AUTONOMOUS FILE SHARING FOR SMART ENVIRONMENTS

Jussi Kiljander, Matti Eteläperä, Janne Takalo-Mattila, Juha-Pekka Soininen
VTT Technical Research Centre of Finland, Kaitoväylä 1, Oulu, Finland

Kari Keinänen
VTT Technical Research Centre of Finland, Vuoriniementie 3, Espoo, Finland

Keywords: Smart Environment, Smart-M3, File Transfer, Semantic Interoperability, Ontology.

Abstract: Smart Environment is a physical place where different kinds of devices interact meaningfully with each other to assist and support us in our everyday life. A key requirement for enabling these kinds of smart systems in physical spaces is the ability to share files autonomously between various devices in the environment. In this paper a novel solution for autonomous file sharing between heterogeneous devices is presented. Here autonomous means that the devices interact with each other seamlessly and the heterogeneity of devices and services providing file sharing is hidden from the end user. Our approach is based on presenting information about the files and file sharing protocols by using ontologies and we utilize Smart-M3 as the platform for sharing semantic information between devices in a physical space. To demonstrate our approach for autonomous file sharing in practice we have implemented a meeting application to various mobile platforms.

1 INTRODUCTION

Our environment is full of heterogeneous electronic devices such as computers, mobile phones and home appliances, for example. Pervasive and Ubiquitous Computing are computing paradigms aiming to enable Smart Environments where these devices interact meaningfully with each other in order to provide useful services for people in the environment (Weiser, 1991). Files are a significant part of persistent storage in modern computer systems and file sharing is an integral part of many inter-device applications. These applications include the Web and email, for example. It is also apparent that many Smart Environment applications would require devices to share files such as videos, music, images, documents and even software with each other. File sharing is therefore also an important part of interaction between ubiquitous devices and a common solution for file sharing in Smart Environment is needed.

A typical scenario of file transfer in Smart Environment can be divided into two parts: discovery of file sharing services and execution of a selected functionality of the service. The discovery functionality is usually provided by some Service Oriented Architecture (SOA) solution such as Universal Plug and Play (UPnP), Web Services or Common Object Request Broker Architecture (CORBA), for example (UPnP forum, 2008), (W3C, 2002), (OMG, 2002). Also other non-SOA solution for service discovery exist e.g. Bluetooth Service Discovery Profile (SDP) and the Zeroconf standard (Cheshire and Steinberg, 2005).

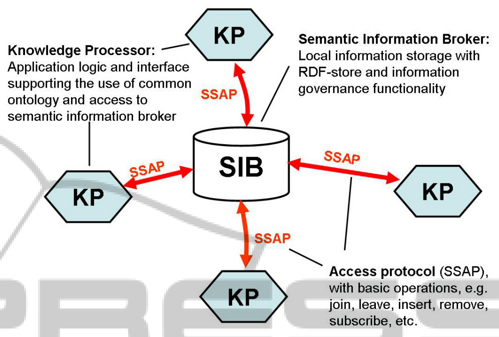
In addition to discovering the file services, a device must be able to interact with the services to execute specific file transfer functionality such as to store, modify or request a file, for example. For this purpose many standard file sharing protocols based on the traditional client/server paradigm exists. These protocols include the File Transfer Protocol (FTP), Secure Copy (SCP) and Apple Filing Protocol (AFP), just to name a few. Additionally numerous SOA solutions have their own methods for describing services that provide file sharing capabilities. As an example UPnP requires a priori standardization of the service interfaces were as the Web Services and CORBA, for example, utilize technologies such as Web Service Description Language (WSDL) and Interface Definition language (IDL) for describing the service interfaces in a machine interpretable format.

As can be seen there are many service solutions and protocols available that provide file sharing capabilities for devices. In addition the variety of communication technologies such as the ZigBee, Bluetooth and Wireless Local Area Network (WLAN) creates their own challenges for autonomous file sharing in Smart Environments. With a large amount of different communication technologies, file sharing technologies and SOA-based file sharing services it can be a difficult both to find all available file transfer services and to interact with the services to perform the required operation. Basically this requires that each device sharing a file must use all discovery mechanisms it supports for finding the available file services and then publish the file to each service it is capable to interact with. Additionally to interoperate with devices that do not utilize SOA solutions the devices are also required to advertise the file with each communication technology it supports. It is apparent that this creates a lot of overhead both to the code size of applications and to the communication between the devices.

In this paper we propose a novel solution for autonomous file sharing with heterogeneous devices, file transfer methods and SOA solutions by utilizing an interoperability platform called Smart-M3 on top of these existing solutions. As presented by (Liuha, Soininen, Otaloea, 2010) the Smart-M3 achieves semantic interoperability between devices of Smart Environments by providing architecture for utilizing ontology-based information presentation in a physical space context. In our approach we present a common ontology for defining information about the available files, file transfer protocols and file services in semantic form and publish this information available to all devices in the Smart Environment. This enables devices sharing the common ontology to negotiate about the possible solutions for file sharing and then select the appropriate method for transferring the file.

2 SMART-M3

Smart-M3 is an independent information level interoperability architecture that can be implemented on top of any communication or service level solution. The fundamental idea of Smart-M3 is to utilize the Semantic Web ideas of ontologies and semantic interoperability in a physical space context (Berners-Lee, Hendler, Lassila, 2001). In addition to the ontology-based interoperability model Smart-M3 defines a functional architecture that specifies the methods for accessing the semantic information in a physical space. The Smart-M3 functional architecture consists of Knowledge Processors (KP) and Semantic Information Brokers (SIB). Fig. 1 presents the Smart-M3 concept including the core elements and their relations.



Knowledge Processor: Application logic and interface supporting the use of common ontology and access to semantic information broker
KP
Semantic Information Broker: Local information storage with RDF-store and information governance functionality
SSAP
SIB
SSAP
SSAP
KP
KP
SSAP
KP
Access protocol (SSAP), with basic operations, e.g. join, leave, insert, remove, subscribe, etc.

Figure 1: Smart-M3 Functional Architecture.

The Smart-M3 functional architecture is based on publish/subscribe paradigm and therefore it enables reactive event-based communication between KP and SIB entities. SIB is the information repository of the Smart Space and it provides operations for modifying and requesting the semantic information. KPs form the actual application for the end user by utilizing the SIB service for sharing semantic information with each other. In order to hide the complexity of both information presentation and communication from the application logic, the KP is divided into KP Interface (KPI) and use case logic parts. KPs that interact with each other via SIB are interoperable if they have a common ontology that specifies the concepts and relationships between the concepts in a given domain.

Smart Space Access Protocol (SSAP) is the communication protocol of Smart-M3 and it defines the rules and syntax of operations for accessing the semantic information in the SIB. The SSAP specifies operations such as insert, remove and update for modifying the semantic information as well as various formats of query and subscribe operations. To ensure that the content of the SIB does not change during the execution of any operation each SSAP transaction is executed in an atomic fashion. In addition to these operations the SSAP contains join and leave operations that are used to provide access control and authentication. The information access and security in Smart-M3-based systems is further described in (Suomalainen, Hyttinen and Tarvainen, 2010).

In Smart-M3 the Resource Description Framework (RDF) and RDF-Schema (RDFS) technologies are the backbone of the ontology-based interoperability model (W3C, 2004). RDF is a family of W3C specifications for modelling metadata in form of subject, predicate and object triples. Because RDF triples are similar to basic sentences of a human language, it is a natural way to make statements about information with certain properties and relationships. RDFS is an extensible knowledge presentation language that defines vocabularies for building simple ontologies using RDF. The current reference implementation of the SIB provides three languages for querying (and subscribing to) the RDF data: Triple, SPARQL and Wilbur Query Language (WQL) (W3C, 2008), (Lassila, 2002).

3 APPROACH FOR AUTONOMOUS FILE SHARING WITH SMART-M3

In our approach we divide the challenge for autonomous file sharing into three distinct levels: communication, service and information level. Fig. 2 presents a logical view of our approach.

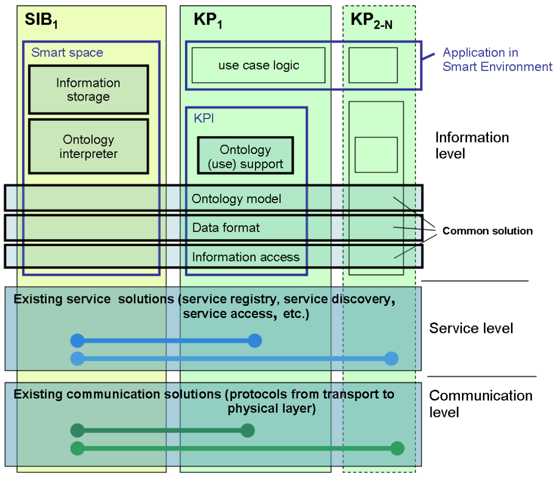


Figure 2: Logical view of approach for autonomous file sharing in Smart Environments.

The communication level covers the Open System Interconnection (OSI) model layers from L1 to L4. It provides the basic functionality required for transmitting bytes of data between devices.

The role of the service level is to provide technologies for devices to share services with each other. The Service level covers existing SOA solutions and protocols. In the file sharing context these protocols include the FTP, SCP, Bluetooth's File Transfer Profile and BitTorrent, for example. In addition there are SOA solution such as UPnP, Web Services, OSGi and NoTA that have their own services for providing file transfer capabilities (OSGi Alliance, 2005), (NoTA World, 2007).

Information level is the highest level in our approach. The role of the information level is to make information of devices available and the semantics of the information interpretable for other devices in the environment. We propose Smart-M3 to be used in the information level for sharing semantic information required in the file sharing context.

In our approach necessary information about files, transfer protocols and services is modelled as a common ontology and the SIB is used as a common search extent of this semantic information. This way the devices are able to discover the available files in a given environment and then select the most appropriate file transfer technology for transferring the file between devices.

We utilize RDFS vocabulary as the ontology language in our approach. This is because the current implementation of the SIB with WQL supports RDFS level reasoning (also the owl:sameAs and owl:InverseFunctionalProperty features of the Web Ontology Language (OWL) are supported) so the RDFS is the most practical choice. Fig. 3 illustrates the ontology of our approach.

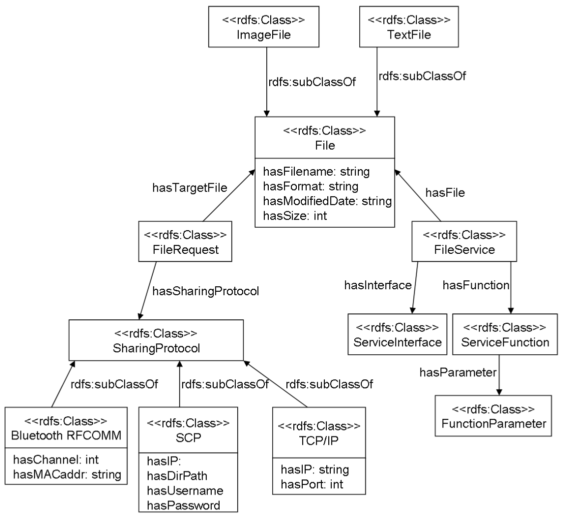


Figure 3: File Sharing Ontology.

The File class is the base class for all files. It contains basic properties describing the name, format, size (in bytes) and modification date of the file. The ImageFile and TextFile are example subclasses of the File class that can be used to describe the type of the file more specifically. It is of course possible to introduce new subclasses for the File class and for these subclasses as well. The FileService class is the class for all file transfer services. The ServiceInterface class specifies the interface of services. This is done through the ServiceFunction and FunctionParamater classes. In addition to accessing files through SOA services, we also developed an elegant way to request files by interacting only on the semantic level. For this purpose the FileRequest and SharingProtocol classes have been defined. FileRequest class presents a request to a file in the SIB and it has two properties. The hasTargetFile property defines the file (instance of the File class) to be requested and the hasSharingProtocol property specifies the protocols supported by the requester. SharingProtocol class is a base class for all various protocols enabling file sharing. The Fig. 3 also illustrates several example subclasses for the SharingProtocol class and it is possible to introduce new subclasses when necessary.

To achieve autonomous file sharing on the semantic level a KP has to be both be able to subscribe to all the available files and all file requests to the files the KP has published into the SIB. This way the KP will be informed when new file has been published or a file request to his file has been made. Because WQL hides the RDFS reasoning from the application developer it is practical to use WQL when performing these operations. An abstract syntax for Wilbur value subscription can be presented in the following way:

$$R = S_{WQL}(n, path) \tag{1}$$

where S_{WQL} is the WQL subscription operation, R denotes the result nodes, n is the start node of the subscription and path specifies the labels (predicates) to be traversed from the start node in the RDF-graph. The WQL subscription that returns all available files can be now presented as:

$$R = S_{WQL}(\mathit{File}, \mathit{inv}(\text{rdf:type})) \tag{2}$$

where *File* is the URI of the File class, *inv()* is a WQL specific operations that requires the path defined by the subexpression *e* to be traversed in a reverse direction and rdf:type is a RDF property denoting that resource is an instance of a Class.

With WQL it is also possible to perform a single subscription that returns all instances of the SharingProtocol class and its subclasses that are connected to the given file via hasSharingProtocol and hasTargetFile properties. The WQL subscription that returns all SharingProtocol instances when a new file request is inserted to the SIB can be presented as:

$$R = S_{WQL}(n_{file}, \mathit{seq}(\mathit{inv}(\text{'hasTargetFile'}), \text{'hasSharingProtocol'})) \tag{3}$$

where n_{file} is the instance of the File class and both *seq()* and *inv()* are WQL specific operations. The operation $seq(e_1,\dots e_n)$ traverses a sequence of n steps of subexpressions $e_1,\dots,e_n$. Fig. 4 illustrates how the queries presented in (2) and (3) traverse through a simplified RDF-graph and return all File and SharingProtocol instance.

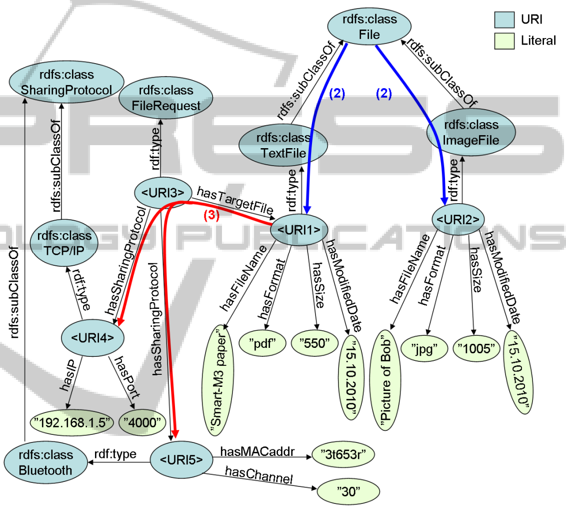


Figure 4: WQL value subscriptions to the instances of SharingProtocol and File classes.

To concretize our approach a scenario that describes how utilizing semantic interoperability enables autonomous file sharing between devices in a Smart Environment is presented next. In the beginning of the scenario there is a Web Service with FTP server capabilities available and information about this service is presented in the SIB. Now a device that is capable of communicating with the service can discover it and then store a file and publish information about the file into the SIB. Another device that has subscribed to all the files in the SIB will now get an indication that a new file is available. Let's then say that the device would need this file but is unable to interact with Web Services because it supports only bluetooth for transferring files. Without the semantic interoperability the device would not even know that there is this kind of file available. However, now a KP in the device can publish information (along the common ontology in Fig. 3) that it wants to request this file. The devices that have this file will get notification from the SIB that the file request has been made. If either of the two devices support bluetooth based file transfer they can now send the file to the bluetooth address specified by the requester.

4 IMPLEMENTATION

In order to demonstrate our approach for autonomous file sharing in practice we have implemented a meeting application for several heterogeneous mobile platforms. These mobile platforms include N900, N97, iPhone and Nexus One. TCP/IP over Wireless Local Area Network (WLAN) is used as the communication solution and Zeroconf protocol provides the SIB discovery in the WLAN. Fig. 5 presents how the mobile devices used in the demonstration interact with each other via the SIB.

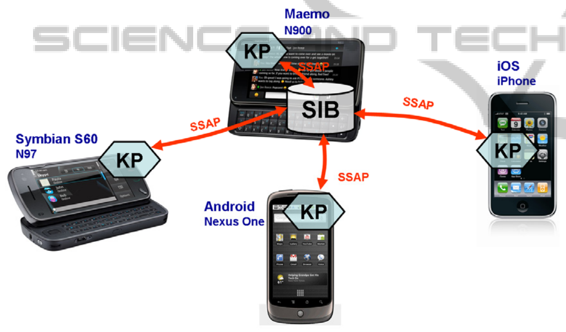


Figure 5: Meeting demonstration set-up.

In addition to the file sharing ontology presented in the section 3 two other classes have been specified for the ontology used in the meeting application. Meeting class presents the available meetings in the Smart Environment and it has three properties. The "hasName" property defines the display name of the meeting. The "hasParticipant" property specifies the participants of a given instance of the Meeting class and the "hasFile" property defines the available files in the given meeting. Participant class represents contact information of meeting participants. The "hasName" and "hasPhoneNumber" properties present the name and phone number of the participant respectively. Fig. 6 illustrates the Meeting ontology classes and their relations.

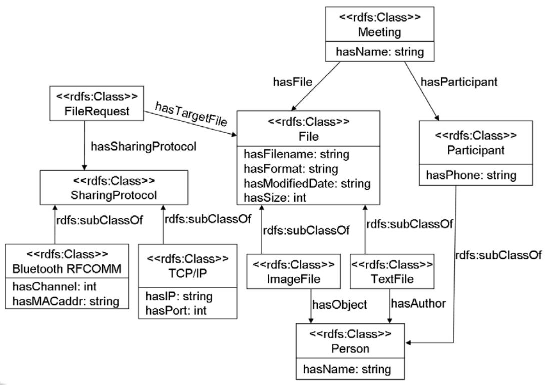


Figure 6: Meeting ontology for file and contact sharing.

With this simple ontology we have created a meeting application where users can create and participate to meetings and share their contact information and files with each other. By introducing new properties for ImageFile and TextFile classes we have also demonstrated how easy it is to create new features to applications by mashing-up existing information. An example scenario of the interaction between meeting KPs and the SIB is presented next.

Let's say that a participant of a meeting publishes a scientific article to the meeting through his KP and defines himself as the author the paper. Another user subscribed to all the files of the meeting via his KP will now be get indication that a file has been published. The second participant is interested in the article published by the first participant and makes a request to the file into the SIB. Now the KP in the first participant's device receives indication from the SIB and sends the file using the file transfer capabilities the receiving KP has. In this case plain TCP/IP was used to stream the file data between devices.

A new feature to this application can be developed by utilizing the properties defined for ImageFile and TextFile classes. By performing a WQL subscription to authors of the file the KP of the second user can be informed when additional information about the author is available. Let's say a third person now decides to take a picture of the author and publishes this picture with the metadata specifying that the author is in the picture into the SIB. Assuming there is also contact information of the first participant available the KP in the second participant's device can now autonomously request the picture and inform the user that there is additional information about the author available. The third user can now contact the possibly unknown author if he wants to discuss about the article. Fig. 7 illustrates the interaction between KPs and the SIB in the presented scenario.

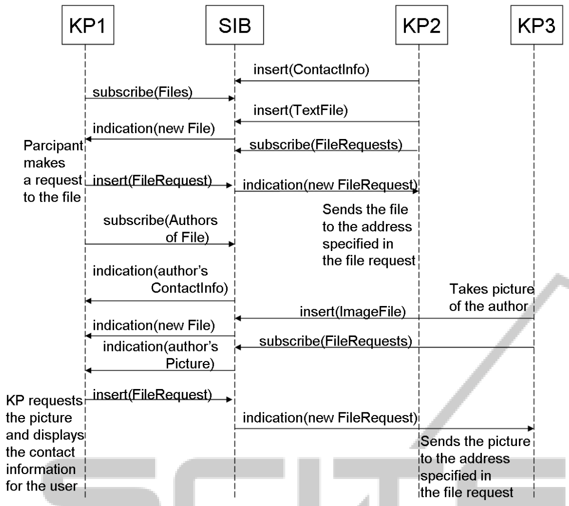

KP1
SIB
KP2
KP3
insert(ContactInfo)
subscribe(Files)
insert(TextFile)
indication(new File)
subscribe(FileRequests)
Parcipant makes a request to the file
insert(FileRequest)
indication(new FileRequest)
Sends the file to the address specified in the file request
subscribe(Authors of File)
indication(author's ContactInfo)
Takes picture of the author
insert(ImageFile)
indication(new File)
indication(author's Picture)
subscribe(FileRequests)
KP requests the picture and displays the contact information for the user
insert(FileRequest)
indication(new FileRequest)
Sends the picture to the address specified in the file request

Figure 7: Sequence diagram illustrating the interaction between KPs and the SIB.

5 CONCLUSIONS AND FUTURE WORK

This paper presented a novel approach for autonomous file sharing in Smart Environments. The fundamental idea of the approach was based on utilizing semantic interoperability platform called Smart-M3 in file sharing context. Necessary information about files, file transfer protocols and file services was modeled using ontology and a novel solution for negotiating about the possible file sharing methods was presented. The approach was demonstrated by implementing a meeting application for heterogeneous mobile platforms from various vendors. The meeting application also illustrated the flexibility and expandability advantages of the ontology-based information presentations approach to create new applications and features by mashing-up existing information.

The paper also presented how describing information about the content of files can take the autonomous file transfer to the next level. However, we only scratched the surface by defining the authors of text files and objects of images. The next step to be done is to describe the contents of files more specifically. This way the KPs could automatically make more advanced searches to the files based on the context and user preferences.

ACKNOWLEDGEMENTS

This work has been funded by the Device and Interoperable Ecosystem (DIEM) and Open Ubiquitous Technology (OPUTE) projects.

REFERENCES

Berners-Lee, T., Hendler, J., Lassila, O., 2001, The Semantic Web, *Scientific American Magazine*, [online] Available at: http://www.scientificamerican.com/article.cfm?id=the-semantic-web [Accessed 18 October 2010].

Cheshire, S., Steinberg, D., 2005. *Zero Configuration Networking, the Definitive Guide.* O'Reilly Media.

Lassila, O., 2002. Taking the RDF Model Theory Out for a Spin", ISWC 2002, *1st International Semantic Web Conference*, 9-12 June 2002 Sardinia, Italia, pp. 307-317.

Liuha, P., Soininen, J., Otaolea, Raul, 2010. SOFIA: Opening embedded information for smart applications. In *Embedded World 2010,* Nuremberg, Germany, 2-4 March, 2010.

NoTA World Open Architecture Initiative website, [online] Available at: http://www.notaworld.org/ [Accessed 12 October 2010].

OMG: The OMG's CORBA Website, [online] Available at: <http://www.corba.org/> [Accessed 12 October 2010].

OSGi Alliance website, [online] Available at: http://www.osgi.org/Main/HomePage [Accessed 12 October 2010].

RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004, [online] Available at: http://www.w3.org/TR/rdf-schema/ [Accessed 14 October 2010].

SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, [online]. Available at: http://www.w3.org/TR/rdf-sparql-query/ [Accessed 14 October 2010].

Suomalainen, J., Hyttinen, P., Tarvainen, P., 2010. Secure information sharing between heterogeneous embedded devices. *In Proc. ECSA 2010*, pp. 205-212.

UPnP forum, 2008. *UPnP Device Architecture 1.1*, 136 p, [online] Available at: < http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf> [Accessed 12 October 2010].

W3C: The W3C's Web Services Activity website, [online] Available at: <http://www.w3.org/2002/ws/> [Accessed 12 October 2010].

Weiser, M., 1991. The Computer for the 21st Century, *Scientific American*, 263(3), pp. 94-104.