when used in our scenario.

3.1 Service Federations

The concept of federating services is a topic of interest especially in the Web Engineering community [11] [10]. The main benefit of this approach is to federate services (e.g., Web Service) across organizational boundaries. The concepts developed there are very flexible and easy to implement. Furthermore, essential methods such as security and accounting are provided. Based on this idea the concept of Federated Device Assemblies (FDX) [5] is an approach to integrate real-world devices into service federations. Therefore, a FDX is designed to encapsulate the capabilities of devices. These capabilities are summarized as operations, status variables, and events. Using a XML Web Services these three member types are exposed through a uniform interface called STAIVE interface. It is possible to access status variables and invoke functionality provided by the device as well as to subscribe to events by this interface.

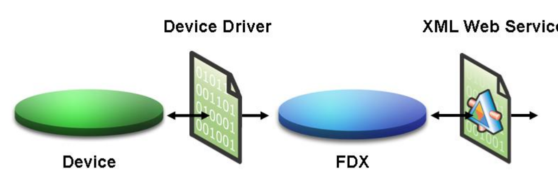


Figure 2. FDX Communication

A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

Since the underlying technology is already capable of crossing organizational boundaries such as corporate or home network borders, the device's functionality can be exposed in the same way.

The FDX approach allows encapsulating single devices as well as complete classes of devices. A reference implementation for managing devices following the UPnP standard is described in [7]. As a result, each device or software supporting the FDX approach may communicate with the UPnP FDX and as such with any available UPnP device within the network. Furthermore, a major drawback of UPnP is neutralized since the FDX approach allows to federate UPnP devices across local network borders.

To build and run a FDX federation, several centralized infrastructure services are required. These services consist of a repository for discovering and managing available FDX services. The so-called WAM Service stores information about each FDX within the federation. Lookups for available FDX and their functionality is therefore always directed to the WAM Service. However, a failure of these centralized services may lead to severe problems within the federation.

3.2 Modeling Federations

The WebComposition Architecture Model (WAM) [10], [11] is especially designed for modeling loosely coupled service-based architectures. Furthermore, the WAM supports integrating devices (actuating as well as sensing devices) into the model [5]. Beyond federating, the WAM also allows for creating models exceeding organizational boundaries, so-called realms, as depicted in Figure 3. Communication between these realms takes place by adding so-called Invocation Profiles. These allow a more detailed description of the invocation process among entities such as services and applications. To keep the WAM diagram simple, the description for such an Invocation Profile is not part of the model itself. Detailed information is maintained separately and referred to by a named link as seen in Figure 3, which depicts the example in Section 2: Bob's application (Bob's App) accesses a print service (PS1) to print his document. By querying the Localization Service (LS1) before, Bob can determine the printer next to him.

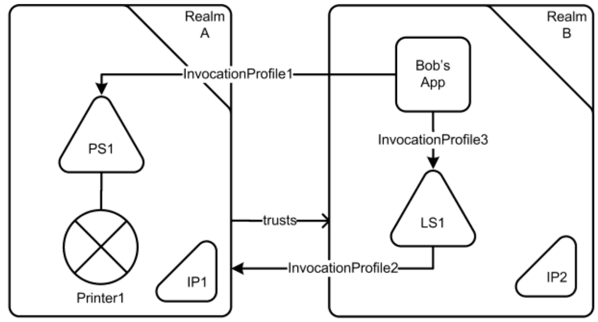


Figure 3. WAM example diagram

Beside the graphical notation, the WAM is based on a XML-based machine-readable format. Thus, the models can be automatically processed within different programs and are not bound to a specific type of application. While this approach allows the decentralized design of intra- and inter-organizational interconnections of services, the approach does not yet support the localization of services and devices used in the model. In the following sections, we describe our approach, which closes this gap.

3.3 P2P Infrastructure

Recently more and more centralized services get replaced through the use of peer-to-peer (P2P) technology. One of the main benefits of P2P is the ability to self-organize. This allows to abandon all centralized servers as all data now is distributed fairly among all peers.

Most common P2P systems optimize for availability or latency, such as Pastry [15]. We use Geostry [8] because of its optimization towards locality of data. This means that information about a certain location will be stored on a node, which is actually close to this location. To achieve locality, the world is partitioned into sectors of equal size, each one represented by a unique ID. The ID will be given such that two devices, which are geographically close, will be numerically close in this P2P-ID space, too. In doing so, we find one of the nearest devices, which is available at a certain coordinate. Currently the ID is realized as 56-bit encoded suffix that allows for covering areas of less than 0,007m². This is sufficient to even manage a high population of devices. The nodes are then organized in a ring structure similar to Pastry to allow for an efficient routing in $O(\log_n)$.

A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

4. The Extended FDX Approach

In our previous work [5], we proposed the WAM Service as substantial infrastructure element to discover, store, and manage information about available Federated Device Assemblies (FDX) respectively devices. The proposed solution proved to be well suited for service and device federations within a reasonable complexity. In large-scale systems, the required investment in hardware and work force in form of infrastructure management tasks can be very high.

As described above, FDX are already designed to communicate among each other irrespectively of the hardware addressed underneath. Thus, introducing peer-to-peer (P2P) technology for FDX management tasks enables us to reduce infrastructure costs significantly. In addition, through this extension the FDX solution evolves towards a self-organizing, context-based approach, detached of any concrete infrastructure. We can now eliminate the drawback of accessing a centralized server to store and retrieve information about available FDX from the master repository. Now, each FDX provides a certain contingent of bandwidth, CPU time, and disk space to the P2P system, thus creating a distributed service repository, which puts the federation into effect.

First step in this approach is to determine the area of interest. For this, we intend to store the distributed information about FDX predominantly in other FDX located geographically next to each other. Therefore, we extend the Geostry approach (cf. Section 3.3). The way nodes are organized in the P2P structure serves as fundamental algorithm to organize the FDX network. The complete process can be briefly summarized as follows: After booting, the FDX determines its current location, derives the corresponding ID and joins the Geostry ring.

4.1. Managing FDX

In this Section, we introduce the new components, which were inspired by the approaches introduced in Section 3. Figure 4 further illustrates this concept. Several new components are presented to achieve location-awareness, which every FDX requires to establish or join the Peer-to-Peer network. The *Location Manager* represents the fundamental component to determine a FDX's location. Several ways are possible to accomplish this task: For stationary FDX the location can be stored within a configuration file. The FDX reads the information at startup to determine its unique identifier. If the host system or the attached devices provide any location specific information such as Global Positioning System (GPS) [19] or Global System for Mobile Communications (GSM) [2] data, the Location Manager reads this information from the specific device. The formerly used WAM Service, which has been used to store and manage information about available FDX within a federation, is now replaced by a decentralized organization of FDX. Therefore, each FDX contains a *FDX Manager*. Besides the bootstrapping, the FDX Manager handles incoming requests about the availability of FDX and the registration of newly available FDX within the federation. New FDX are either stored in the local FDX Manager or forwarded to another, already known FDX, if the distance between the identifiers of the new FDX an the known one is less that the distance to the current one: $|idOther - idNew| < |idSelf - idNew|$.

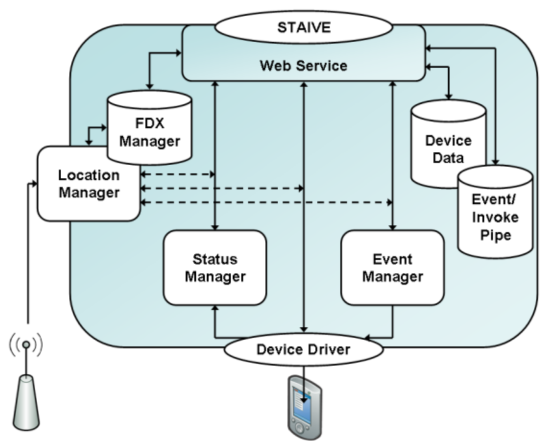


Figure 4. Extended FDX

If a request about a specific FDX α is sent to the current FDX and there is no information within the local FDX Manager available, then the request is forwarded to the known FDX with the least distance to the FDX α. Since each STAIVE respectively FDX operation's response consists of a status code (cf. Table 1) the FDX returns status code 303 and the Uniform Resource Locator (URN) of the FDX.

Status Code	Message Text
200	OK
201	Created
204	No Response
301	Moved
302	Found
303	See Other
404	Not Found

A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

Table 1. FDX status codes

The request is forwarded up to a FDX that can respond (status code 302). If no information can be found and forwarding is no longer possible, the FDX responds with status code 404. An exemplary invocation sequence is depicted in Figure 5.

The third component added to the Extended FDX architecture is the local *Event/Invoke Pipe* (EIP). Subscribing to an event takes place in form of submitting a callback reference to the subscribed FDX. These callback references are stored within the local EIP.

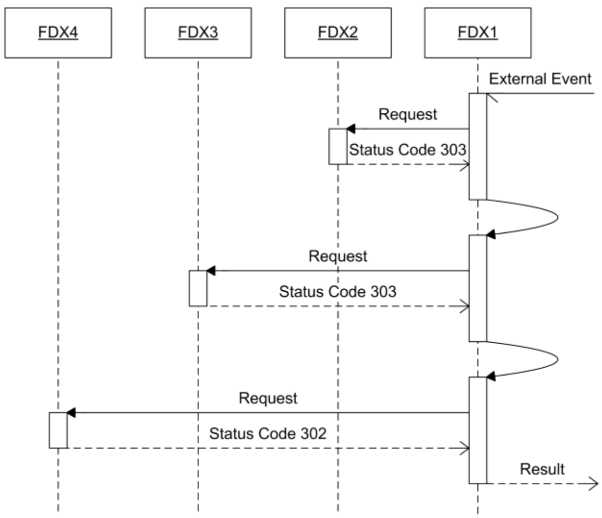


Figure 5. Request forwarding

To federate the FDX, detailed knowledge about its certain location is necessary. Thus, in the next section we discuss the usage of location-based identifiers with the Federated Device Assemblies to determine global identifiers based on the approach introduced in Section 3.

4.2 Location-based FDX Identifier

Originally, each device connected to a FDX received a unique identifier from a central WAM Service. The service URL and a unique device identifier allow for direct addressing of each device within the federation. Furthermore, each operation, status variable, and event name together with the service URL and the device identifier form a unique identifier for each operation, status variable, and event within the network in form of a URN. The identifier for each device assigned by the WAM Service consists of a 128-bit Globally Unique Identifier (GUID). Following this approach, we continue specifying this 128-bit identifier more in detail for the usage within the Extended FDX approach. The lower 56 bits are used to store the location identifier. This part of the identifier is maintained by the FDX and thus used as a base for each identifier of attached devices. The FDX also is in charge to assign the appropriate 72-bit prefixes to the attached devises (e.g., serially numbering) to create a unique identifier for each device (cf. Figure 6). A range of 2^{72} identifiers for devices is appropriate even for large-scale sensor networks managed by a single FDX.

Figure 6. Unique identifier

By moving towards a decentralized infrastructure, the identifier cannot be assigned by one single infrastructure service anymore. However, the location identifiers are distinctive and the device identifier is determined by the FDX itself, the composed identifier remains unique. Furthermore, each FDX can resolve the identifiers for its devices on its own and the centralized infrastructure service is obsolete for this task. To ensure compatibility with the original FDX approach, the URN namespace identifier *sdfid*, used by the formerly approach, is replaced by the namespace for the Extended FDX *fdxid* as depicted in Figure 7.

```
extendedIdentifier ::= id "::" type ":" name
id ::= "urn:" nsIdentifier ":" guid
nsIdentifier ::= "sdfid" | "fdxid"
type ::= "var" | "function" | "event"
name ::= letter | "_" {letter | digit | "_"}
```

Figure 7. Extended identifier definition

Thus, a URN with a determined identifier can be clearly identified as such and does not conflict with a maybe already existing identifier with a randomly created GUID.

4.3 FDX Location Updates

Updating the FDX's location immediately has an effect on its location identifier and thus on the service and device URN. To minimize traffic caused by updates, old and new identifiers are kept simultaneous for a certain amount of time. Different timeout periods are currently being evaluated. The invoked service does disclose its identifier by submitting status code 301 and its URN including the new determined identifier. Thus, the calling FDX can update its internal repositories to the new location identifier of the invoked FDX.

In this Section, we have introduced the extensions necessary to enhance the FDX approach towards a location-based P2P system. In the next section, we present the prototypes we used to test the functionality of device federations in a distributed network.

5. Context-based Device Interaction

Context can be understood as a combination of the location of entities [18] and the current environment and situation [14]. The FDX approach already allows determining this environment since each FDX can request the functionality description of any other FDX. Combined with the newly introduced FDX extensions a FDX can determine its own location and find any other FDX within its area of interest. Thus, the federating device functionality can be particularized in more detail. Functionality within a federation, but beyond the area of interest, may thus be ignored. Referring to our example from Section 2, the system can be designed to ignore any printer available on the corporate network until it is within a range of at most 15 meters from the user.

Even if a FDX does not understand the semantic meaning of the device's functionality yet, a FDX can be programmed to use a publish/subscribe algorithm, allowing to subscribe to a specific kind of sensor event within its area of interest without particularly knowing each sensor. Regarding the example from above this could also be the next printer in range. Thus, the subject (e.g., Bob with his Laptop) can move through an environment, always connected with the nearest printing device.

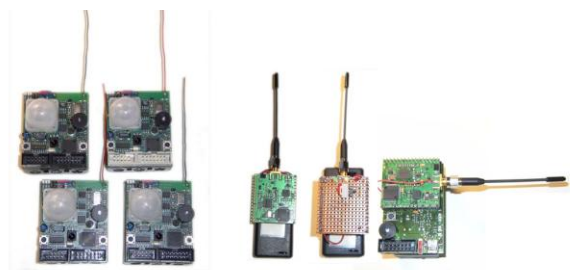
5.1 FDX Prototypes

In order to validate the Extended FDX Approach we modified one existing FDX and developed an additional one to support the context-based device interaction.

Interface Kit 8/8/8   Text LCD Display

Figure 8. Supported Phidget devices

The existing FDX chosen is the one supporting Phidgets [13]. From the already supported devices we focus on the *Phidget Text LCD* and the *Phidget Interface Kit 8/8/8* (cf. Figure 8), providing several different sensor inputs. The second class of devices supported within the Extended FDX approach are ScatterWeb [4] sensor boards. The Embedded Sensor Boards (ESB) provide several on-board sensors (cf. Figure 9) such as temperature, luminosity, noise, vibration, and movement sensors. In addition, each board contains a real-time clock, IR sender and receiver as well as a sender and receiver for wireless communication. The manufacturer provides an API supporting Microsoft's .NET Framework. The API grants access to the sensor boards on a very high abstraction level. Thus, the programmer can focus on the program logic and does not have to deal with low-level hardware issues such as power management, sleep mode, or field strength for transmissions. Finally, the sensor boards are self-organizing and the developer has not to deal with routing of sensor information within the wireless sensor network (WSN).



(a) ScatterWeb ESB   (b) ScatterWeb ESB Prototypes

Figure 9. ScatterWeb ESB

We also support the first prototypes of the newly introduced ESB prototypes. These boards are reduced in size and provide a modular setup of sensors. The new prototypes are already prepared to support a combined localization of GPS and GSM to provide fast and accurate information about the current location of each sensor board. This localization technology is already in use at student courses at the Freie Universität Berlin, Germany and will be included in the ScatterWeb FDX with the next hardware update.

5.2 Federated Interaction Modeling

Figure 10 shows a first prototype to model the interaction of devices within a federation. To federate the devices and model their interaction, we built this visual designer. This especially allows non-technical users to play with the system easily.

A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

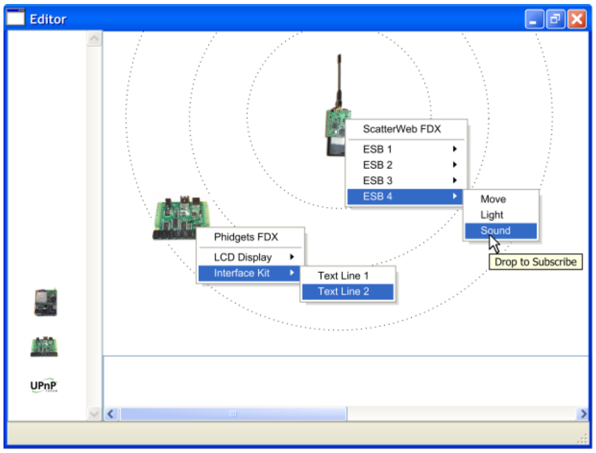


Figure 10: Interaction Editor

Each FDX, available within a federation can be accessed by it's icon on the left-handed toolbar. After choosing a FDX icon from this toolbar, the user can specify the area of interest, and thus the geographical distance, other FDX within the federation are in. The user can select each device, attached to a FDX. Operations exposed by these devices can be subscribed to events of FDX within the area of interest.

Removal of subscribed FDX from the federation (e.g., by switching off, running out of battery power or by a defect of the hardware) usually requires additional effort to keep the subscription list consistent. Applying the approach, introduced with the local Event/Invoke Pipe in Section 4.1, subscriptions have not to be removed after a FDX within the federation got unavailable within the federation. Instead the affected subscriptions are not removed necessarily but can stay within the system until the corresponding FDX is available again.

The main contribution of our approach is to connect the functionality of devices to each other corresponding to their physical location. To explain the process using the prototypes described above, we renew our example from Section 2. To simplify matters we use the term service for both software services as well as device functionality exposed by a service. Two main activities can be identified during his process: (a) Identifying the service the user needs and (b) locating the corresponding services providing this functionality. Consequently Bob selects the service he wants to use first. As Bob wants to print a document, he chooses a printing service located on his laptop. He also specifies the maximum distance to a printer he is willing to walk. Since the printing service describes its functionality through the STAIVE interface, the system now can contact all services within range and match the interface signatures of the printing service with those exposed by the services within the specified range. The available services can now be displayed on the visual designer where Bob can select the printing device he wants to use. This example shows the benefit of our approach: Identification and matching of device functionality is easy to accomplish since the STAIVE interface provides the necessary information. The FDX approach then allows to pair and connect devices that are not initially designed for this purpose. Since the Extended FDX architecture is already based on a federated concept, implicit appropriateness of security and accounting mechanism can be applied. Finally the main contribution of this paper, the localization of the requested resources within a federation of devices and services can be easily accomplished.

6. Related Work

Existing approaches can rather be attributed to one of the following purposes: providing geo-centric ('Where is …') or location-based ('What is near me ?') services. MSN Virtual Earth [12] or Google Maps [6] serve as a good example for the upcoming interest in geo-centric services. They provide information about real-world services, which are enlisted in corresponding directories. Furthermore, databases such as GeoURL [17] allow for locating services such as weblogs regarding their physical location in form of geographical coordinates. On the other hand, there are projects that tend to pursue the location-based approach, such as the SenseWeb [16] project. Its target is the aggregation of the growing amount of sensors by allocating and exposing geo-centric sensor data. The Amigo [1] project follows our idea of unifying devices from different domains, such as the mobile domain or devices from home automation.

However, the projects described above neglect to take into account, that devices may belong to several owners and therefore may not be easily federated.

The drawback is that these services access centralized databases and directories to provide information about real-world services at a certain location. Information about these services within these directories has to be inserted explicitly.

7. Conclusion and Future Work

In this paper, we proposed a location-based approach for federating services and devices. In contrast to existing approaches in the field of location-based and geo-centric services, we look beyond federating devices available at a certain area. Our main contribution is the ability to connect the functionality of devices corresponding to their physical distance, regardless their ownership. Thus, allowing Bob to print on the closest printer available. We achieved this by extending our FDX approach through the use of P2P technology. This results in a higher availability and stability as the system no longer depends on centralized components. We introduced two prototype implementations and a visual tool usable within our approach. In the future, we plan to extend the Phidget FDX to support a wider range of Phidgets devices and to implement the location functionality of the ScatterWeb ESB with the next hardware update. We also work on the rule-based engine allowing to refine the FDX internal behavior.

Since we showed the feasibility of the technical realization, we will undertake further steps in extending and formalizing the concepts of modeling location-based services. The focus of our future research will be on modeling location-based services. The WAM approach introduced in Section 3 seems to be the best fitting solution. Already providing information about organizational boundaries and devices within, the WAM Modeling language is designed to be easily extended with additional layers.

8. Acknowledgement

Mirko Knoll was funded by DFG Excellence Center 627 'Nexus'.

9. References

[1] *Amigo Short Project Description*. 2005, Information Society Technologies.

[2] 3rd Generation Partnership Project, *3GPP Homepage* - Website (2006): http://www.3gpp.org/ (13.06.2006).

[3] Echolon Corporation, *Introduction to the LonWorks System*. 1999.

[4] Freie Universität Berlin, *ScatterWeb* - Website (2006): http://scatterweb.mi.fu-berlin.de (14.06.2006).

[5] Gaedke, M., Meinecke, J., and Heil, A. *FDX – Federating Devices and Web Applications*. In Sixth International Conference on Web Engineering (ICWE). Palo Alto, USA. 12-14th July 2006.

[6] Google, *Google Maps* - Website (2006): http://maps.google.com/ (15.06.2006).

[7] Heil, A. and Thiele, C. *Alles Verbunden - Von UPnP zur Geräte-Föderation*. In Informatiktage 2006 - Fachwissenschaftlicher Informatik-Kongress. Bonn, Germany. 31. Mar - 1. Apr, 2006: Gesellschaft für Informatik.

[8] Knoll, M. and Weis, T. *A P2P-Framework for Context-based Information*. In 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI) at Pervasive 2006. Dublin, Ireland. 07 May 2006.

[9] Konnex Association, *System-architecture* - KNX Standard Public Document (2004): http://www.konnex.org/uk/downloads-service/downloads/ (09.09.2006).

[10] Meinecke, J. and Gaedke, M. *Modeling Federations of Web Applications with WAM*. In Third Latin American Web Congress (LA-WEB 2005). Buenos Aires, Argentina. 31 Oct - 2 Nov, 2005: IEEE Computer Society.

[11] Meinecke, J., et al. *Capturing the Essentials of Federated Systems*. In Fourteenth International World Wide Web Conference (WWW). Edinburgh, UK. 23-26 May 2006.

[12] Microsoft Corporation, *Windows Live Local* - Website (2006): http://local.live.com/ (15.06.2006).

[13] Phidgets Inc., *Phidgets* - Website (2003): http://www.phidgets.com/ (15.02.2006).

[14] Rodden, T., et al. *Exploiting Context in HCI Design for Mobile Systems*. In First Workshop on Human Computer Interaction with Mobile Devices. Glasgow, Scotland. 21 - 23rd May, 1998.

[15] Rowstron, A. and Druschel, P. *Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems*. In 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). Heidelberg, Germany.

A. Heil, M. Knoll, T. Weis; The Internet of Things – Context-based Device Federations, Proceedings of the 40th Annual Hawaii International Conference on System Science 2007 (HICSS 2007), Pages 58-58, Waikoloa, HI, USA, Jan 3 – 6 2007.

[16] Santanche, A., et al., *SenseWeb: Browsing the Physical World in Real Time*, ACM/IEEE IPSN 2006, Nashville, TN, USA, April 2006

[17] Schachter, J. and Hansen, A.B., *GeoURL 2.0* - Website (2006): http://geourl.org/ (14.06.2006).

[18] Shilit, B.N. and Theimer, M.M., *Disseminating active map information to mobile hosts*. IEEE Network, 1994. 8(5): p. 22-32.

[19] U.S. Coast Guard, *GPS SPS Signal Specification, 2nd Edition* - Website (1995): http://www.navcen.uscg.gov/pubs/gps/sigspec/default.htm (13.06.2006).

[20] Upnp Forum, *UPnP Forum* - Website (2006): http://www.upnp.org (06.02.2006).