Using Capability Profiles For Appliance Aggregation

Mark H. Butler
Information Infrastructure Laboratory
HP Laboratories Bristol
HPL-2002-173 (R.1)
November 27th , 2002*

E-mail: mark-h_butler@hp.com

appliance aggregation, capability negotiation, device profile, delivery context

This paper reviews how several existing standards try to address specific use cases for appliance aggregation. These standards all rely on devices being able to describe their capabilities to other devices. In order to do this, each standard defines its own idiosyncratic profile structure and profile vocabulary. Furthermore applications using this information often need to perform standard tasks such as selection, generation or adaptation based on this information. The problems of devising profile vocabularies, profile structures and matching and selection algorithms are already being explored in the Adaptable Web Delivery Project in HP Labs. This paper discusses the relevance of this work to appliance aggregation and highlights steps that can be taken to resolve the current proliferation of standards.

* Internal Accession Date Only Approved for External Publication
IEEE International Workshop on Networked Appliances, Liverpool, UK, 30th October 2002

Using Capability Profiles for Appliance Aggregation

Mark H. Butler, PhD
mark-h_butler@hp.com
Digital Media Systems Laboratory,
HP Labs Bristol

***Abstract*-This paper reviews how several existing standards try to address specific use cases for appliance aggregation. These standards all rely on devices being able to describe their capabilities to other devices. In order to do this, each standard defines its own idiosyncratic profile structure and profile vocabulary. Furthermore applications using this information often need to perform standard tasks such as selection, generation or adaptation based on this information. The problems of devising profile vocabularies, profile structures and matching and selection algorithms are already being explored in the Adaptable Web Delivery Project in HP Labs. This paper discusses the relevance of this work to appliance aggregation and highlights steps that can be taken to resolve the current proliferation of standards.**

I. INTRODUCTION

In the future, emerging wireless technologies such as 802.11[5] or Bluetooth[3] will allow users to easily set up personal or environmental area networks. Such networks are distinct from conventional local area networks in a number of ways: firstly devices may be added or removed on an ad-hoc basis. Secondly they feature appliances that use embedded software. A major difference between appliances and PCs is whereas PC's allow modification to the software environment e.g. the addition of device drivers or application software to support a device, on appliances it is often simply not possible to upgrade software in this way. Thirdly conventional device drivers often suffer from configuration or compatibility problems. Such problems cannot be tolerated here as aggregation and disaggregation occurs frequently so must operate in a seamless manner. Fourthly in appliance aggregation the software that allows devices to work together must be able to cope with wide variation in input and output capacities amongst appliances. For example an OCR application associated with a scanner will require different user interfaces depending on whether it is mediated via a PDA or via a PC.

The author of this paper is involved in the Adaptable Web Delivery Project at HP Labs Bristol. Although this project is not considering the problem of appliance aggregation directly, we are working on two related problems: firstly how devices can describe their capabilities to other devices, and secondly how these capability descriptions can be used to adapt web content or services to those devices. For more details of this work, see the author's web page[21]. An early survey[22] by this project looked at related standards for synchronizing address databases between devices and for connecting devices in an on-demand fashion. Just as with appliance aggregation, here the problems are compounded by the fact we are dealing with embedded devices so it is not possible to update software. This paper discusses how this work relates to appliance aggregation and identifies some important research issues that are common to both areas.

II. USE CASES

In order to better understand the problems of appliance aggregation, it is useful to consider some concrete use cases for personal area networks. Note the use cases presented here are not exhaustive and where possible multiple examples have been chosen to demonstrate the pervasive nature of these use cases for different types of devices.

A. Seamless device interaction

In the first use case, we want to connect two or more devices in a seamless fashion without having to install device drivers. For example we may want to connect a phone, a PDA or a PC to a printer. Here the devices need to negotiate a protocol and format as it is likely that a PC, by virtue of being a more complex device, will support more protocols and formats than a phone so will be capable of using the printer in a more complex way. An alternative use case might be connecting a more unconventional device such as a cordless keyboard to a phone, a PDA or a printer. Two examples are used deliberately to demonstrate that simply defining an appliance protocol for printers will not solve this use case: rather we need a generalised mechanism for capability description, capability selection and subsequent appliance interaction that can be applied to devices in general.

B. Seamless device synchronization

In the second use case, we may want to synchronise data between two or more devices in a seamless fashion. A common example here is synchronising the address book between a phone and PDA or a phone and a PC. For an alternative example consider a user who has a high capacity portable digital audio device that they wish to synchronisewith their home media server. Again the fact that multiple use cases exist point to the need for a generalised mechanism for synchronization rather than mechanisms just aimed at address books and other mechanisms aimed at media.

C. Seamless device independence

The third use case describes the problem of device independence[13] i.e. we want to be able to display and interact with web content, services or documents on any appliances present in the aggregation. Devices in the aggregation may vary greatly in terms of what input and output modalities they possess e.g. one device may support speech input, pen input, sound output and video output whereas another device only supports keyboard input and video output. In addition even within a given modality, the capabilities of devices may vary e.g. screen size in pixels, type of keyboard available etc.

D. Seamless aggregate multi-modality

Our fourth use case is the primary use case from multi-modality i.e. we want to be able to concurrently display and interact with web content, services or documents on two or more appliances present in the aggregation. Here the adaptation process may be complicated as we may require that aggregates of devices render content in different ways to those same devices when used in isolation. Secondly it is necessary to provide synchronization between devices so that when interaction occurs, all the devices involved are updated concurrently. Examples of this use case include using a phone and a PC concurrently to use an on-line bank; alternatively a PDA may aggregate with a media server and an interactive TV to allow it to act as a universal remote that allows a user to query, select and preview content prior to it being displayed on the main display.

E. Seamless session transfer

The fifth use case is an extension of three and four. Here we may wish to switch between devices or groups of devices mid-session. This will require a re-negotiation of adaptation. For example web content may be rendered in a simplified manner on a smart phone. However when we wish to output that content to a printer, we may require a high quality version equivalent to the version we would receive when using a PC. This requires that different devices in the aggregation can negotiate for content or documents on an individual basis.

F. Seamless session transfer with transfer of session data

In the sixth use case, we require that devices not only aggregate and disaggregate in a seamless manner, but that information essential to the task being performed by the user migrates between devices or is retrievable by all devices in the aggregation. Here possible examples include booking a plane ticket via a web service using a PDA, but then switching to a PC to complete the booking.

III. EXISTING TECHNOLOGIES

Hopefully the reader will agree that the use cases in the previous section do all fit within the broad remit of appliance aggregation. This section will outline a number of existing technologies that try to solve a number of these use cases.

A. Universal Plug and Play (UPnP)

Universal Plug and Play[11] is an interconnectivity standard being promoted by Microsoft aimed at seamless device interaction. It specifies how devices can be addressed, discovered, described, controlled, generate events and present information. Here we will concentrate on the description stage. UPnP uses XML in order to provide a structure for describing device capabilities and service descriptions. Currently UPnP does not specify specific device vocabularies; instead it is expected that device manufacturers will devise these vocabularies themselves.

B. SyncML

The SyncML Initiative[8] aims to develop a common synchronization protocol for data between mobile devices and servers i.e. address the seamless device synchronization use case. Devices such as phones only support a limited number of applications: for example most have an address book and some have diaries. SyncML can be used to synchronize entries in these applications, but it is envisioned that it could be used for potentially any file type. Crucially when two devices undergo a synchronization, they have to exchange a description of their capabilities. This is done using the SyncML Device Information (DevInf) standard, which is based on XML but uses a vocabulary created specifically for DevInf.

C. Media Feature Sets

Media Feature Sets (MFS) were proposed by the Internet Engineering Task Force (IETF) to allow devices to describe their capabilities to servers when retrieving web content, exchanging fax messages or emails i.e. seamless device independence. Like the other standards, MFS define a syntax[30] allowing a complex description of capabilities and requirements but allows vendors to define vocabularies via a tag registration procedure[28]. Unlike the standards previously mentioned, MFS is not based on XML. This is a disadvantage as it is easier to process files based on XML due to easy availability of XML parsers. One of underlying design decisions of MFS is that device capabilities can be regarded as constraints. As a result it allows these constraints to be explicitly joined using Boolean operators such as AND, OR and NOT. This is very useful as it can be used to describe when a device has different capabilities for the same vocabulary property depending on the modality or the particular mode of operation. For example, it might be useful to describe that a PDA has different capabilities depending on whether it is being used in landscape or portrait mode. There are a number of IETF drafts and RFCs on MFS: for more details see [31], [32], [33], [34] and [36].

D. CC/PP

CC/PP[12] is a standard proposed by the W3C. Like MFS, CC/PP allows devices to describe their capabilities to servers so addresses seamless device independence. Currently the author of this paper is the chair of the W3C Working Group for CC/PP and technical co-lead on Java Specification Request 188 for CC/PP.

Unlike the other standards mentioned so far, CC/PP is based on RDF[14], the metadata standard being proposed by the W3C as the Semantic Web[15]. CC/PP grew out of earlier work by the Open Mobile Alliance (formerly the WAP Forum)[9] on a standard called UAProf[10] designed specifically for mobile phones. CC/PP and UAProf feature a syntax for describing capabilities based on a two level hierarchy: a CC/PP profile has a number of components and each component has a number of attributes. Therefore in terms of flexibility in structure, CC/PP is less flexible than MFS as it does not allow properties to be explicitly grouped using Boolean operators. However CC/PP, as it is based on RDF, does have the advantage that it supports XML namespaces[20] so profiles can easily incorporate properties defined in different vocabularies at the same time. Also as the WAP Forum is committed to deploying UAProf on current and future devices, CC/PP has received more interest than MFS. UAProf, unlike CC/PP, does define a vocabulary for describing devices. CC/PP is described primarily in one W3C Recommendation Track document: CC/PP Structure and Vocabularies[35]. The author has produced a Java API for servlets that supports UAProf and CC/PP called DELI[24] and as a result of this work a number of issues have been identified with these standards as described in [25] and [26].

E. Wireless Village Initiative

The Wireless Village Initiative[19] seeks to define and promote a set of universal specifications for mobile instant messaging and presence services aimed at mobile devices, mobile services and Internet-based instant messaging services. Like all the other standards described here, Wireless Village also defines its own device capability description format and vocabulary. Here the device capability format[18] is based on XML and it is noticeable that there is some overlap in the vocabulary and the UAProf and SyncML vocabularies: for example all three define the MIME types that a device supports.

F. Bluetooth

In Bluetooth[3], different profiles (confusing in Bluetooth terminology, a profile is a use case) may use different approaches to query information from devices. For example in the Basic Printing Profile[2], the Simple Object Access Protocol (SOAP)[16] is used to query printer attributes. SOAP allows the exchange of XML documents so the attributes are returned to the requesting device encoded in XML. Note here the device has to explicitly request attributes from the printer rather than just receiving a standard profile. As BPP offers a facility called print by reference, where the device tells the printer the URL of the content this could be coupled with device independence to solve the seamless session transfer use case. However apart from using XML and SOAP, it seems Bluetooth does not adopt a standardized approach to creating device capability descriptions and capability vocabularies.

IV. DESIGN PRINCIPLES FOR CAPABILITY NEGOTIATION

As the previous sections emphasise, there is currently a proliferation of standards used by embedded devices that all define a profile structure and often an associated vocabulary. This section discusses the implications of this and what design principles could be used to overcome these issues.

A. Advantages and disadvantages of the proliferation of standards

The current proliferation of standards has both advantages and disadvantages: one advantage is it has allowed groups addressing different use cases to proceed independently without having to consider other use cases. This is important as it reduces the complexity of creating the standard and makes it easier to gain agreement. However the proliferation of standards also has some disadvantages: firstly it has resulted in a duplication of effort when devising standards as the different standards have all had to address the same problems i.e. how to structure profiles, how to create vocabularies and how to process these profiles. There has also been a duplication of effort when writing software to implement these standards: specifically using a single standard here would reduce effort, decrease the cost of software development and also potentially increase reliability. Finally as the section on use cases highlights, when we begin to consider appliance aggregation more widely we find that use cases such as interaction, synchronization or independence are pervasive so having standards that deal with bounded use cases may actually limit appliance aggregation and interaction.

B. Avoiding Proliferation

So what can we do to simplify matters? As already noted these standards differ because they adopt different vocabularies to describe devices, they adopt different formats for conveying the device capability data and they may adopt different negotiation or adaptation schemes once the device has sent information to the other device. Therefore in order to simplify matters, we need to consider each of these variations and determine what we can do to resolve them.

C. Standardising Vocabularies

Clearly the best and most pragmatic way of simplifying interoperability between these standards is to use a set of standard vocabularies. Already, many of the standards convey the same information although they may represent it in slightly different ways: for example most standards need to convey the MIME types supported by the device, the preferred language of the user and the preferred character set. In the author's opinion, achieving commonality in vocabulary is more important than commonality in structure as the syntactic structure of a profile can easily be translated by a machine, but changing the semantic meaning of a profile is much more difficult and may require human intervention.

For example a number of vocabulary modules could be created to describe different input or output modalities or data domains for synchronisation tasks. Then when vendors wish to create a vocabulary that deals with a specific use case, they can re-use these existing vocabularies adding additional properties only where necessary. This re-use should be achieved by referencing the standard vocabulary namespace, then referencing a new namespace for additional properties. Unfortunately some capability standards have already duplicated existing properties when creating new versions of vocabularies. This approach creates complex backward compatibility problems as it is then difficult to map these vocabularies back on to previous versions.

Currently the W3C Device Independence Activity[13] is about to commence work on creating a device vocabulary to deal with the seamless device independence use case but such an activity could potentially encompass other use cases also. Inspiration here could be drawn from the Dublin Core metadata Initiative[4]. This initiative is trying to devise a vocabulary for metadata to supplement existing methods for searching and indexing Web-based metadata, regardless of whether the corresponding resource is an electronic document or a "real" physical object. It does this by providing a semantic vocabulary for describing the "core" information properties, such as "Description" and "Creator" and "Date". Interestingly Dublin Core has been explicitly created in a format independent way so it can be represented using HTML META tags, XML or RDF. As it may be more difficult to standardise the data structures used in the different device capability standards, adopting a standard vocabulary like Dublin Core is an important first step towards integration. However the creation of standard vocabularies, also known as ontologies[6], is not a trivial matter but has been the subject of much academic research in this area. It is important that standardisation efforts consult authorative sources to learn more about this such as [39].

D. Standardising Structures

A second way of simplifying interoperability is to use a common syntactic structure for representing profiles. This was the design rationale behind CC/PP. CC/PP specifies a common structure for capability information but does not specify any vocabularies, as it was hoped this will simplify interoperability. However in the author’s opinion standardising profile structure i.e. syntactics is less important than standardising semantics. Specifically there seems to be a misunderstanding that as RDF is the basis of the Semantic Web, that simply using RDF adds “semantic understandability”. RDF provides well founded model-theoretic semantics[28] i.e. an abstract mathematical account of how to interpret an RDF model but it does not automatically provide real world semantics, i.e. an interpretation of the model as statements about the external world. Furthermore RDF may be regarded as a kind of “machine code” for knowledge representation, similar in some ways to conceptual graphs proposed by John Sowa[39]. Therefore for many data modelling tasks it may be easier to start with an ontology language, such as DAML+OIL[1], that is operating at a higher level abstraction that can then be mapped back to RDF. If we cannot use a ready-made ontology language, then it is essential we have a clear understanding of the data model we require before we start to think about RDF. Unfortunately CC/PP does not clearly differentiate its data model from the RDF representation.

E. Similarities in structure in existing standards

By examining similarities in the structure of existing standards, it is possible to start to identify a data model for capability negotiation. For example several of the standards mentioned previously describe device capabilities using profiles consisting of two types of elements: descriptive elements and structural elements. *Descriptive elements* describe the properties of the device e.g. the width of the device screen in pixels. *Structural elements* on the other hand perform two roles: they group related elements together or they provide a *context* so that it is possible to distinguish between multiple descriptive elements that refer to the same property in a profile or a set of profiles i.e. support disambiguation. Here the word context is used as defined in [39] as a way to allow subjects to have multiple values of the same property because the contexts for the property values are different.

When the other device involved in the capability negotiation receives the profile, it typically interprets the descriptive elements in one of three ways. If a descriptive element has a single value it is regarded as a constraint that must be met by the other device. If it contains multiple values then it is either regarded as a choice available to the other device or it is necessary to perform resolution i.e. examining the contexts of the multiple values in order to select one and use it as a constraint or select a subset of values which may be regarded as a choice. In capability negotiation, there is an important distinction between application specific resolution methods where the resolution mechanism is specific to the application and application independent resolution mechanisms. For example the UAProf standard specifies application independent resolution mechanisms whereas CC/PP does not propose a complete set of resolution mechanisms so makes the assumption that application dependent resolution mechanisms will be used. Clearly in order to avoid the current proliferation of standards it will be necessary to propose application independent approaches to resolution.

F. Data Validation

Another important problem when describing device capabilities is it is very desirable to have a method of validating profiles. This can be used to offer some guarantee of interoperability when two different devices interact. Currently XML has much better provision for validation than RDF. This is because there are a number of schema languages available for XML such as XML Schema[17] or Relax-NG[7]. Although RDF does have a schema language called RDF Schema, it is not really intended for data validation so it does not offer any mechanisms for performing tasks such as checking literal values. Clearly if RDF is to be used, it is desirable to reuse one of the existing XML schema languages in some way in order to validate profiles. This is an area for further research [38].

G. Preference Ordering

Another shortcoming of CC/PP compared to Media Feature Sets or even content negotiation in HTTP/1.1[23] is that often when comparing the capability descriptions of two devices, there are several alternatives that both devices have in common. Therefore there needs to be some way of determining which is the preferred alternative: in HTTP/1.1 this is performed by a mechanism called preference ordering. Device independence, device synchronisation and device interoperability may all require some mechanism for preference ordering. Currently it is not clear what the best solution is for preference ordering: for example the approach used in HTTP/1.1 is interesting because it allows both devices to express a preference but has received some criticism because it uses arbitrary numerical values to represent preferences so the outcome of a preference negotiation may be unpredictable. Alternatively it is possible to regard the preferences expressed by each device as complete orderings. Assuming there are no conflicts between these orderings, it is possible to derive a partial ordering from this information. However in all cases additional information is then required in order to turn this partial ordering back into a complete ordering in order to select choices.

H. Selection And Matching

Significantly most of the existing standards have not explicitly considered query, selection or adaptation and have left this to implementers. However as vocabularies and profile structures increase in complexity, it may be desirable to investigate this matter further. For example often devices have to select, generate or adapt data based on information in profiles. As vocabulary size increases, device profiles become increasingly complex so the profile can be considered as a point in a high dimensional space of possible alternatives. Recent work[27] by the author has investigated a mechanism called capability classes to map this to a lower dimensional space in order to simplify matching, selection and adaptation. Such mechanisms may be universally useful with different capability description standards or a future single capability description standard may wish to consider this matter in more depth.

I. Dealing With Future Use Cases

Some of the use cases described are not met by current standards but in the author's opinion, these use cases may also require the creation of device profiles. Therefore where possible, they should strive to consider the points made in the previous section with a caveat: none of the standards considered here adequately addresses security, trust or privacy issues. When we start to think of session migration, we may need to consider trust and privacy issues particularly if we are dealing with user data. Clearly this is an area for future work: for more details of work that has considered the problems of privacy in CC/PP, see [37].

V. CONCLUSIONS

This report has explained how the problems of capability vocabulary creation, profile structure and profile matching, negotiation and selection, all of which have been studied within the Adaptable Web Delivery Project, are pervasive problems in the domain of appliance aggregation. It is hoped that by highlighting the need to reduce the proliferation of capability description standards in this area it will be possible to simplify the task of aggregating appliances and allow such aggregations to support a wider range of use cases than are being currently contemplated by the existing, fragmented standards.

REFERENCES

[1] About DAML, http://www.daml.org/about.html

[2] Bluetooth Basic Printing Profile, http://www.bluetooth.org/docs/BPP_0_95a.pdf

[3] Bluetooth SIG Member Site, http://www.bluetooth.org/

[4] Dublin Core Metadata Initiative, http://www.dublincore.org/

[5] IEEE P802.11, The Working Group For Wireless Lans, http://grouper.ieee.org/groups/802/11/

[6] Introduction to ontologies, http://www.semanticweb.org/knowmarkup.html#ontologies

[7] OASIS Technical Committee Relax-NG, http://www.oasis-open.org/committees/relax-ng/

[8] The SyncML Initiative, http://www.syncml.org/

[9] The WAP Forum, http://www.wapforum.org/

[10] The WAP Forum UAProf, http://www1.wapforum.org/tech/terms.asp?doc=WAP-248-UAProf-20011020-a.pdf

[11] Universal Plug and Play, http://www.upnp.org/

[12] W3C CC/PP Working Group, http://www.w3.org/Mobile/CCPP/

[13] W3C Device Independence Activity, http://www.w3.org/2001/di/

[14] W3C RDF Working Group, http://www.w3.org/RDF/

[15] W3C Semantic Web Activity, http://www.w3.org/2001/sw/

[16] W3C SOAP / XML Protocol Activity, http://www.w3.org/2000/xp/Group/

[17] W3C XML Schema Activity, http://www.w3.org/XML/Schema

[18] Wireless Village Device Capability Format, see section 6.6.1 of the Client Server Protocol DTD document, http://www.wireless-village.org/docs/WV_CSP_DTD_v1.0.pdf

[19] Wireless Village Initiative, http://www.wireless-village.org/

[20] T. Bray, D. Hollander and A. Layman, W3C Recommendation, Namespaces in XML, January 1999, http://www.w3.org/TR/REC-xml-names/

[21] M. H. Butler, External web page, http://www-uk.hpl.hp.com/people/marbut/

[22] M. H. Butler, Current Technologies For Device Independence, HPL-2001-83, http://www.hpl.hp.com/techreports/2001/HPL-2001-83.html

[23] M. H. Butler, HPL-2001-190, Implementing Content Negotiation using CC/PP and UAProf, http://lib.hpl.hp.com/techpubs/2001/HPL-2001-190.pdf

[24] M. H. Butler, HPL-2001-260, DELI: A delivery context library for CC/PP and UAProf, http://www-uk.hpl.hp.com/people/marbut/DeliUserGuideWEB.htm

[25] M. H. Butler, HPL-2002-35, CC/PP and UAProf: Issues, improvements and future directions. http://www.hpl.hp.com/techreports/2002/HPL-2002-35.html

[26] M. H. Butler, HPL-2002-73, Some questions and answers on CC/PP and UAProf, http://www-uk.hpl.hp.com/people/marbut/someQuestionsOnCCPP.htm

[27] M. H. Butler, HPL-2002-89, Using Capability Classes to classify and match CC/PP and UAProf profiles, http://www-uk.hpl.hp.com/people/marbut/capClass.htm

[28] P. Hayes, W3C Working Draft, RDF Model Theory, 29 April 2002, http://www.w3c.org/TR/rdf-mt/

[29] K. Holtman, A. Mutz and T. Hardie, IETF RFC 2506: Media Feature Tag Registration Procedure, March 1999 http://www.faqs.org/rfc/rfc2506.txt

[30] G. Klyne, IETF RFC 2533: A Syntax for Describing Media Feature Sets, March 1999 http://www.faqs.org/rfc/rfc2533.txt

[31] G. Klyne and L. McIntyre, IETF RFC 2531: Content Feature Schema for Internet Fax, March 1999 http://www.faqs.org/rfcs/rfc2531.txt

[32] G. Klyne, Draft, A revised media feature set matching algorithm, July 1999 http://www.ics.uci.edu/pub/ietf/http/draft-klyne-conneg-feature-match-00.txt

[33] G. Klyne, IETF RFC 2913, MIME content types in media feature expressions, September 2000 http://www.faqs.org/rfcs/rfc2913.html

[34] G. Klyne, IETF RFC 2912, Indicating media features for MIME content, September 2000 http://www.ietf.org/rfc/rfc2912.txt

[35] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, W3C Working Draft, Composite Capability / Preference Profiles (CC/PP): Structure and Vocabularies, 15 March 2001 http://www.w3.org/TR/CCPP-struct-vocab/

[36] L. Masinter, D. Wing, A. Mutz, K. Holtman IETF RFC 2534: Media Features for Display, Print, and Fax, March 1999 http://www.faqs.org/rfc/rfc2534.txt

[37] H. Ohto, J. Hjelm, W3C Note, CC/PP Implementors Guide: Privacy and Protocols, December 2001, http://www.w3.org/TR/CCPP-trust/

[38] C. Smith, M. H. Butler, HPL Technical Report, Validating CC/PP and UAProf Profiles (Forthcoming)

[39] J. F. Sowa, Knowledge Representation, Brooks/Cole, 2000. ISBN: 0-534-94965