# Persistent identifiers - an overview

Author: Juha Hakala
Senior adviser, The National Library of Finland
juha.hakala@helsinki.fi

## Abstract

This article describes five persistent identifier systems (ARK, DOI, PURL, URN and XRI) and compares their functionality against the cool URIs. The aim is to provide an overview, not to give any kind of ranking of these systems.

## Introduction

Bibliographic identifiers such as ISBN and ISSN have been in use since the 1970s. Web-driven rapid increase of electronic publishing presented a major challenge to many traditional identifier systems; issues concerning for instance the scope and granularity have been tackled in various ways, but it would be premature to claim that we know exactly how the traditional identifiers can be used in the Internet.

One of the major changes digital publishing has fostered is the development of identifiers for works (understood according to the FRBR model). ISTC, International Standard Text Code [1] is an identifier for textual works and their various expressions, such as Joyce's Ulysses and the translations made from the original English version. ISBN, International Standard Book Number [2] can be used to identify manifestations of these works / expressions, such as the PDF version of the English text. In the Web, we have been concerned with manifestations, but Persistent identifier (PI) systems should accommodate works as well, since they are a convenient means of linking the manifestations together.

Traditional identifiers such as ISBNs are not and will not be actionable in the Internet as such. This means, among other things, that the character string “ISBN 951-45-9942-X” is not and will not be interpreted as a hyperlink by Web browsers, whereas a persistent identifier incorporating this ISBN is a hyperlink when expressed as HTTP URI. Unlike most other URIs, http://urn.fi/URN:ISBN:951-45-9942-X is a persistent link to the resource.

Persistent identifiers have several tasks, but perhaps the most important ones are that they render the traditional identifiers actionable in the Web, and provide persistent links to the resources. Using a PI, the user can trust that he or she will get the appropriate work, even if the physical location of its manifestation has changed. In practice, the PI has to be mapped to an up-to-date locator or locators which facilitate access to physical manifestation(s) of the resource.

Before we move on, it is necessary to clarify what we mean by persistence. In this context, electronic resources and the Web, the meaning of the term is not clear. The main developer of the ARK system, John Kunze, has suggested, in the DCC workshop on persistent identifiers, that persistence simply means that 'an identifier is valid for long enough' [3]. It may be better to say that

- persistent identifiers should only be assigned to resources that will be preserved for long term, that is, over several hardware and software generations;
- a persistent identifier and the services it provides should be at least as persistent as the resource identified. The resource may undergo several migrations and the outdated version / versions may no longer be accessible and / or usable. A user who has a PI of an old manifestation of a resource should be redirected to the latest version available, or to work level metadata, which may enable acquisition of the work in some other form, such as print.

## PI systems

The first PI systems emerged in the Mid-1990s, soon after the Web itself (and the problem of non-persistence of the URLs) was introduced. Good overviews have been written about PIs over the years; the most complete one being [4], but there is still some lack of clarity concerning for instance the relation of traditional and persistent identifiers and the relation between cool URIs and persistent identifiers. This overview aims at clarifying these and some other PI-related issues.

The major PI systems are, in chronological order:

- Handle, 1994
- Persistent URL (PURL), 1995
- Uniform Resource Name (URN), 1997
- Archival resource keys (ARK), 2001
- Extensible resource identifier (XRI), 2005

The organizational background of these systems is diverse. URNs were developed by the Internet Engineering Task Force IETF [5] Uniform Resource Names working group [6], which was closed in 2002. Those who are interested in history of PIs in general and URNs in particular may find it interesting, that a part of the email archive of the working group has been included in the email archive of the present URN email list [7].

A URN-related Birds of a Feather -meeting was held at the IETF 78 conference in Maastricht, The Netherlands in July 2010 [8]. The main aim of that meeting was to assess the need for re-establishing the URN working group; the answer was affirmative. Three URN-related RFCs have already been revised and published as Internet drafts, namely URN syntax [9], and namespace registrations for ISBN [10] and NBN [11].

ARK and Handle have a direct link to IETF. Although IETF did not start the work, the developers of these systems (for ARK, California Digital Library [12] and John Kunze, and for Handle the Corporation for National Research Initiatives [13], decided to establish these systems as Internet standards or at least to publish

them as informational RFCs. CNRI achieved the latter aim when informational RFCs 3650, [14] 3651 [15] and 3652 [16] were published in November 2003. They provide an overview of the Handle system, and specify the Handle service and protocol version 2.1. But as Informational RFCs, these documents are not (Internet) standards. The ARK specification was released as an Internet draft in 2002, and 15 versions have been made available until now, the latest one being [17]. It is not clear if and when ARK specification will become an RFC, and whether it will be informational, experimental or standards track -RFC. The draft 15 expired in November 2008 and, apparently, no new ARK-related Internet drafts have been published since then.

Persistent URLs [18, 19] were developed by the Online Computer Library Center, OCLC [20]. The original PURL toolkit was influenced by OCLC's close involvement with IETF's URI working groups. Subsequent versions, developed by Zepheira [21], have benefited from the latter company's W3C participation. But OCLC has never tried to standardize PURLs in IETF or elsewhere, which means that unlike most other PIs PURLs are and will remain purely a technical solution.

XRI was developed by OASIS, the Organization for the Advancement of Structured Information Standards [22]. The purpose was to provide a standard means for identifying any resource, independent of a physical manifestation of it. There may not be a manifestation at all, in which case the XRI would identify a work. XRI has been heavily influenced by IRIs, Internationalized Resource Identifiers [23], which is another IETF initiative.

To sum up: while most traditional identifiers have been under the wing of ISO, most PIs are linked to IETF in one way or another. There is an obvious reason for this: by definition, PIs must be actionable on the Internet, and IETF is the key organization developing Internet standards. URNs were developed by IETFs own initiative. While this has not prevented other initiatives from striving towards publishing RFCs, it may well be that URN has the best chance of reaching the status of Internet standard.

The technical features of PI systems will be discussed below. It should be noted, that this overview neither evaluates these systems systematically, nor tries to put them into an order of preference. All listed persistent identifiers (except XRI) have a broad installed base with millions of assigned identifiers. Thus, we can safely assume that most of the listed PI systems will themselves be persistent, and are going to remain with us for some time. Even persistent identifiers, however, don't have a guaranteed life time of decades or even less centuries.

Some previous reviewers of PI systems have blurred systems and their implementations. The Digital Object Identifier was not listed above, because it is an implementation (and an important one) of the Handle system (and will be discussed as such below). Neither National Bibliography Numbers nor Life Science Identifiers [24] were listed, since they are just two of the URN namespaces.

Some reviewers have included OpenURL. But it is not a (persistent) identifier, although OpenURL metadata may contain an identifier, and is therefore not included here.

Cool URIs [25, 26] and IRIs, which are an extension of URIs, are not reviewed here as persistent identifiers, since they are based on the notion that PIs such as URNs are not needed. With the help of DNS, claim the proponents of this view, a URL can be made as persistent as a PI can ever be. In an article concentrating on PIs it is still a good idea to analyze this claim a bit.

Nicholas [27] concludes that URIs can be used as persistent identifiers if they are properly managed, but is concerned by conclusions people draw from this and which he believes do not follow from the fact that URIs can be used as identifiers:

- *A universal service protocol (such as HTTP) is the same thing as a universal identifier scheme.*
- *HTTP URIs are the preferred identifier for all authorities (although they may well be preferred for HTTP-oriented authorities);*
- *HTTP URIs are the preferred identifiers in contexts where HTTP services are not relevant (e.g. internal document management);*
- *HTTP will always be a universal protocol, and persistent identifier providers should assume it will be;*
- *HTTP URIs will capture all functionality, data, or services presented by other identifier schemes;*
- *Identifiers in other schemes should be maintained only to the extent of exposing them under HTTP.*
- *All identifiers, even when mapped to an HTTP URI, must be meaningfully dereferencable through a Web browser.*

As Nicholas notes, underlying the cool URI approach is an idea of HTTP URIs as identifiers. However, even a simple comparison between cool URIs and traditional identifiers reveals many fundamental differences. Assignment of for instance ISBN numbers is strictly controlled by the ISBN standard and related documentation, and usually done by well trained professionals who know what they are doing. An ISBN, once assigned, will never be given to another book, and the identified book itself will not change. In contrast, anyone can give a URI to anything he / she wants, and the intellectual content, available in a Web location such as http://www.w3.org/, can and probably will change more or less often and may be available in the same time at numerous other locations. Cool URIs cannot track the changes of such web pages and help the user to find the exact version he or she is interested in. This may not look like a problem for W3C, but it is definitely an issue for e.g. National libraries which preserve the history of the Web in their Web archives (and have the option of using URNs or other PIs to identify each Web page and files it contains).

If a system allows anyone to provide persistent identifiers, it is likely that the timeframe its designers had in mind is relatively short. There is a broad consensus that persistence of a resource (and of the links to it) is primarily an organizational issue. A document and link maintained by one person is not likely to survive longer than the person himself / herself, and cool URI –type simple infrastructure may be sustainable in this time frame. But a National library is likely to be able to preserve digital publications longer than most other organizations, given its legal obligation to do that. In a similar manner, national archives will preserve access to digital governmental documents for the future generations. These organizations will also build full scale long-term preservation systems which will require more functionality than e.g. cool URIs can facilitate.

PI systems do differ with respect to services. Some are, at least in theory, able to provide a broader set of them than the others. In practice, the differences may be less pronounced than on paper, since some features of e.g. the URN system have not been implemented.

A common topic in PI discussions is protocol independence. Some experts believe that the systems inherently dependent on the HTTP will persist since HTTP itself will not disappear, or at least that by the time HTTP dies there will be a workaround in place. Pessimists believe that over sufficient time the changes will be so fundamental that no underlying technical infrastructure such as transfer protocols will survive. It will take some time before mankind knows the answer, but meanwhile it is possible to play it safe by choosing a protocol independent identifier system.

## PIs and standardization

In spite of the claims for the contrary [28], there is not a single fully standardized PI system out there. For instance, there are three informational RFC which outline the Handle system, but it is a fundamental mistake to describe Handle-related RFCs [14, 15, 16] as standards. They are informational documents that do not specify an Internet standard of any kind. As of this writing, there are no attempts to turn the Handle system into an IETF standard.

There is an American national standard [29] specifying the DOI syntax. Following the standardization in the U.S.A., the International DOI Foundation initiated the DOI standardization process in ISO TC 46. By summer 2010, this work, which has led to numerous modifications of the original NISO DOI standard, is coming close to completion. But the Handle technical infrastructure that DOI uses will not become an ISO standard.

Most RFCs describing the URN system are also informational or experimental. RFC 2141 [30] which outlines the URN syntax belongs to the former group. This RFC, written in 1997, is by now out of date and the PersID initiative [31] is revising it, alongside with URN namespace registrations of ISBN and NBN. The intention is to review all URN-related RFCs in the light of technological advances and existing URN implementations, and put these RFCs on standard track, that is, update their status into Internet standards. Once this work is completed, nothing prevents the standardization of these RFCs in ISO, but it remains to be seen if there is an interest to do this.

Practical experiences gained from Handle / DOI and URN indicate that standardization of a PI system in its entirety is not easy. ARK is not an exception from this rule - there has been 15 versions of the Internet draft describing ARKs between 2001 and 2008, but IETF has not approved of a publication of even an informational RFC yet.

In the JISC standards catalogue [32], PURL is defined as an identifier standard. However, OCLC has not made an attempt to standardize PURLs, and there does not seem to be an immediate interest to start the process.

Each PI system implementer may consider whether it matters if the solution they have is based on standards. But it should be kept in mind that changing a standard is a controlled and relatively open process. Anyone can in principle participate in the revision of Internet standards, and getting involved with ISO standards work is not too complicated either. In contrast, OCLC can in theory revise PURLs any way they see fit, without consulting the user community. In practice, PURL – standard and tools – have been developed in close cooperation with the user community.

## PIs and traditional identifiers

The relation between PI systems and traditional identifiers is a bit complex, since most persistent identifier systems have a dual character. Each PI specifies a mechanism for resolution and retrieval, including a set of

services a human or software user can request. This functionality enables us to make the traditional identifiers actionable and thus complements them in a useful manner.

PIs may however also compete with the traditional identifiers, since almost all of them (the odd one out being URN) are also identifier systems. It is possible to assign an ARK to a book and use it instead of the book's ISBN (whether this makes any sense is a different matter).

This potential conflict has been dealt with in three different ways. PURL, ARK and XRI ignore it completely; it is up to the user to do whatever makes sense. The ISO Draft International Standard version of the DOI makes it very clear that whenever there is a standard identifier for a resource, then that identifier shall be used as a part of DOI.

The URN solution is embedded: the whole system is based on existing identifier systems, so e.g. serials will be identified by ISSN and books by ISBN, using the URN namespaces assigned for these systems. This arrangement does not eliminate the risk of overlap, since some URN namespaces (that is, identifier systems to be used as a part of URN) may allow the user to bypass the usage of more appropriate standard identifiers.

Possible overlap between traditional identifiers and PIs has not caused real-life problems yet. But this is nevertheless an issue that should be considered carefully when PI systems are standardized. Every identifier system, be it traditional or PI, can be misused, and digital resources are notoriously complex from an identification point of view. But clear scope statements will give the users a clue on what is acceptable and what is not.

## Uniform Resource Name (URN)

Despite the similarities in their basic missions, there are significant differences between PI systems. They will be discussed in the following chapters.

The current URN syntax, specified in RFC 2141 [30], looks simple:

"urn:"<NID>":"<NSS>

where <NID> is a namespace identifier (to distinguish between different identifier schemes) and where <NSS> is the namespace-specific string. Thus, when 'ISBN' is the NID for the ISBN, each URN based on an ISBN begins with URN:ISBN: followed by a namespace specific string; in this case, an ISBN.

Each namespace has to be registered using the process described in RFC 3406 [33]. A namespace can be experimental, informal or formal. As regards formal namespaces, the general principle is [ibid., p. 3]:

*'A formal namespace may be requested, and IETF review sought, in cases where the publication of the NID proposal and the underlying namespace will provide benefit to some subset of users on the Internet. That is, a formal NID proposal, if accepted, must be functional on and with the global Internet, not limited to users in communities or networks not connected to the Internet. '*

Standard identifiers such as ISBN must acquire formal URN namespaces. The resulting RFC must supply for instance the registrant name, and describe how URNs based on this identifier can be resolved in the

Internet. Informal and experimental namespaces have less stringent requirements. There are forty formal namespaces and seven informal ones defined up to now [34], but not all of them are active. There are no experimental namespaces.

URN is unique among PI systems in its requirement of a namespace registration. This has a single drawback (somebody must write the document) and a number of benefits; the registration tells the community how the identifier in question is to be used as a URN. Although experimental and informal namespaces provide a lot of flexibility, the URN users are strongly encouraged to use formal namespaces (existing identifier systems) instead of inventing new / local ones. URN is not an identifier system per se, meaning that URNs must be based on an existing identifier system. This has a big impact on for instance the discussion of the scope of URNs in general. It is pointless to discuss whether URNs can be applied to textual works or collections; if the communities using International Standard Text Code (ISTC, the standard identifier for textual works) or developing International Standard Collection Identifier (ISCI) register URN namespaces, then there will be URNs for names and collections.

The discussions about the scope and granularity (what kind of objects may receive an URN; how do we deal with resources with component parts such as articles within a journal issue and images within the articles) of URNs will take place on the namespace (standard identifier) level, where these issues must be solved in any case. With other PI systems, scope and granularity problems are primarily linked to the use of the PI system as an identifier in its own right. ARK has been designed to facilitate identification of information objects [17]; this certainly accommodates every existing ISO standard identifier and a lot more. The same applies to the Handle system, which has been designed to facilitate identification of digital objects [35] .

Let us look at the URN namespace specification process in a bit more detailed manner. If someone wants to use URNs for identification of names, one – and probably the best - solution will be to specify a formal namespace for the International Standard Name Identifier, ISNI [36]. The ISNI system is designed in such a way that there will be a global database containing all ISNIs and the related metadata [37]. This means that it should be technically easy to fulfill the demands of RFC 3406 for registration of an URN:ISNI namespace. Whether a standard identifier system is suitable for URN resolution depends on how many resolution services there are; if there are more than one, the identifier string must contain some information which supports the resolution process by helping to find a correct resolver.

Whether the ISNI Registration Authority will see the need for the URN namespace registration and subsequent creation of URN resolution services is of course a different matter.

When humans and software agents parse URNs, they can recognize the traditional identifiers used (if any). This may not be the case with other PIs, either because there is nothing indicating what identifier has been used, and/or because the traditional identifier has not been preserved in its original form.

For instance, here is an URN based on ISBN, and then the same ISBN incorporated in a DOI:

URN:ISBN:951-45-9942-X
10.95145/9942-X

The DOI example is based on the ISO DIS 26324 Draft International Standard. In this draft standard, the syntax used to express ISBN as DOI is different than the proposed syntax for ISSN. The ISSN community opted for a syntax which makes parsing easy (by adding “issn.” in the beginning of the DOI suffix, e.g. 10.1038/issn.1476-4687). Some other identifier standard community may choose yet another DOI syntax.

This gives flexibility to these communities, but may make the management of the DOI system a bit more difficult.

## Services

There is no common agreement on services the URN system (or PI systems in general) should support. The experimental RFC 2483 [38] suggests the following:

| | |
|---|---|
| I2L | (URI to URL) |
| I2Ls | (URI to URLs) |
| I2R | (URI to resource) |
| I2Rs | (URI to resources) |
| I2C | (URI to resource description) |
| I2Cs | (URI to resource descriptions) |
| I2N | (URI to URN) |

These services have never been widely implemented, and there is no accepted resolution mechanism supporting them, since, as pointed out by Daigle [39], none of the formal URN namespaces are using the Dynamic Delegation Discovery System (DDDS) outlined in informational RFC 3401 [40] and specified in subsequent Standards track RFCs 3402-3404 [41, 42, 43] and one Best Current Practice–RFC [44]. RFCs 3402-3404 are the only URN-related RFCs which are Internet standards, but like many other standards they have never gained wide acceptance. One reason is the perceived complexity of using DNS Name Authority Pointer Records in URN resolution. According to [45] there are only seven DNS implementations which support NAPTR.

Another problem is that based on the experiences of the PersID project, the services specified in RFC 2483 are not sufficient. The RFC was published in 1999 and reflects therefore the state of the art in digital asset management systems about one decade ago. Since then a lot has happened, especially in long-term preservation of digital resources.

Other PI systems have their own service offerings, to be discussed in respective chapters. But we can still ask why the services specified by RFC 2483 may be insufficient.

If a digital archive opts for a migration strategy, every work will eventually be represented by a “long tail”, a set of manifestations, produced via successive migrations of the files to more up-to date formats. Since each manifestation will be identified separately, many persistent links will point to outdated versions of the works. Most users will probably prefer the latest manifestation, although some may prefer the original (or a version which is as close to it as possible), to minimize quirks, the impact the migrations have had on the resource. Finding the version which fits best is possible only if all manifestations are linked to the work level metadata and each other using the PIs, and there is sufficient technical metadata to assist the user to choose the most appropriate version of the resource.

Thus, a user having just a URN of a single manifestation of a work must be able to acquire the list of URNs related to the one he / she already has, or request descriptive or administrative metadata related to the work and its manifestations. Administrative metadata may be technical (for instance, specification of hardware and software needed for rendering the document), preservation-oriented (description of any changes that have taken place during migration) or rights-related (who can utilize the resource and how).

In addition to the work related services, it is important to be able to check organizational issues, such as the level of commitment the digital archive has in preserving the relevant resource.

To conclude, the RFC 2483 list seems to lack at least the I2Ns service (URI to URNs), a service for querying organizational matter, and finally service parameters with which to specify the desired (descriptive / administrative) metadata in greater details. How to accommodate these services into the future version of RFC 2483 is an open issue, and it may be even less clear how to facilitate this functionality in the resolution service. There are various options, such as DDDS or HTTP content negotiation [46], but none of these may meet all the needs without at least some changes.

As will be seen, none of the existing PI systems provides everything that is needed. But most of them can be extended, should the user community deem that necessary.

## Digital Object Identifier (DOI) as an example of Handles

The DOI system is managed by the International DOI Foundation [47]. This consortium, consisting of both commercial and non-commercial partners, has been active in promoting the system. More than 45 million DOIs have been assigned.

DOI syntax looks simple [48]:

prefix/suffix

Thus, 10.1002/joc.1130 is a valid DOI, consisting of the DOI identifier within the Handle system ("10"), an identifier of the organization that has assigned the DOI ("1002"), and the suffix ("joc.1130") which identifies the resource. In practice, DOIs are usually expressed in the Web as hyperlinks:

http://dx.doi.org/10.1002/joc.1130

This DOI identifies an article published in the International Journal of Climatology. It is based on a local identifier. That is understandable because the standard identifier for journal articles, SICI, Serial Item and Contribution Identifier (ANSI/NISO Z39.56-1996) [49, 50], has never become popular. Note that had URN been used as a PI, a local identifier such as joc.1130 would not have been acceptable since such local identifier system does not have a URN namespace. But it would be possible to express the DOI as URN (urn:doi:10.1002/joc.1130), if DOI had been registered as a URN namespace. Conversion from DOI to XRI or ARK could be difficult or impossible, given the syntax constraints of those systems.

The prefix may be subdivided further. This should not be taken to imply any organizational sub-divisions; a DOI is an opaque string with no embedded meaning. Since there is no limit on the length of either suffix or prefix, DOI can in principle be used by an unlimited number of organizations to identify any number of (and any kind of) resources.

Each DOI (and Handle) has a set of values attached to it. In a way the Handle system contains a record for each Handle, consisting of a group of fields including URL (URIs specifying the location of the object identified by DOI / Handle), EMAIL ( email address of e.g. the administrator of the Handle server containing the DOI) and DESC (unstructured textual description of the object). It is possible to add new values, such as URN (Uniform Resource Name of the object). This architecture makes the system very extensible, at least in

theory. The character set allowed is Unicode with UTF-8 encoding, with the limitations imposed by the URI generic syntax [51].

With the exception of the DOI signature "10", these features are shared by all Handle system–based identifiers.

A major difference between URN and DOI is that with the former there will be just one URN for each instance of a traditional identifier such as ISSN or ISBN. The URN can be created simply by adding urn:isbn: or urn:issn in front of the traditional identifier. Letters "URN" and the namespace identifier can be dropped without sacrificing uniqueness, and parsing the URN string into its three component parts is easy.

In contrast, there can, at least in principle, be several DOIs for each traditional identifier. DOI goes beyond identifying an electronic manifestation of a resource; it also identifies, in the prefix, an access point. The same book may be available via one or more book stores, and they may use a single DOI – as proposed by the ISBN community - which must then resolve to multiple locations, but if that is not possible, each book store may assign a local DOI, using its own organization identifier as the prefix and the ISBN as suffix. In such a case, the DOI prefix (publisher identifier) will direct the users to the correct digital asset management system. With URN, the resolution service must be able to parse the ISBN itself in order to locate the resolver which is able to deal with the URN [10].

A traditional identifier can be transformed into multiple different PIs. Even if a URN:ISBN already exists (created by the National library to facilitate access to the electronic legal deposit collection), a publisher may create a DOI using the same ISBN to enable document delivery via its own server, the California Digital Library may create an ISBN-based ARK which points to the University of California's digital archive, and so on. All these PIs are needed, since they will resolve to different physical locations (URLs). From the users' point of view the problem is not the multiplicity of PIs and resolution services, but the fact that these services are not aware of one another.

Moreover, PIs can be "stacked": http://urn.fi/URN:ISBN:951-45-9942-X is the "real" persistent identifier of the book, but this URN takes the user to a "splash page" describing the resource. From that page, there is a Handle-based persistent link to the resource itself, stored within a DSpace system. When DSpace is replaced by some other digital asset management system in the future, the Handle will be replaced by the URN, or another persistent identifier supported in the new digital asset management system.

## Archival Resource Key (ARK)

ARK [52] was originally developed for the California Digital Library, but the system has other high profile users such as Bibliothèque Nationale de France [53], a strong indication that the system will persist. Technically, it is more versatile than some other PI systems, and unlike for instance URN, all specified functionality has been implemented.

Reflecting the rich functionality, the ARK syntax is more complicated than that of URN and DOI:

[http://NMAH]ark:/NAAN/Name[Qualifier]

Optional data elements are enclosed in square brackets. NMAH stands for Name Mapping Authority Hostport (for instance, example.org or bnf.fr). NAAN means Name Assigning Authority Number. The label "ark:/ distinguishes an ARK from other identifiers.

From the point of view of syntax, the most interesting bit in ARK is the qualifier. It is a string of characters, specifying components and variants, which may involve versions, languages and formats [17].

Example:

http://example.org/ark:/12025/654xz321/s3/f8.05v.tiff

In this ARK, the string /s3/f8 is the component path indicating the relevant part of the resource, and 05v.tiff (meaning version 0.5, TIFF file format) is the variant path.

In addition, adding "?" into the ARK will instruct the ARK resolver to supply a brief metadata record in both human and machine readable form. "??" will yield a preservation commitment statement from the current provider [17]. While the path component of the ARK syntactically conforms to the URI Generic Syntax, it is not obvious if "?" and "??" stick to what RFC 3986 has to say about the use of query, but they do provide functionality that might benefit other PI systems as well. Please note that according to the RFC 3986 the component path could be expressed as fragment.

The ARK qualifier is a powerful tool for the identification of components or variants of the resource. There is a marked difference between ARKs and other PIs in this respect. With DOI, identification of component parts and variants of the resource are possible, but in an uncontrolled manner, in a local identifier. If URNs are used, then it is necessary to use URNs all the way (for instance, it is necessary to give a separate identifier to every version and each fragment of a resource). Some identifiers such as NBN do allow this kind of naming policy, whereas some others do not. Since NAAN can be a traditional identifier such as ISBN, ARK provides a framework within which to increase the version control features and granularity of the latter kind of identifier systems.

For URN and other PIs, ARK suggests that standards development is required / possible in two levels. First, the identifier syntax may allow encoding of qualifier-like data structures for e.g. indicating fragment identifiers. Each namespace could then use this functionality without further ado. The revised edition of the URN syntax [9] does allow fragment usage, but some namespaces may disallow this feature as not applicable to the identifier in question. But it may be a wise thing to give identifier communities the choice: although the ISBN community may not cherish the opportunity for identifying book chapters via extending the ISBN with URN fragment identifiers, but the ISSN namespace could be extended in such a way that it accommodates SICIs as fragment identifiers. Any formal identifier community interested in identification of fragments should, however, specify a policy which defines the acceptable practices. More relaxed namespaces such as URN:NBN cannot have general but only organization-level policies.

Second, it is possible to extend the functionality and syntax specified in the URI Generic syntax. XRI (see below) is a prime example of this approach, also "?" and "??" of ARK belong to this category. Not everyone thinks that tweaking the URI syntax is a good idea; a caveat in such an approach is that these URIs can only be interpreted correctly by applications which do understand the PI syntax fully.

It is possible, that PI systems will gradually evolve into the direction pioneered by ARK and XRI, providing a rich set of services and tools for the purposes of identification, location and retrieval of Web resources.

Whether this happens will depend on the functional requirements of the communities using PIs, and the resources they have for application development. But if PI systems do not provide services and functionality that extend those supplied by cool URIs, the motivation to implement PI systems may eventually diminish.

If the PI systems evolve, the key questions are, what services will be needed, and how they will be supplied. The answers will vary, and may have a fundamental impact on the existing specifications.

## Persistent URL (PURL)

Persistent URLs are just URLs, but they are also [54]:

*'…Web addresses that act as permanent identifiers in the face of a dynamic and changing Web infrastructure.'*

The basis for the PURL development was the participation of the OCLC Office of Research in the early URI development work in the IETF. PURL tools have been available for quite some time, and they are still being developed.

Like any URL, PURLs consist of scheme (http), authority (a domain name and host port like example.com:80) and path. A human or software user has no inherent way of knowing if a URL is just a normal short lived URL or a cool URI or a PURL. Technically these options are quite different. URLs and cool URIs describe the actual location, but PURL

*'... does not directly describe the location of the resource to be retrieved but instead describes an intermediate (more persistent) location which, when retrieved, results in redirection (e.g. via a 302 HTTP status code) to the current location of the final resource.'* [55]

This redirection is a common feature in PI systems, and an extra step cool URI proponents wish to avoid.

The Wikipedia article [55] makes a rather bold comment about the future of PURLs:

*'PURLs are an interim measure — while Uniform Resource Names (URNs) are being mainstreamed — to solve the problem of transitory URIs in location-based URI schemes like HTTP.'*

Whether any of the current or future PI systems can or should be seen as interim measures is an interesting question. Interim solutions may sometimes become persistent, and PURL has been around for 15 years already, which is a long time in the Internet. PURL has a large installed base and the server software has been constantly developed during the last years. For instance, a federation feature which allows PURL servers to co-operate in covering each other during e.g. service outages was added in March 2010 [56]. It would be nice if all PI system software tools were updated as actively as those of PURL.

## Extensible resource identifier (XRI)

While all the other PI systems listed here have a broad installed base, it is difficult to know how many users XRI has. It was developed by an OASIS XRI Technical Committee [57]. The committee is still active, developing Extensible Resource Descriptor [58], a simple tool for describing and discovering resources.

According to the XRI syntax specification [59], XRI is based on both URI and IRI (Internationalized Resource Identifiers) [23]. While IRI extends the URI character set (and specifies the way of "dumbing" IRIs down to URIs), XRI extends the IRI syntax and functionality further.

XRI syntax looks like this:

xri://authority/path?query#fragment

where authority equals (roughly) the ARK Name Mapping Authority Hostport. Unlike other PIs, XRI does utilize URI Generic syntax facilities to the full, which means that XRI has some interesting features not shared by other PIs. Both queries and fragments can be encoded into XRI strings. Moreover, the syntax allows the internal components of an XRI reference to be explicitly designated as either persistent or reassignable. Cross-referencing is possible, that is, XRI references may contain other XRI references or IRIs as syntactically delimited sub-segments. There can also be various authority types: instead of a domain name such as example.com, the authority part of the XRI can be for instance (mailto:john.doe@example.com), in which case the entire XRI could be:

xri://(mailto:john.doe@example.com)/favorites/home

instead of

xri://@example.com/favorites/home

The path component is – following the stipulations of URI syntax - a hierarchical sequence of path segments separated by slash ("/") characters and terminated by the first question mark ("?") or number sign ("#"). The syntax for expressing queries and fragments equals that of IRI.

XRIs are not URNs, but the syntax document claims [59] that XRIs consisting entirely of persistent segments are designed to meet the requirements of RFC 1737, Functional requirements for Uniform Resource Names [60]. Be that as it may, the concept of a persistent identifier with non-persistent components is a novel one, and it is not immediately clear, at least to this author, what benefits such a feature may provide.

Both IRI and XRI represent a step beyond cool URIs (XRI, with its extended functionality, being further removed), but it is not clear whether they qualify as PIs. Neither IRI nor XRI can accommodate existing identifiers, unless they exist somewhere in the path, which may be at least unwieldy, if not impossible. For communities which are committed to using the existing identifiers as a part of PIs whenever possible, this is a show stopper. Any PI, which is unconnected to the existing identifiers, introduces policy issues such as if it is OK to give an XRI to an electronic book or its component parts when there is already an ISBN?

Seen from the library community, XRI – and perhaps also IRI – lack a community that would adopt it for identification of published materials. Many scientific publishers are using DOIs and several National libraries have adopted URNs, and millions of either kind of persistent identifiers have been assigned, but where are the major users of XRI, and what are the shortcomings in other PI systems that it eliminates?

## Summary

Although PI systems have been in existence for more than 15 years, they are not well understood and there is no general agreement on their usefulness compared with cool URIs.

This uncertainty is based on two factors. First, there is no agreement on services (beyond mere persistent linking) PI systems should supply, and no solid and shared technical basis (resolution service) for providing them. Instead of using the Domain Name System and Name Authority Pointer Record for resolution purposes, some PIs are adding service parameters into the identifier strings, which enables simple HTTP transfer.

Second, there is no consensus on how Internet resources should be identified. The choice between persistent identifiers and cool URIs and – should the former option be chosen – between different PIs is not an easy one; [61] is an interesting example of this debate within one URN community.

As a proponent of persistent identifiers, I wonder if a location can be a sufficient identifier, when the same resource can be available in multiple locations, and when, over time, there may be many different resources (or versions of the same work) available in the same location, as amply demonstrated by the Internet Archive. As of August 2010, there were almost 2000 versions of the W3C homepage (http://www.w3.org) available at http://web.archive.org/web/*/http://www.w3.org. Of course the Internet Archive is a proof that some URLs can be persistent; the version of the W3C home page available in May 20th 2010 can be accessed via http://web.archive.org/web/20100520212308/http://www.w3.org/ as long as the Internet Archive exists. A user, however, has no way of knowing if the URI at hand is a persistent one. If it isn't, it may be hard to find a URI that actually is. With persistent identifiers such problems should not exist.

## Links

[1] http://www.istc-international.org/html/

[2] http://www.isbn-international.org/

[3] http://www.ariadne.ac.uk/issue44/dcc-pi-rpt/ Hunter, P.: DCC workshop on persistent identifiers. Ariadne Issue 44, July 2005.

[4] http://www.knaw.nl/ecpa/publ/pdf/2732.pdf , Hilse, H.-W. & Kothe, J.: Implementing persistent identifiers. 2006.

[5] http://www.ietf.org/

[6] http://datatracker.ietf.org/wg/urn/charter/

[7] http://www.ietf.org/mail-archive/web/urn/current/maillist.html

[8] http://www.ietf.org/proceedings/78/agenda/urnbis.txt

[9] http://tools.ietf.org/html/draft-ah-rfc2141bis-urn-02 Hoenes, A.: Uniform resource name (URN) syntax. Internet draft version 02. 2010.

[10] http://tools.ietf.org/html/draft-hakala-rfc3187bis-isbn-urn-00 Huttunen, M. et al.: Using International Standard Book Numbers as Uniform Resource Names. Internet draft version 00. 2010.

[11] http://tools.ietf.org/html/draft-hakala-rfc3188bis-nbn-urn-00 Hakala, J. & Hoenes, A.: Using National Bibliography Numbers as Uniform Resource Names. Internet draft version 00. 2010.

[12] http://www.cdlib.org/

[13] http://www.cnri.reston.va.us/

[14] http://www.ietf.org/rfc/rfc3650.txt Sun, S. et al.: Handle system overview. RFC 3650. 2003.

[15] http://www.ietf.org/rfc/rfc3651.txt Sun, S. et al.: Handle system namespace and service definition. RFC 3651. 2003.

[16] http://www.ietf.org/rfc/rfc3652.txt Sun, S. et al.: Handle system protocol (ver. 2.1) specification. RFC 3652. 2003. [This is the latest version of the Handle protocol]

[17] http://tools.ietf.org/html/draft-kunze-ark-15 Kunze, J. & Rodgers, R.: The ARK identifier scheme.

[18] http://sites.google.com/site/persistenturls/

[19] http://purl.org/docs/index.html

[20] http://www.oclc.org

[21] http://zepheira.com/

[22] http://www.oasis-open.org/home/index.php

[23] http://www.ietf.org/rfc/rfc3987.txt Duerst, M. & Suignard, M.: Internationalized Resource Identifiers (IRIs).

[24] http://wiki.tdwg.org/twiki/bin/view/GUID/LSID

[25] http://www.w3.org/Provider/Style/URI Berners-Lee, T.: Cool URIs don't change

[26] http://www.w3.org/TR/cooluris/ Sauermann, L. et al.: Cool URIs for the Semantic Web

[27] http://www.ariadne.ac.uk/issue62/nicholas-et-al/ Nicholas, N. et al.: Abstract modelling of digital identifiers. Ariadne January 2010, Issue 62.

[28] http://www.ariadne.ac.uk/issue56/tonkin/ Tonkin, E.: Persistent identifiers: considering the options. Ariadne Issue 56 July 2008.

[29] http://www.doi.org/handbook_2000/appendix_1.html ANSI/NISO Z39.84-2005: Syntax for the Digital Object Identifier.

[30] http://www.ietf.org/rfc/rfc2141.txt Moats, R. URN Syntax. RFC 2141. 1997.

[31] http://www.persid.org

[32] http://standards.jisc.ac.uk/catalogue/PURL.phtml

[33] http://www.ietf.org/rfc/rfc3406.txt Daigle, Leslie et al.: Uniform Resource Names (URN) namespace registration mechanisms. Best Current Practice. 2002.

[34] http://www.iana.org/assignments/urn-namespaces/urn-namespaces.xml

[35] http://www.handle.net/

[36] http://www.isni.org/

[37] http://metadaten-twr.org/2010/02/03/international-standard-name-identifier-an-introduction/

[[the tag isni can be assigned to other items in the future, the long URL is unique for your article]]

[38] http://www.ietf.org/rfc/rfc2483.txt Mealling, M. & Daniel, R.: URI resolution services necessary for URN resolution. RFC 2483. 1999.

[39] http://www.isoc.org/tools/blogs/ietfjournal/?p=1705 Daigle, L.: The Curious History of Uniform Resource Names. IETF Journal, vol. 6 issue 1 (June 2010).

[40] http://www.ietf.org/rfc/rfc3401.txt Mealling, M.: Dynamic Delegation Discovery System (DDDS). Part One: The Comprehensive DDDS. RFC 3401. 2002.

[41] http://www.ietf.org/rfc/rfc3402.txt Mealling, M.: Dynamic Delegation Discovery System (DDDS). Part Two: The Algorithm. RFC 3402. 2002.

[42] http://www.ietf.org/rfc/rfc3403.txt Mealling, M.: Dynamic Delegation Discovery System (DDDS). Part Three: The Domain Name System (DNS) Database. RFC 3403. 2002.

[43] http://www.ietf.org/rfc/rfc3404.txt Mealling, M.: Dynamic Delegation Discovery System (DDDS). Part Four: The Uniform Resource Identifiers (URI) resolution application. RFC 3404. 2002.

[44] http://www.ietf.org/rfc/rfc3405.txt. Mealling, M.: Dynamic Delegation Discovery System (DDDS). Part Five: URI.ARPA assignment procedures. RFC 3405. 2002.

[45] http://en.wikipedia.org/wiki/NAPTR_record

[46] http://en.wikipedia.org/wiki/Content_negotiation

[47] http://www.doi.org/

[48] http://www.doi.org/hb.html DOI Handbook. Version 1. February 2010.

[49] http://en.wikipedia.org/wiki/Serial_Item_and_Contribution_Identifier

[50] www.niso.org/standards/z39-56-1996r2002

[51] http://www.ietf.org/rfc/rfc3986.txt Berners-Lee, T. et al.: Uniform Resource Identifier (URI): Generic syntax. RFC 3986. 2005.

[52] https://wiki.ucop.edu/display/Curation/ARK

[53] http://bibnum.bnf.fr/identifiants/index.html

[54] http://purl.org/docs/index.html

[55] http://en.wikipedia.org/wiki/Persistent_Uniform_Resource_Locator Persistent Uniform Resource Locator. Wikipedia article, quoted 2010-07-09.

[56] http://sites.google.com/site/persistenturls/

[57] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri

[58] http://docs.oasis-open.org/xri/xrd/v1.0/cd02/xrd-1.0-cd02.pdf

[59] http://www.oasis-open.org/committees/download.php/15377 Reed, D. et al. Extensible resource identifier (XRI) syntax version 2.0. 2005.

[60] http://www.ietf.org/rfc/rfc1737.txt Sollins, K. & Masinter, L.: Functional requirements for Uniform Resource Names. RFC 1737. 1994.

[61] http://www.hyam.net/blog/archives/325